Tru64 UNIX

Installation Guide — Advanced Topics

This manual discusses advanced installation topics for the HP Tru64 UNIX operating system.

Contents

3   The UNIX Shell

4   Disk Space Planning

5 Customizing the Installation Process

6 Installation Cloning

7 Configuration Cloning

A Installation CDF Attribute–Value Pairs

Tables

About This Manual

This manual describes advanced installation procedures for the HP Tru64 UNIX operating system software. Topics include starting a Full or Update Installation from a RIS server, Installation Cloning, Configuration Cloning, disk space planning, and installing Worldwide Language Support (WLS) after a Full Installation.

Audience

This manual is intended for experienced installers who want to take advantage of the wide range of advanced installation features, such as installation and configuration cloning.

New and Changed Features

The following changes have been made since the Version 5.1A release:

- Links to documents that describe how to work with internationalized software have been added to *Section 2.9*.

- The differences between the UFS and AdvFS file systems has been clarified, as described in *Section 4.3.2*.

- During a configuration cloning, systems to which patches have been applied should not be used to create the `config.cdf` file. This restriction and alternative work arounds are described in *Section 7.4*.

Previous versions of this manual are available on the World Wide Web at the following location:

http://www.tru64unix.compaq.com/docs/

See the New and Changed features section of those versions to see the evolution of this manual.

Organization

This manual is organized as follows:

Chapter 1	Describes how to prepare for and invoke a Full or Update Installation from a Remote Installation Services (RIS) network server.
Chapter 2	Describes how to install the WLS software using the `wwinstall` script or `setld` utility after a Full Installation.

Chapter 3	Describes how to restore file systems, modify disk labels, and perform system maintenance in the UNIX shell environment.
Chapter 4	Describes the disk space planning process if you decide not to use the recommended disk partitions and you want to manually partition your disks.
Chapter 5	Describes how user-defined files and configuration description files (CDFs) are used to customize and extend the installation process.
Chapter 6	Describes the Installation Cloning process, which duplicates the installation from a model system to one or more target systems during a Full Installation.
Chapter 7	Describes the Configuration Cloning process, which duplicates the configuration from an already configured model system to one or more target systems.
Appendix A	Lists and defines the attribute-value pairs in the installation configuration description file (CDF).
Appendix B	Provides samples of user-supplied scripts to invoke during a Full or Update Installation.

Related Documentation

The following documents may be useful references when you are performing advanced installation tasks:

- See your hardware documentation for information about how to set up the processor and its additional devices, setting console environment variables, and troubleshooting guidelines.

- See *Network Administration: Services* for information about network setup and network administration.

- See *Sharing Software on a Local Area Network* for information about setting up Remote Installation Services (RIS) servers, creating and serving software environments, and registering client systems.

- See *AdvFS Administration* for information about administering the Advanced File System (AdvFS).

- See *System Administration* for information about administering and maintaining your system after an installation.

- See *Logical Storage Manager* for information about configuring and administering the Logical Storage Manager (LSM).

- See the *Documentation Overview* for information about all manuals in the documentation set.

Tru64 UNIX documentation is available on the World Wide Web at the following location:

http://www.tru64unix.compaq.com/docs/

Icons on Tru64 UNIX Printed Manuals

The printed version of the Tru64 UNIX documentation uses letter icons on the spines of the manuals to help specific audiences quickly find the manuals that meet their needs. (You can order the printed documentation from HP.) The following list describes this convention:

G   Manuals for general users

S   Manuals for system and network administrators

P   Manuals for programmers

R   Manuals for reference page users

Some manuals in the documentation help meet the needs of several audiences. For example, the information in some system manuals is also used by programmers. Keep this in mind when searching for information on specific topics.

The *Documentation Overview* provides information on all of the manuals in the Tru64 UNIX documentation set.

Reader's Comments

HP welcomes any comments and suggestions you have on this and other Tru64 UNIX manuals.

You can send your comments in the following ways:

• Fax: 603-884-0120 Attn: UBPG Publications, ZKO3-3/Y32

• Internet electronic mail: readers_comment@zk3.dec.com

   A Reader's Comment form is located on your system in the following location:

   /usr/doc/readers_comment.txt

Please include the following information along with your comments:

• The full title of the manual and the order number. (The order number appears on the title page of printed and PDF versions of a manual.)

• The section numbers and page numbers of the information on which you are commenting.

• The version of Tru64 UNIX that you are using.

- If known, the type of processor that is running the Tru64 UNIX software.

The Tru64 UNIX Publications group cannot respond to system problems or technical support inquiries. Please address technical questions to your local system vendor or to the appropriate HP technical support office. Information provided with the software media explains how to send problem reports to HP.

Conventions

`%` `$`	A percent sign represents the C shell system prompt. A dollar sign represents the system prompt for the Bourne, Korn, and POSIX shells.
`#`	A number sign represents the superuser prompt.
`% `**`cat`**	Boldface type in interactive examples indicates typed user input.
`>>>`	The console subsystem prompt is three right angle brackets.
file	Italic (slanted) type indicates variable values, placeholders, and function argument names.
`cat(1)`	A cross-reference to a reference page includes the appropriate section number in parentheses. For example, `cat(1)` indicates that you can find information on the `cat` command in Section 1 of the reference pages.
Ctrl/*x*	This symbol indicates that you hold down the first named key while pressing the key or mouse button that follows the slash. In examples, this key combination is enclosed in a box (for example, Ctrl/C).

1

Installing from a Remote Installation Services (RIS) Server

This chapter contains the following information:

- The prerequisites for installing from a RIS server (Section 1.1)
- A simple test to make sure the network is configured before you begin (Section 1.2)
- How to start an Update Installation from a RIS server (Section 1.3)
- How to start a Full Installation from a RIS server (Section 1.4)

1.1 Prerequisites to Installing from a RIS Server

If your system is located within a networked environment, you may be able to install the operating system from a server on the network if the following items are in place:

- Your site has a system that has been configured to be a **Remote Installation Services** (RIS) server
- Your system is a registered client of a RIS server that is serving Version 5.1B of the operating system

_____ **Note to RIS Administrators** _____

The RIS server must be running Version 4.0 or later of the operating system to be able to serve Version 5.1B to clients.

_____

See *Sharing Software on a Local Area Network* if you are a network administrator and need more information about setting up a RIS server, creating software environments to serve, and registering a client system to the right software environment.

1.2 Is Your System Connected to the Network?

If your system already is running a version of the operating system and you want to perform an Update Installation, ensure that your system can communicate with the RIS server by executing the `/sbin/ping` command to verify the network connection.

To test the network connection, enter the following command and replace *ris_server_name* with the name of your local RIS server:

/sbin/ping -c2 *ris_server_name*

Successful output shows a 0 (zero) percent packet loss, which indicates that your system can communicate with the RIS server over the network. In Example 1–1, the RIS server name is server1.

Example 1–1: Output of the /sbin/ping Command

```
# ping -c2 server1
PING server1 (16.59.124.96): 56 data bytes
64 bytes from 16.59.124.96: icmp_seq=0 ttl=255 time=1 ms
64 bytes from 16.59.124.96: icmp_seq=1 ttl=255 time=0 ms

----server1 PING Statistics----
2 packets transmitted, 2 packets received, 0% packet loss
round-trip (ms)  min/avg/max = 0/0/1 ms
```

Your system is not able to communicate with the RIS server if you obtain the following results from the /sbin/ping command:

- Only the system name and IP address is displayed in the command output if the name server recognizes the system name but the network connection is not set up.

- The message Unknown host is displayed if your system does not recognize the RIS server name.

If you experience any of these results, ask your network administrator to troubleshoot the problem. The *Network Administration: Connections* and *Network Administration: Services* manuals contain extensive network troubleshooting information.

1.3 Starting an Update Installation from a RIS Server

Follow this procedure to start an Update Installation from a RIS server:

1. Ensure that your system meets the prerequisites to installing over the network as described in Section 1.1 and the system is configured on the network as described in Section 1.2. See your network administrator or *Sharing Software on a Local Area Network* if you encounter problems.

2. Complete all Update Installation prerequisite tasks that are described in the *Installation Guide*. These tasks include backing up your current operating system, reading the *Release Notes*, and updating system firmware.

3. As superuser or root, place the system in single-user mode by using the shutdown command.

- The following example shows how to switch to superuser and then shut down the system to single-user mode:

```
% su -
password:
# shutdown +10 Please log out
```

In the previous example, +10 shuts down the system in ten minutes and sends the message Please log out to all logged in users.

After your system is in single-user mode, the screen looks similar to the following:

```
.
.
.
Halting processes ...

INIT: SINGLE-USER MODE
#
```

4. Mount the local file systems:

```
# /sbin/bcheckrc
```

The bcheckrc command invokes the mount -a command and mounts all file systems in the /etc/fstab file, not just the standard UNIX file systems (/, usr, and var). The bcheckrc command also runs the file system check command, fsck, on UNIX file systems (UFS) and starts the Logical Storage Manager (LSM) if necessary. If fsck finds a problem with the / (root) partition, the system shuts down and requires a reboot to fix the file system.

5. Delete the table of Internet addresses to ensure that the routed and gated daemons do not start up during the Update Installation:

```
# route flush
```

6. Enter the /sbin/installupdate command with the following syntax:

/sbin/installupdate [-u] [-nogui] {*ris_server_name:*}

The following describes each option:

- The optional -u flag runs the Update Installation without any user intervention. This option automatically removes blocking layered products, deletes obsolete files, and installs all kernel components thereby eliminating the need for you to respond to any questions during the update.

- The optional -nogui (no graphical user interface) flag invokes the text-based interface on systems with graphics capabilities.

- The RIS server name must be appended with a colon (:).

For example, to start an Update Installation from a RIS server named server1, enter the following command:

```
# /sbin/installupdate server1:
```
As another example, you would enter the following command to run the Update Installation in unattended mode using the text-based interface from a RIS server named `server2`:

```
# /sbin/installupdate -u -nogui server2:
```

_____ **Note** _____

If you have problems starting the Update from a RIS server, see the *Sharing Software on a Local Area Network* manual for troubleshooting information.

_____

7. After the Update has started, you can follow along with the remainder of the Update Installation process as documented in the *Installation Guide*.

1.4 Starting a Full Installation from a RIS Server

Follow this procedure to boot your system over the network to start a Full Installation from a RIS server:

1. Ensure that your system meets the prerequisites to installing over the network as described in Section 1.1 and the system is configured on the network as described in Section 1.2. See your network administrator or *Sharing Software on a Local Area Network* if you encounter problems.

2. Complete all Full Installation prerequisite tasks that are described in the *Installation Guide*. These tasks are in a self-contained chapter and include backing up your current operating system, reading the *Release Notes*, and updating system firmware.

3. Bring your system down to console mode (the >>> prompt). Do one of the following depending upon the current state of your system:

   • If your system is up and already running a version of the operating system, shut down and halt the processor using a command similar to the following :

     ```
     # shutdown -h +10 Please log out
     ```

     In the previous example, the system is shut down and halted in 10 minutes and sends the message `Please log out` to all logged in users. See the *System Administration* manual and shutdown(8) if you need more information about shutting down a system to console mode.

   • If the system is turned off, turn on the processor. The console subsystem prints various start-up and diagnostic messages and ends with the console mode prompt (>>>). However, if your system

is configured for automatic reboots, it will not stop at the console mode but will boot to multiuser mode. To bring the system to the console mode, use the `shutdown -h` command as shown in the previous bullet list item.

4. Certain processors require one or more console environment variables to be set in addition to the standard console variables that are set for all processors. Look for the corporate logo on the front panel of your hardware to determine what kind of processor you have. Next, go to the *Full Installation Procedures* chapter in the *Installation Guide* to find out if your system requires special console variables to be set before the system is booted over the network. Then, continue with Step 5 in this procedure.

5. Clear and reset the `boot_osflags` console variable:

   ```
   >>> set boot_osflags ""
   ```

6. To ensure that your system returns to the console mode in the event of a system crash or power failure during the installation, set the `auto_action` console variable:

   ```
   >>> set auto_action halt[1]
   ```

7. Determine the network adapter device name:

   ```
   >>> show device
   ```

   A device information table similar to the following is displayed:

   ```
   dka400.4.0.6.0          DKA400      RRD43  2893
   dva0.0.0.0.1            DVA0
   ewa0.0.0.13.0           EWA0        08-00-2B-3E-B6-C8
   pka0.7.0.6.0            PKA0        SCSI Bus ID 7
   ```

   The network boot device is shown in the middle column next to the hardware Ethernet address in the third column. In this example, the hardware Ethernet address is `08-00-2B-3E-B6-C8`, and the boot device is `EWA0`.

8. Initialize the `bootp` boot request protocol using the following command syntax:

   set *network-device*_protocols bootp

   set *network-device*_inet_init bootp

   The network boot device name is case insensitive and can be entered in lower case or upper case letters. Using the network adapter device name obtained in Step 7 as an example, enter the following `bootp` initialization commands:

---

[1] After the installation is complete, you may want to reset `auto_action` to `restart` to allow your system to reboot automatically after a power or processor failure.

```
>>> set ewa0_protocols bootp
>>> set ewa0_inet_init bootp
```

9. Before you boot the system in the next step, check Table 1–1 to make
    sure there are no special boot commands for your processor type.

    _____ **Note** _____

    Every attempt was made to make the information in
    Table 1–1 complete and accurate for every supported
    system. However, it is recommended that you also check
    the hardware documentation for your processor to make
    sure it does not require any other console variables or boot
    flags to be set. Hardware documentation is customized for
    each system type, and it is the definitive source for required
    console variables and boot flags.

    _____

Table 1–1: Processor-Specific Network Boot Commands

Processor or Option	Boot Commands
AlphaStation 500, 600, 600A AlphaServer 800, 1000, 1000A Personal Workstation au Series Ultimate Workstation au Series Professional Workstation XP1000	`>>> boot -fl "" ewa`
AlphaServer 4000, 4100 Alphaserver 8200, 8400 ES40, GS60, and GS140 Servers	`>>> boot -fl "" ewa0`
Systems with FDDI devices	See Section 1.4.1

10. Reset the boot file:

    ```
    >>> set boot_file ""
    ```

11. Enter the boot command with the following syntax:

    boot _network-device_

    Using the information obtained in Step 7 as an example, the boot
    command is:

    ```
    >>> boot ewa0
    ```

This completes booting the system over the network to invoke a Full
Installation. The _Installation Guide_ documents the remainder of the Full
Installation.

If you encounter trouble booting your system over the network, see the
Sharing Software on a Local Area Network manual.

1.4.1 Network Boot Option: Fiber Distributed Data Interface (FDDI)

Booting over an FDDI network interface is supported for DEFTA devices.

To boot from an FDDI network interface device, find the device names by entering the show config command when the system is at console level. For example, a DEFTA device appears as PMAF-FA when you use the show config command. Identify the slot number and enter the boot command as noted in Table 1–2.

Table 1–2 shows the required boot devices for booting over the network if your system uses FDDI. Follow the instructions in your hardware owner's guide to update the FDDI firmware before booting over the network.

Table 1–2: FDDI Boot Devices by Bus Type

Bus Type	Boot Device
EISA (Extended Integrated System Architecture)	fra0[a]
PCI (Peripheral Component Interconnect)	fwa0[a]
XMI (Extended Memory Interface)	fxa0[a]

[a] Before you boot over the network, your system must be registered with the RIS server and you need to know the hardware address. To determine the hardware address, at the console mode prompt (>>>), enter the show dev command.

1.4.2 Troubleshooting Network Boot Failures

If your initial network boot fails, enter init at the console prompt and attempt to boot over the network again. If you encounter other problems during the remote server installation, see the *Sharing Software on a Local Area Network* manual, which contains troubleshooting information for network boot failures.

1.4.3 Network Reboot Considerations for Systems with Graphics Devices on an ISA Bus

If your system does not reboot automatically after the software subsets are loaded, and your system has a graphics device on its ISA bus that requires a kernel device driver, you must modify the isacfg entry to match the kernel device driver before rebooting the system.

When you issue the following command, let the input line wrap; do not press the Return key in the middle of the line. The backslash character (\) represents line continuation; do not enter a backslash in the command line.

```
>>> isacfg -mod -slot slot_number -dev device_number    \
    -handle vendor_handle  -etyp 1 -enadev 1
```

In the previous example, replace *vendor_handle* with the handle supplied in the vendor's installation documentation.

If you performed a RIS installation from a RIS area that already has a kernel graphics device driver built into the generic kernel on the RIS server, and you already set the handle to the one specified in the vendor's installation documentation, you do not need to execute this command. If your system does not support the automatic reboot feature, the boot commands will be displayed on the screen.

2

Worldwide Language Support Installation Procedures

This chapter contains the following information:

- An overview of Worldwide Language Support (WLS) (Section 2.1)

- Preparing the system for a WLS installation (Section 2.2)

- Increasing available disk space for WLS software (Section 2.3)

- Using the `wwinstall` script to install WLS software subsets from CD-ROM (Section 2.4.1)

- Using the `setld` command to install WLS software subsets over the network from a RIS server (Section 2.4.2)

- Responding to prompts during the WLS installation procedure (Section 2.5)

- Invoking `/usr/sbin/wwconfig` to tailor the kernel to include Asian terminal options (Section 2.6.1)

- The backup files created by a worldwide installation (Section 2.7)

- Setting language-specific environment variables (Section 2.8)

- Links to related documentation (Section 2.9)

2.1 Worldwide Language Support Overview

The WLS software subsets provide support for many languages and countries other than United States English, which is installed by default. Installing WLS software subsets enables software developers to develop internationalized software that can be used in different countries and lets users work in their native languages.

You can install software to support one or more languages during a Full Installation as documented in the *Installation Guide*, which is the recommended method to install WLS software. However, you also can add language support after the base operating system is installed by using the procedures described in this chapter.

WLS software is installed in the `/usr/i18n` directory, which can be created as its own file system or can be created as a symbolic link to another file system. If a `/usr/i18n` directory or file system does not already exist, you

are prompted to create it if you use the `wwinstall` command to install WLS. i18n is the abbreviation for *internationalization*.

For more information about the terminals and printers supported for different languages, see the *Software Product Description* (SPD). The SPD is located on the CD-ROM labeled *Operating System Volume 1*, and various printable formats of the SPD are located in the `/DOCUMENTATION` directory.

2.2 Step 1: Preparing for a WLS Installation

Perform the following tasks before you start a WLS installation:

1. Read the preinstallation tasks section of the *Installation Guide*, which describes the general preparation you should do before any installation.

2. Make sure that the current version of the base operating system is already installed on your system.

3. Have in your possession a copy of the current version of the *Operating System Volume 1.* If a WLS subset depends on an operating system subset that is not installed on your system, use the `setld` command to install the operating system subset. Dependencies on base operating system software subsets is determined during the WLS installation procedure. You will have to stop the WLS installation and install the dependent subset or subsets before proceeding.

4. If you are installing the WLS software subsets from a RIS server, ensure that your system is registered as a client on the RIS server. The RIS area to which your system is registered must contain the WLS product.

   If you are a network or RIS administrator and need more information about how to set up a RIS server and RIS client, see the *Sharing Software on a Local Area Network* manual.

2.3 Optional Step: Increasing Available Disk Space for WLS Software

The WLS installation procedure loads most WLS files to the subdirectories under the `/usr/i18n` directory. If the `/usr/i18n` directory does not exist, the installation procedure creates it. During the software selection process, all WLS software subsets are considered during `/usr` file system size calculations. If the `/usr/i18n` directory exists, the installation procedure uses it.

If you find that there is insufficient disk space for the WLS software subsets, and you know that you have additional space on other disks or disk partitions on your system, follow the procedures to increase disk space shown in Section 2.3.1 or Section 2.3.2 depending upon the file system type.

2.3.1 Increasing Disk Space in UFS File Systems

Before beginning the installation procedure, follow these steps to increase disk space for the /usr/i18n directory in a the UNIX file system (UFS):

1. Log in as root or use the su command to become superuser.

2. Create the /usr/i18n directory if it does not exist:

    # **mkdir /usr/i18n**

3. Add a line similar to the following to the /etc/fstab file so that the newly created directory is a mount point to a disk partition where there is additional space:

    /dev/disk/dsk2c   /usr/i18n   ufs,rw 0 0

4. Mount the new mount point of /usr/i18n:

    # **mount -a**

2.3.2 Increasing Disk Space in AdvFS File Systems

Before beginning the installation procedure, follow these steps to create an AdvFS file domain for the /usr/i18n directory:

1. Log in as root or use the su command to become superuser.

2. Create a /usr/i18n directory if it does not exist:

    # **mkdir /usr/i18n**

3. Create an AdvFS domain using the following command syntax

    mkfdmn /dev/disk/dsk<*disk_number*>c *domain_name*

    For example, create the i18n_domain domain on dsk2:

    # **mkfdmn /dev/disk/dsk2c i18n_domain**

4. Create an AdvFS fileset:

    # **mkfset i18n_domain i18n**

5. Use a text editor of your choice to add the following line to the /etc/fstab file so that the newly created domain can be mounted:

    i18n_domain#i18n   /usr/i18n   advfs   rw, 0 0

6. Mount the new file domain:

    # **mount -t advfs i18n_domain#i18n /usr/i18n**

2.4 Step 2: Starting a WLS Installation

Starting a WLS installation differs slightly depending upon the source of the distribution media:

- Go to Section 2.4.1 if you are using the CD–ROM labeled *Associated Products Volume 1*.

- Go to Section 2.4.2 if you are using a Remote Installation Services (RIS) server.

During the installation procedure, you are asked some questions about configuring the system. If you need help, enter a question mark (?) to display online help.

2.4.1 Installing from a CD-ROM

Use this procedure to invoke the wwinstall script from CD-ROM:

1. Load the *Associated Products Volume 1* CD-ROM into the CD-ROM drive.

2. Make a directory to be the mount point of the CD-ROM, and then mount the CD-ROM:

   ```
   # mkdir /cdrom
   # mount /dev/disk/cdrom0c /cdrom
   # cd /cdrom/Worldwide_Language_Support/kit
   ```

   In the previous example, /dev/disk/cdrom0c is the name of the CD-ROM device. If you do not know the device name of your CD-ROM device, enter the following command:

   ```
   # ls /dev/disk/cdrom*c
   ```

3. Change directory to the location of the wwinstall script:

   ```
   # cd /cdrom/Worldwide_Language_Support/kit
   ```

4. Invoke the wwinstall script:

   ```
   # ./wwinstall
   ```

Go to Section 2.5 to continue the WLS installation procedure.

2.4.2 Installing from a RIS Server

This section describes how to use the setld utility to install the WLS software subsets from a RIS server. If you want to use the wwinstall command instead, you use a Network File System (NFS) mount to mount the exported RIS area. You than can run wwinstall from the mounted RIS area (this procedure is not documented).

Use this procedure to start a WLS installation from a RIS server:

1. Use the Quick Setup application to set up basic networking services if the network is not configured on the system you want to install. Quick Setup is available from the System Setup application :

```
# /usr/sbin/checklist
```

2. Ask the RIS server administrator to make sure your system is registered as a client of the server that is serving the current version of the WLS software.

3. Start the WLS installation process:

    ```
    # setld -l ris_server_name:
    ```

    Replace *ris_server_name* with the name of your RIS server appended with a colon (:). Go to Section 2.5 to continue the WLS installation procedure.

2.5 Step 3: User Responses During the WLS Installation Process

The prompts displayed during the WLS installation apply to systems where all the prerequisite base operating system software subsets are installed. The installation screens are similar for installations regardless of whether you are using CD–ROM or RIS. However, if you used the setld utility to install from a RIS server, you do not have the option to create a symbolic link to another partition to hold the WLS software.

The wwinstall script calculates the disk space available in the /usr/i18n area and gives you the opportunity to choose how to create the /usr/i18n area. The amount of disk space required in /usr/i18n depends on the country or language you select. For example, installing support for the Japanese language requires more disk space (about 200 MB) than installing support for the Italian language (about 20 MB). See the *Release Notes* for the disk space requirements of each language.

If you are installing from CD-ROM or ran the wwinstall command to install WLS from a RIS server, a message similar to the following is displayed:

```
Most of the subsets will be installed under the /usr/i18n directory.
Depending on the number of subsets you choose to install, you may
need more than 200 MB of free disk space for installation.

You have the following amount of free disk space
available in /usr:

$ df -k /usr/i18n
Filesystem      1024-blocks        Used    Available Capacity  Mounted on
usr_domain#usr     716800       238120       456512     35%    /usr

Two ways to set up the /usr/i18n directory :

[1] Create the /usr/i18n directory
[2] Set up a symbolic link to another partition that has enough
    free disk space for installation

Which way do you want ? [1] :
```

Do one of the following:

- Press Return or enter **1** to create the `/usr/i18n` directory. The following message is displayed:

```
Directory /usr/i18n has been created
```

- Enter **2** to create a symbolic link to another disk partition. The following message is displayed, and you specify the path of your choice:

```
You have chosen to make a symbolic link to another partition.
Please enter the installation path: /var/i18n

You have the following amount of free disk space available in
/usr/i18n :

$ df -k /usr/i18n
Filesystem      1024-blocks        Used   Available Capacity  Mounted on
usr_domain#usr      716800       238120      456512    35%     /usr
Do you want to continue this installation procedure? (y/n) [y]:y
```

The installation script lists the countries and languages that are available to install:

```
***********************************************************************
*                                                                     *
*   Tru64 UNIX WORLDWIDE LANGUAGE SUPPORT INSTALLATION PROCEDURE       *
*                                                                     *
***********************************************************************

Please select the countries for which you want to install support subsets:

  1) Belgium (French)                2) Canada (French)
  3) China (Hong Kong)               4) China (Simplified Chinese)
  5) China (Taiwan)                  6) Czech Republic
  7) Euro (Latin-9 & Unicode)        8) France
  9) Germany                        10) Greece
 11) Hungary                        12) Israel
 13) Italy                          14) Japan
 15) Korea                          16) Lithuania
 17) Poland                         18) Russia
 19) Slovakia                       20) Slovenia
 21) Spain (Catalan)                22) Spain (Spanish)
 23) Sweden                         24) Switzerland (French)
 25) Switzerland (German)           26) Thailand
 27) Turkey

 28) All of the above
 29) None of the above

Choices (for example, 1 2 3) :
```

If you enter more than one number at the prompt, separate each number with a space. After making your selections, a message similar to the following is displayed:

```
You are installing localized software for the following countries:

<list of countries>

Is this correct? [n]
```

To respond to the question:

- Enter **n** to display the software subset selection menu again to reselect countries.

- Enter **y** if you are satisfied with your selection.

Depending upon the countries you selected, the following questions may be asked:

- To install outline fonts:

```
Would you like to install outline fonts for printing
and display? [y] :
```

  Enter **y** to install outline fonts for better printing and displays. Outline fonts consume a considerable amount of disk space.

- To install public domain source files:

```
Would you like to install program sources of the public domain
software packaged in the Worldwide Language Support kit? [n] :
```

  Enter **y** to install program sources of the public domain software packaged in the worldwide language support software. The components NEMACS, MULE, and Wnn are sourced from public domain and are shipped with their source code because of GNU license guidelines. MULE is a multilingual enhancement of GNU Emacs and is based on GNU Emacs Version 19. Wnn is a public domain Japanese input method service provided for entering the Japanese language.

Section 2.5.1 describes the WLS software selection process; Section 2.5.2 describes the WLS software load process; and Section 2.5.3 describes the WLS software configuration process.

2.5.1 Selecting Worldwide Software Subsets

A menu of available software subsets is displayed. First on the menu is a list of the mandatory software subsets for the country or language you selected. These software subsets will be loaded automatically. The menu then displays a list of optional software subsets that you can install depending on which countries you have selected. If you specify more than one number at the prompt, separate each number with a space or a comma. Separate a range of numbers with a hyphen (–).

The following example shows the optional software that is available when Japan is chosen. The software subset list is similar to the following:

```
*** Enter subset selections ***

The following subsets are mandatory and will be installed automatically
unless you choose to exit without installing any subsets:

      * Japanese Standard Kernel Modules
      * Japanese CDE Mail Interface
      * Japanese Base System
      * Japanese Base System Management Applications and Utilities
```

```
        * Japanese Graphical Base System Management Utilities
        * Japanese Graphical System Administration Utilities
        * Japanese Basic X Environment
        * Japanese CDE Desktop Environment
        * Japanese CDE Minimum Runtime Environment
        * Japanese DECwindows 100dpi Fonts
        * Japanese DECwindows 75dpi Fonts
        * Japanese DECwindows Fonts
        * Japanese Netscape Communicator V4.7

The subsets listed below are optional:

    There may be more optional subsets than can be presented on a single
    screen. If this is the case, you can choose subsets screen by screen
    or all at once on the last screen. All of the choices you make will
    be collected for your confirmation before any subsets are installed.

 - Japanese Support - General Applications :
     1) Additional Japanese Software

--- MORE TO FOLLOW ---
Enter your choices or press RETURN to display the next screen.

Estimated free diskspace(MB) in root:53.1 usr:346.7

Choices (for example, 1 2 4-6):

     2) Japanese DOS tools
     3) Wnn Input Method

 - Japanese Support - Reference Pages :
     4) Japanese Ref. Pages for Additional Software
     5) Japanese Reference Pages
     6) Japanese Windows Reference Pages

 - Japanese Support - Software Development :
     7) Japanese CDE Software Development
     8) Japanese Ladebug Debugger Version 4.0
     9) Japanese Software Development
    10) Japanese Software Development Desktop Environment
    11) Japanese X Window Software Development
    12) Wnn Software Development

 - Japanese Support - System Administration :
    13) Japanese Advanced File System Graphical User Interface
    14) Japanese Logical Storage Manager GUI

 - Japanese Support - Windowing Environment :
    15) Japanese (SJIS) CDE Online Help
    16) Japanese CDE Online Help
    17) Japanese DECwindows Additional 100dpi Fonts
    18) Japanese DECwindows Additional 75dpi Fonts

 - Japanese Support - Windows Applications :
    19) Japanese Additional DECwindows Applications
    20) Japanese CDE Additional Applications

 - Worldwide Language Support - General Applications :
    21) Worldwide MULE


 - Worldwide Language Support - Obsolete Components :
    22) Worldwide Obsolete Commands and Utilities
```

```
  - Worldwide Language Support - Operating System :
    23) Worldwide European Unicode Locales

    24) Worldwide Phrase Input Support

    25) Worldwide User Defined Character Support

  - Worldwide Language Support - Software Development :
    26) Worldwide SVE MNLS Migration Tools
    27) Worldwide Software Development
    28) Worldwide X Window Software Development

  - Worldwide Language Support - Windowing Environment :
    29) Worldwide Composite Unicode Fonts
    30) Worldwide DECwindows Additional Fonts
    31) Worldwide Two-Byte Outline Font Renderer
    32) Worldwide User Defined Character Workstation Service
    33) Worldwide X/Motif 1.1

--- MORE TO FOLLOW ---
Add to your choices or press RETURN to display the next screen.

Estimated free diskspace(MB) in root:53.1 usr:337.0

Choices (for example, 1 2 4-6):  3

The following choices override your previous selections:

    34) ALL mandatory and all optional subsets
    35) MANDATORY subsets only
    36) CANCEL selections and redisplay menus
    37) EXIT without installing any subsets

Estimated free diskspace(MB) in root:53.1 usr:337.0

Add to your choices, choose an overriding action or
press RETURN to confirm previous selections.

Choices (for example, 1 2 4-6):  3
```

You have a chance to verify your choices as shown in the following example:

```
You are installing the following mandatory subsets:

        Japanese Standard Kernel Modules
        Japanese CDE Mail Interface
        Japanese Base System
        Japanese Base System Management Applications and Utilities
        Japanese Graphical Base System Management Utilities
        Japanese Graphical System Administration Utilities
        Japanese Basic X Environment
        Japanese CDE Desktop Environment
        Japanese CDE Minimum Runtime Environment
        Japanese DECwindows 100dpi Fonts
        Japanese DECwindows 75dpi Fonts
        Japanese DECwindows Fonts
        Japanese Netscape Communicator V4.7

You are installing the following optional subsets:

  - Japanese Support - General Applications :
        Wnn Input Method
```

```
Estimated free diskspace(MB) in root:53.1 usr:337.0

Is this correct? (y/n): y
```

Enter **n** if you want to begin the software subset selection again. Enter **y** if the list is correct.

Section 2.5.2 describes the WLS software load process.

2.5.2 Loading Worldwide Software Subsets

The installation process verifies that there is enough disk space to load the selected subsets. A message similar to the following is displayed:

```
Checking file system space required to install selected subsets:
        Working....Fri Aug 24 13:47:50 EST 2001

File system space checked OK.
```

If there is not enough disk space to hold all the software subsets you selected, go back and select fewer optional software subsets.

Next, the installation process installs the software subsets on your system. Messages similar to the following are displayed:

```
14 subsets will be installed.

Loading subset 1 of 14 ...

Japanese Base System
   Copying from system9 (inet)
        Working....Fri Aug 24 13:49:58 EST 2001
   Verifying

Loading subset 2 of 14 ...

Wnn Input Method
   Copying from system9 (inet)
        Working....Fri Aug 24 13:50:30 EST 2001
   Verifying

Loading subset 3 of 14 ...

Japanese Basic X Environment
   Copying from system9 (inet)
   Verifying
  .
  .

Loading subset 12 of 14 ...

Japanese Base System Management Applications and Utilities
   Copying from system9 (inet)
   Verifying

Loading subset 13 of 14 ...

Japanese CDE Mail Interface
   Copying from system9 (inet)
   Verifying

Loading subset 14 of 14 ...
```

```
Japanese Standard Kernel Modules
   Copying from system9 (inet)
   Verifying

14 of 14 subsets installed successfully.
```

Section 2.5.3 describes the WLS software configuration process.

2.5.3 Configuring Worldwide Software Subsets

Subset configuration occurs next, which is the process of tailoring
the operating system software for use. Review the screen output
carefully; depending upon the software subsets you installed, you may
be instructed to run setup scripts. If you performed the installation
from CD–ROM, after the installation is complete, review the log file,
`/var/adm/smlogs/wwinstall.log` file for a record of the installation. A
log file is not created during RIS installations.

```
Configuring "Japanese Base System" (IOSJPBASE540)
Configuring "Wnn Input Method" (IOSJPWNN540)
Configuring "Japanese Basic X Environment" (IOSJPX11540)
  .
  .
  .
Configuring "Japanese CDE Mail Interface" (IOSJPCDEMAIL540)
Configuring "Japanese Standard Kernel Modules" (IOSJPBIN540)
```

Section 2.5.4 describes the kernel build process.

2.5.4 Building the Kernel

If necessary, a kernel build begins automatically after software subset
configuration.

_____ **Note** _____

If you performed a dataless installation, the kernel build does not
happen automatically. Use the `doconfig` command to build the
kernel manually. See `doconfig`(8) for more information.

_____

The messages displayed by the kernel build are similar to the following.
Please read this information carefully because it may require user action to
restart the X Server to include the language option on the first login screen:

```
***********************************************************************

   Reconfiguring kernel to incorporate Asian/Thai tty drivers
***********************************************************************

 **** Adding Worldwide Support tty Features into Kernel Configuration File ****

Loading I18N tty kernel modules ... done
```

```
The installation software has completed the installation process.
The log file /var/adm/smlogs/wwinstall.log contains a record of
your installation.

Total installation time = 23 minutes 35 seconds.

Do you want to start i18nconfig at this time? (y/n) [y] : y
```

If the kernel build fails, check the log file at `/var/adm/smlogs/it.log`
for information to diagnose the problem. Section 2.5.5 describes the i18n
configuration utility, which you have the option to use at this point.

2.5.5 Using the i18n Configuration Application

After the kernel build is complete, you have the option to use the
`i18nconfig` application. The `i18nconfig` application provides the ability
to set up and maintain the worldwide support options installed on the system
including drivers, fonts, Wnn character-cell input methods for the Japanese
language, user accounts, and to view installed keymaps and locales.
Configuration of these options establishes an operating system environment
for writing and using internationalized applications. Figure 2–1 shows the
main window for the application:

Figure 2–1: i18n Configuration Application
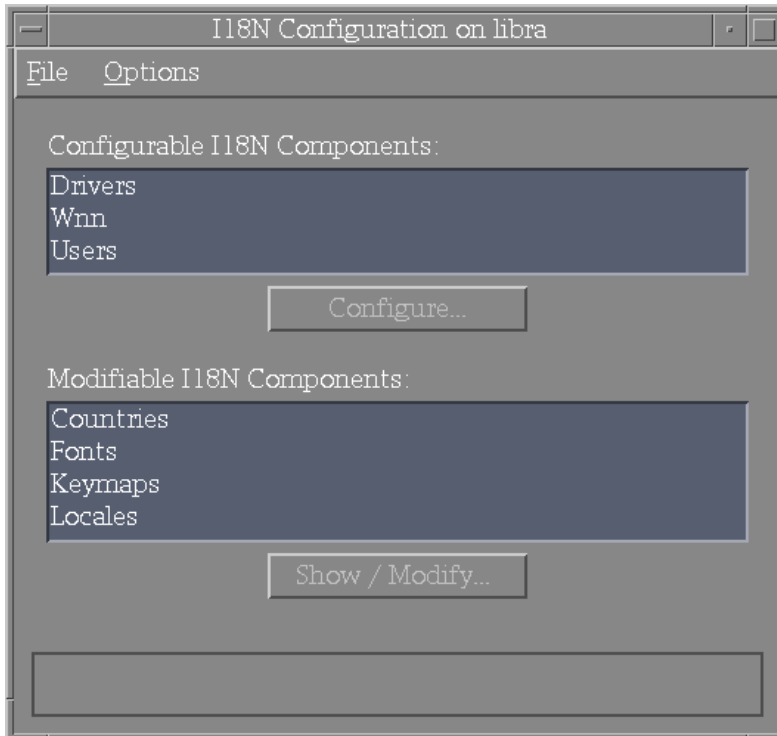
2.6 Optional Step: Building an Asian Kernel After the Installation

If you installed support software subsets for Japan, China, Hong Kong, Korea, Taiwan, or Thailand, the worldwide installation process builds a kernel with all the installed Asian or Thai terminal supports. Afterwards, you reboot the system with the new kernel to enable Asian or Thai terminal support in the kernel.

If you want to enable or disable some of the Asian or Thai terminal supports from the kernel, Section 2.6.1 describes the procedure to rebuild an Asian kernel.

2.6.1 Reconfiguring the Kernel to Support the Asian Terminal Driver and Daemons

To reconfigure the kernel to support the Asian terminal driver and daemons, invoke the wwconfig script with the -a option:

```
# /usr/sbin/wwconfig -a
```

See `wwconfig`(8) for more information.

If you installed `IOSWWBIN540` and installed at least one of the following subsets: `IOSWWUDCOS540` (on-demand font loading), `IOSWWPHRASE540` (software input method), or `IOSJPBASE540` (kana-kanji conversion), a UTX configuration selection table similar to the following is displayed. UTX is the mechanism to support communication between the Asian terminal driver and daemons.

```
*** UTX  CONFIGURATION SELECTION ***

    Selection   Asian service Daemon
---------------------------------------------------------
      1         On Demand Font Loading (odld)
      2         Software Input Method  (simd)
      3         Kana-Kanji Conversion  (kkcd)
      4         All of the above
      5         None of the above
---------------------------------------------------------

Enter the selection number for each daemon you want.
For example, 1 2 :
```

After you make your selection, the daemons are displayed for your confirmation. If you choose 4 `(All of the above)`, the following confirmation message is displayed:

```
You specified the following daemons:

    On Demand Font Loading (odld)
    Software Input Method  (simd)
    Kana-Kanji Conversion  (kkcd)

Is this correct? (y/n) [n]:
```

Enter `y` if the list includes the daemons you want to set up.

The installation procedure then asks how many UTX devices you want to create.

```
How many UTX devices do you want to create? [default: 32] :
```

The number you enter is saved in the `/var/i18n/sys/stanza.loadable` file. The actual creation of the UTX devices occurs when you reboot your system.

There is one utxd master daemon that uses one UTX device. Each invocation of one of the `odld`, `simd`, and `kkcd` daemons uses one UTX device. Each user who turns on `odld` on a database not already served by another `odld` starts a new `odld` process. See `stty`(1) and `cedit`(1) for more information. Each user session that has the Software Phrase Input Method turned on requires

one `simd`. Each user who turns on Kana-Kanji Conversion on a database (see `stty(1)` for more information) not already served by another `kkcd` starts a new `kkcd` process. For example, a system needs 31 UTX devices to support all three services for each of 10 users.

If you installed the `IOSZHBIG5540`, `IOSZHTELEX540`, and `IOSZHCONV540` software subsets, the following menu is displayed:

```
*** ADDITIONAL TERMINAL CODESETS SELECTION ***

    Selection    Terminal Codeset
---------------------------------------------------------------
       1         BIG-5
       2         Telecode
       3         Traditional & Simplified Chinese Conversion
       4         All of the above
       5         None of the above
---------------------------------------------------------------

Enter the selection number for each codeset you want.
For example, 1 2 :
```

Selecting a terminal codeset at this prompt means that you want to build support for that terminal codeset into the terminal driver. When codeset support is built into the terminal driver, users can select that codeset as their terminal code by using the `/usr/i18n/bin/stty` command.

Choose `3` if you want to support the proper codeset conversion when the terminal code is set to a Simplified Chinese codeset and the application code is set to a Traditional Chinese codeset. The reverse is also true.

If only two out of the three software subsets are installed, the selection menu is displayed but the missing component is not in the list.

If you installed just one of the software subsets, a question is asked instead. As shown in the following example, if you installed the `IOSTHBIN540` software subset, the procedure asks if you want to add the Thai terminal driver to the kernel:

```
Do you want to install the Thai tty driver? (y/n) [y] :
```

The Thai terminal driver supports Thai terminal input/output (I/O). The other Asian languages are supported by the Asian terminal driver. If you have installed only the `IOSTHBIN540` software subset and not the `IOSWWBIN540` software subset, the previous question is the only question asked.

The installation procedure then asks if you want to rebuild the kernel.

```
If you wish, you may use an automated kernel build procedure by
answering 'y' to the next question.
```

```
You will need about 10 MB available in the /sys file system
for the kernel build.  If you do not have this much space,
do not choose an automated build.

You have the following amount of free disk space available:

df -k /sys
Filesystem    kbytes       used      avail capacity  Mounted on
/dev/disk/dsk3g    825507    670890     72066    90%    /ufs/dsk3g
Do you want this procedure to rebuild your kernel? (y/n):
```

If you enter y, the kernel build starts, and messages similar to the following
are displayed:

```
Starting kernel rebuild...

*** KERNEL CONFIGURATION AND BUILD PROCEDURE ***

Saving /sys/conf/SYSTEM9 as /sys/conf/SYSTEM9.bck

Do you want to edit the configuration file? (y/n) [n]: n


*** PERFORMING KERNEL BUILD ***
      Working....Fri Aug 24 13:54:25 EST 2001
      Working....Fri Aug 24 13:56:25 EST 2001
      Working....Fri Aug 24 13:58:25 EST 2001

The new kernel is /sys/SYSTEM9/vmunix

Saving /vmunix as /vmunix.IOS540.3
Copying /sys/SYSTEM9/vmunix to /vmunix
```

In the previous example, SYSTEM9 is the system name.

_____ **Note** _____

You can invoke wwconfig with the -s flag to build a statically
linked kernel. In that case, the output from the wwconfig
command is different from what is shown in this section.

Whenever you want to enable or disable some of the terminal options, you
must reconfigure and rebuild the kernel using the following command:

/usr/sbin/wwconfig -a

2.7 Backup Files Created by the WLS Installation

During the installation of worldwide support software subsets, some backup
files are created to save the contents of the original files that are replaced
by the installation procedure. Table 2–1 lists the files replaced by the
installation procedure.

The backup files have either the extension .IOS540_sav.* where the
asterisk (*) is an integer, or have the extension .IOS540_sav (without
the integer).

Table 2–1: Backup Files Created by WLS Installations

Files Saved with Extension	File Name
`.<file>.IOS540_sav`	`/usr/bin/X11/dxkeycaps`

Backup files with extension `IOS540_sav` are data or binary files that are not likely to be modified by the system manager. They are restored to the original files when worldwide support software subsets are removed.

Caution

Do not delete files with the `IOS540_sav` extension. If you delete these files, the corresponding data and binary files cannot be restored during the removal of WLS software.

Data files that might be modified by the system managers, depending on the system configuration, have the extension `.IOS540_sav.*` as their backup. You can delete files with the extension `.IOS540_sav.*` to save space. These files are not restored to the originals during subset removal.

2.8 Step 5: Setting Environment Variables

When one language (excluding Euro Latin-9 & Unicode) is installed from the WLS kit for the first time, the CDE desktop starts up in that language by default. However, if more than one language is selected for the first time, the CDE desktop remains in English. If the CDE desktop has been switched to another language, installing more languages at a later time will not affect the default CDE language.

You can reset the default CDE language to English by using the following command:

```
# rm -f /etc/dt/config/Xconfig
```

If you installed support for more than one language, you set the locale by defining the `LANG` or `LC_ALL` environment variables.

To set the language for the common desktop environment (CDE) from the CDE login window each time you log in, click on `Options`, click on `Language`, and select the language you want to run.

2.9 Related Documentation

There are several pieces of documentation that provide more information for your internationalized operating system:

- The *Using International Software* manual describes how to use internationalized software and work in a multi-language environment

- The *Writing Software for the International Market* manual describes how to create international software with the tools provided with the operating system

- Several reference pages are also available: `locale`(1), `i18n_intro`(5), `l10n_intro`(5), and the reference pages for individual languages (such as `spanish`(5), `italian`(5), `japanese`(5), `hebrew`(5) and so on)

3

The UNIX Shell

This chapter contains the following information:

- Description of the UNIX shell (Section 3.1)

- How to invoke the UNIX shell from the Full Installation (Section 3.2)

- Samples of the types of tasks that can be performed from the UNIX shell environment (Section 3.3)

- How to use the UNIX shell to:

    - Create swap space (Section 3.4)

    - Mount file systems (Section 3.5)

    - Restore UNIX file systems (UFS) or Advanced File Systems (AdvFS) from tape backups (Section 3.6)

    - Restore the system image from a damaged disk to a new disk and assign the old device name to the new device (Section 3.7)

    - Change the size of a disk partition using the `disklabel` command (Section 3.8)

- Returning to the Full Installation procedure from the UNIX shell environment (Section 3.9)

3.1 Overview

The UNIX shell option is a way to access standard UNIX commands during a Full Installation. The primary purpose of the UNIX shell option is to provide a way to perform disk and file system maintenance before or during a Full Installation. The UNIX shell provides a way to access all UNIX commands that help you recover from serious problems such as / (root) file system corruption and enables you to perform general file system and disk maintenance tasks. It is usually not necessary to access the UNIX shell during an installation, but the option to do so is offered just in case you need it.

The base operating system distribution media (CD-ROM or RIS) contains file systems that are laid out just as the software would be installed on the system and provides direct access to the /, /usr, and /var directories. The distribution media contains a mixture of compressed and uncompressed software subsets. This format makes all commands and utilities in

the uncompressed software subsets available in the UNIX shell even if
your operating system is not yet fully functional. In effect, the mounted
installation media is almost a full operating system environment. The UNIX
shell option enables you to use the installation media as a tool for disaster
recovery.

You should perform system management activities in the UNIX shell only if
you have previous UNIX operating system experience.

3.2 Invoking the UNIX Shell

How you invoke the UNIX shell from the installation procedure depends
upon whether you are using the graphical or text-based interface. When you
invoke the UNIX shell option, the system is in a Bourne shell in single-user
mode with superuser privileges. See the *Command and Shell User's Guide*
for more information about shells and privileges.

After you boot your system from the distribution media to start a Full
Installation, you will see an Installation window on systems capable of
graphical display. To invoke the UNIX shell, select the Shell Window menu
item from the File menu at the top of the dialog box.

On systems that do not have graphics capabilities, there are two ways to
invoke the UNIX shell from the text-based Full Installation interface. The
first is to choose menu option 3 from the first screen of the text-based
interface. The second method is to press Ctrl/c (at any time except during
software subset loading).

_____ **Note** _____

Any installation selections you have made before invoking the
UNIX shell from the text-based interface are lost. Follow this
procedure to restart the text-based installation and start the
selection process again:

```
# cd /
# restart
```

_____

3.3 UNIX Shell Capabilities

Using the UNIX shell in the installation environment provides most of
the capabilities of a full operating system environment. The difference
between using the UNIX shell in the installation environment and a normal
operating system environment is that it works without a swap device and
with very limited capacity within the memory file systems (MFS) that are
available during a Full Installation. These two factors mean that tasks
requiring large amounts of memory that create the need to swap, or tasks

requiring large amounts of disk space (such as /tmp space), are likely to encounter failures in the installation environment.

_____ **Note** _____

The / (root) file system is located on the CD-ROM or the RIS server, and after the installation distribution media is booted, it is mounted with read-only permissions. The memory file systems (MFS) at /var, /dev, and /devices are mounted with read-write permissions. Any changes you make to files in /var, /dev, and /devices are volatile and will not be saved when you halt the installation and return to the console prompt (>>>).

_____

Here is a sample of the types of tasks you can perform from the UNIX shell; some of these tasks are documented in this chapter:

- Create new file systems with the newfs command for UNIX File Systems (UFS) or with the mkfdmn and mkfset commands for Advanced File Systems (AdvFS).

- Restore file systems with the restore command (for UFS) or the vrestore command (for AdvFS).

- Modify disk partition tables with the disklabel command on systems that do not have graphics capabilities. Otherwise, use the graphical Disk Configuration Utility, which is accessed either by clicking on the Edit Partitions... button on the File System Layout window, or by invoking the /usr/sbin/diskconfig utility at the shell prompt.

- Mount disks and file systems with the mount command.

- Fix UFS file systems with the fsck command. The fsck command is not required for AdvFS file systems.

- View a file with the ed text editor. By default, the EDITOR environment variable is set to ed. To enable the vi text editor:

  - On older systems with console firmware that supports curses capability:

    ```
    # TERM=vt100[1]
    # export TERM
    ```

  - On newer systems with personal computer-like graphics:

    ```
    # TERM=pccons
    # export TERM
    ```

_____

[1] Substitute vt100 if you have a different video terminal type, for instance vt302.

3.4 Creating Swap Space

If you need to perform a task that requires swap space, you can turn
swapping on in the UNIX shell. The only suggestion for turning on swap
space in the UNIX shell environment is that you should not use a partition
that contains data you want to preserve because it will be overwritten. To
avoid this problem, use a disk partition that previously has been used for
swap space.

Follow this procedure to turn on swapping in the UNIX shell:

1. Decide which device (that is, the device name and partition) you want
    to use for the swap area. Be careful not to choose an area that has
    data that you want to preserve.

    _____ **Note** _____

    A disk partition that previously was used for swap space
    will be labeled as `swap` in the disklabel and can be used
    without harm.
    _____

2. Change to the `/dev/disk` directory:

    ```
    # cd /dev/disk
    ```

3. Turn the swap device on. In this example, the swap device is `dsk0b`:

    ```
    # swapon dsk0b
    ```

4. Verify that the swap device is turned on:

    ```
    # swapon -s
    ```

    _____ **Note** _____

    The `restart` command, which is used to restart the installation
    from the UNIX shell, is no longer available after swap space has
    been enabled. To restart the installation, halt the system and
    reboot from the distribution media.
    _____

3.5 Mounting File Systems

The UNIX shell can be used to perform maintenance operations on existing
file systems. For instance, if the kernel (the file called `vmunix`) in your `/`
(root) file system becomes damaged and you have a good backup copy of the
kernel, you can mount your `/` file system and replace the damaged kernel. If
you are using LSM volumes for the `/` file system, see the *Installation Guide*
for information about how to restart LSM.

Follow this procedure to mount an existing root file system located on /dev/disk/dsk0a to another mount point.

1. Create a mount point in /var:

   # **mkdir /var/mnt**

2. Mount the file system:

   a. For UNIX file systems (UFS), enter:

      # **fsck -y /dev/disk/dsk0a**
      # **mount /dev/disk/dsk0a /var/mnt**

   b. For Advanced File Systems (AdvFS), enter:

      # **mkdir -p /etc/fdmns/root_domain**
      # **cd /etc/fdmns/root_domain**
      # **ln -s /dev/disk/dsk0a dsk0a**
      # **mount root_domain#root /var/mnt**

The existing / file system is accessible at /var/mnt and can be modified at this point.

3.6 Restoring File Systems from Backup

The UNIX shell is ideal for restoring damaged / (root) file systems, but you can use the UNIX shell to restore file systems other than the / file system.

It is recommended to restore file systems from a full operating system environment. If such an environment is unavailable due to the need to restore either /var or /usr, boot your system to single-user mode by using your existing or restored / file system to provide more writable disk space than is available in the installation environment. Swap space can be made available in the UNIX shell as shown in the instructions in Section 3.4.

Section 3.6.1 describes how to restore UFS file systems, and Section 3.6.2 describes how to restore AdvFS file systems. Both sets of procedures assume that you are restoring the file system to the same disk.

3.6.1 Restoring UNIX File Systems (UFS) from Tape Backup

Use the following procedure to restore a UNIX file system to the same disk on which it originally resided. The size of the partition you are restoring must be larger than the size of the dump file.

_____ **Note** _____

These instructions assume that your system has been booted
from the distribution media, and you have exited the installation
procedure from the File menu to the UNIX shell.

_____

1. Read the disk label on the disk with the corrupted file system to
   determine the disk type:

   ```
   # disklabel -r /dev/disk/dskna | more
   ```

   The disk type is shown next to the `disk:` field. Record the disk type
   here: _____

   If the disk does not have a disk label, proceed to Section 3.8 to create one.

2. Create a new / (root) file system by using the following command syntax:

   newfs *device_name disk_type*

   The `device_name` parameter specifies the full device pathname of the
   affected disk. The / (root) file system must reside on the `a` partition.
   As an example, to create a new file system on an RZ58 type disk on
   partition `a`, enter the following command:

   ```
   # newfs /dev/disk/dsk0a rz58
   ```

3. Create a mount point in `/var` to prepare to mount the file system :

   ```
   # mkdir /var/mnt
   ```

   Create mount points under the `/var` or `/tmp` directories because these
   are the only directories that are writable from the UNIX shell.

4. Mount the damaged file system by using the following command syntax:

   mount *device_name mount_point*

   The `device_name` parameter specifies the full device pathname of
   the disk device. For example, to mount the file system created in the
   previous step, enter the following command:

   ```
   # mount /dev/disk/dsk0a /var/mnt
   ```

5. Verify that the tape device is powered up and is connected to the system.

   _____ **Note** _____

   If the tape device was not powered up when the system
   initialized, it may not be visible to the system. In that case,
   shut down the system, power up the tape device, and reboot
   the system.

   _____

6. Determine whether or not a tape device special file exists. Use the following command to obtain information about all devices on the system:

   # **hwmgr -view device**

   If the output from this command does not indicate the presence of a device name similar to /dev/tape/tape0, create the tape device special file as described in Step 7. If a tape device is present, proceed to Step 9.

7. Create a device special file for the tape device:

   # **/sbin/dn_setup -install_tape**

   You may see messages such as +tape0 that list the device names that are created during this operation. The plus sign indicates that the device is being added.

8. Use the hwmgr command again to see the full path to the tape device name. You will use this device name in Step 9.

   # **hwmgr -view device**

9. Restore the root file system. If you are restoring dump files from a local file system, change to the /var/mnt directory, insert the medium containing the dump file, and enter the restore command with the following command syntax:

   restore -rf *tape_device*

   The *tape_device* parameter specifies the full path to the tape device containing the dump data. For example, enter the following commands:

   # **cd /var/mnt**
   # **restore -rf /dev/tape/*tape_device***

10. Shut down and halt the system:

    # **shutdown -h now**

11. Reboot the system from the restored disk (use the show dev command if you do not know the console device name associated with the restored disk):

    # **show dev**
    # **boot *console_device_name***

3.6.2 Restoring Advanced File Systems (AdvFS) from Tape Backup

Use the following procedure to restore an AdvFS file system to the same disk on which it originally resided.

_____ **Note** _____

These instructions assume that your system has been booted
from the distribution media, and you have exited the installation
procedure to the UNIX shell.

_____

1. Read the disk label of the disk with the corrupted file system to make
    sure that it has a valid disk label:

    # **/usr/sbin/disklabel -r /dev/disk/dsk***na* **| more**

    If the disk does not have a disk label, proceed to Section 3.8 to create one.

2. Create a new / (root) file domain by using the following command
    syntax:

    mkfdmn -orF *device_name domain*

    The *device_name* parameter specifies the full device pathname of the
    disk device on your system. For example, to create a new file system on
    partition a on dsk0, enter the following command:

    # **mkfdmn -orF /dev/disk/dsk0a root_domain**

3. Create a root fileset in the root_domain file:

    # **mkfset** *domain fileset*

    The *domain* parameter specifies the name of the domain in which to
    create the fileset. For example:

    # **mkfset root_domain root**

4. Create a mount point in /var or /tmp to prepare to mount the fileset:

    # **mkdir /var/mnt**

    It is recommended that you create mount points under the /var or
    /tmp directories because these are the only directories that are writable
    from the UNIX shell.

5. Mount the root fileset by using the following command syntax:

    mount *domain#fileset* mount_point

    The *domain#fileset* parameter specifies the root file domain and the
    root fileset. For example, to mount the fileset created in the previous
    steps, enter the following command:

    # **mount root_domain#root /var/mnt**

6. Verify that the file domain and fileset are indeed mounted:

    # **showfdmn root_domain**
    # **showfsets root_domain**

7. Verify that the tape device is powered up and is connected to the system.

_____ **Note** _____

> If the tape device was not powered up when the system
> initialized, it may not be visible to the system. In that case,
> shut down the system, power up the tape device, and reboot
> the system.

---

8. Determine whether or not a tape device special file exists. Use the
    following command to obtain information about all devices on the
    system:

    `# `**`hwmgr -view device`**

    If the output from this command does not indicate the presence of a
    device similar to `/dev/tape/tape0_d0`, create the tape device special
    file as described in Step 9. Otherwise, skip to Step 11.

9. Create a device special file for the tape device:

    `# `**`/sbin/dn_setup -install_tape`**

    Messages such as `+tape0` list the device names that are created during
    this operation. The plus sign indicates that the device is being added.
    When all tape device special files are created, proceed to Step 11 to
    continue with the recovery of the damaged `/` file system.

10. Use the `hwmgr` command again to see the full path to the tape device
    name. You will enter this device name in Step 11.

    `# `**`hwmgr -view device`**

11. Restore the fileset using the `vrestore` command. To restore files
    from a local file system, change to the `/var/mnt` directory, insert the
    medium containing the dump file, and enter the `vrestore` command
    using the following syntax:

    vrestore -vxf *tape_device_name*

    The `tape_device_name` parameter obtained in Step 10 specifies the
    pathname of the tape device containing the dump data.

    `# `**`cd /var/mnt`**
    `# `**`vrestore -vxf /dev/tape/`***`tape_device_name`***

    _____ **Note** _____

    > You can restore a UFS format dump tape to AdvFS (for
    > instance if you are converting a UFS `/` file system to
    > AdvFS) and you can make a `vdump` tape on UFS. The
    > restore command you use depends on the format of the
    > tape (`dump` or `vdump`). Use `vrestore` to restore AdvFS
    > dumps performed with the `vdump` command and `restore`
    > to restore UFS dumps performed with the `dump` command.

The corresponding restore command is used regardless of the target file system type.

12. Verify the contents of the /etc/fstab file and the /etc/fdmns directory. The mkfdmn command added /etc/fdmns/root_domain to the temporary / file system in the UNIX shell, which is deleted when you reboot the system.

13. Shut down and halt the system:

    # **shutdown -h now**

14. Reboot the system from the restored disk (use the show dev command if you do not know the console device name associated with the restored disk):

    # **show dev**
    # **boot** *console_device_name*

You can use the UNIX shell to restore other file systems, but it is recommended that you perform file system restores from a full operating system environment. If such an environment is unavailable due to the need to restore either /var or /usr, you should boot your system to single-user mode by using your existing or restored / file system. In the single-user mode, more disk space is available, and swap space can be made available by issuing the following command:

    # **swapon -a**

3.7 Restoring the Root File System Image from a Damaged Disk to a New Disk

This procedure describes how to restore a saved root file system image from a bad or damaged disk to a new disk and then rename the new disk with the old system disk name.

The process of introducing a new system disk is more complicated than just updating the /etc/fstab file because the installation kernel may assign a different device name to the devices on the system than the ones that were assigned on the saved system image. As a result, the restored system image will disagree with the kernel used to restore the image, and you cannot assume that it is possible to determine what the mapping is between them.

The following procedure assumes you have restored the root file
system image to a new disk as described in Section 3.6 .

_____

1. Halt the system:

    # **shutdown -h now**

2. Determine the console disk device names on the system. The console
    device name usually is shown in the middle column of the command
    output. Console device names are similar to DKA100, DKC0, and so on.

    >>> **show dev**

    Record the console device name of the damaged disk here: _____
    and the console device name of the new disk here: _____

3. Boot the system to single user mode using the console device name of
    the new disk (obtained in Step 2) with the restored image :

    >>> **boot -fl s** *console_device_name*

4. In single user mode, mount the / file system as writable:

    # **/sbin/mountroot**

    This process creates a device entry for the newly found disk. You will
    see several processing messages ending with output that is similar
    to the following:

    dsk*N*a dsk*N*b dsk*N*c ...

    In the previous example, *N* is the disk number assigned to the new disk
    that is being seen for the first time and should be the disk that was
    restored. However, if more than one disk is shown in the output, you
    have added multiple disks since the system disk was last saved.

    Record the new disk name here: _____

    Proceed to Step 5 to determine the disk name associated with the old
    damaged disk.

5. Determine the old boot disk name:

    # **/sbin/consvar -g bootdef_dev**

    Output from this command is similar to the following.

    bootdef_dev = dsk*N*

    In the previous command output, dsk*N* is the *old_disk_name*.

    Record the old disk name here: _____

6. Exchange the old disk name with the new disk name:

    # **dsfmgr -e** *new_disk_name old_disk_name*

In the command line, substitute *new_disk_name* with the disk name
obtained in Step 4, and substitute *old_disk_name* with the disk name
returned in Step 5.

_____ **Note** _____

A message is displayed if the old, damaged disk has been
removed and is not active. In that situation, use the move
flag (-m) instead:

dsfmgr -m *new_disk_name old_disk_name*

_____

7. Reset the boot device:

    # **/sbin/consvar -s bootdef_dev** *new_disk_name*

8. Optionally perform additional maintenance tasks, or if you are finished,
    boot the system to multiuser mode:

    # **init 3**

3.8 Changing the Size of a Disk Partition Using the disklabel Command

If you are using the text-based installation interface, use the disklabel
command to change the drive identification or the disk partitions on the
drive or to replace a damaged label or bootstrap. See disklabel(8) for more
information.

To look at the existing disk partition layout, enter the disklabel command
in the following format and replace the variable *n* with the unit number of
the disk. For example, to look at the existing disk partition layout of disk,
enter the following command:

disklabel -r /dev/disk/dsk*n*

The following example uses an rz26l disk. In this example, the size of the
b partition is decreased, and the size of the a partition is increased to 128
MB, which is the minimum size to hold the / (root) file system. Follow this
procedure to change the size of disk partitions.

1. Read the disk label on dsk0:

    # **disklabel -r dsk0**

2. Set the EDITOR environment variable to use the ed editor:

    # **EDITOR=ed**
    # **export EDITOR**

_____ **Note** _____

If you have a VGA monitor and want to use the vi editor, you first have to set the following variables:

```
# TERM=vt100
# export TERM
# EDITOR=vi
# export EDITOR
```

Then, use the corresponding vi commands instead of the commands shown in this procedure to make the disk label changes. The rest of this procedure uses the ed text editor.

_____

3. Copy the disk label for dsk0 to a temporary file:

   ```
   # disklabel -r dsk0 > /tmp/old_disklabel
   ```

4. Edit the copy of the disk label to protect against introducing errors on the real disk label:

   ```
   # ed /tmp/old_disklabel
   ```

5. Display the disk label:

   ```
   1,$p
   ```

6. Search for the b partition:

   ```
   /b:
   ```

   Information similar to the following is displayed:

   ```
   b:  262144  131072  unused  1024  8192  #  (Cyl. 164*- 492*)
   ```

7. Change the size of the b partition from 262144 sectors to 131072 sectors :

   ```
   s/262144/131072/p
   ```

   This reduces the size of the b partition from 128 MB to 64 MB. The revised information is displayed:

   ```
   b:  131072  131072  unused  1024  8192  #  (Cyl. 164*- 402)
   ```

   There is no need to modify cylinder information; cylinder information automatically is modified when you save and exit the file.

8. Change the offset of the b partition from 131072 sectors to 262144 sectors :

   ```
   s/131073/262144/2p
   ```

   This changes the offset of the b partition to start at position 262144. The revised information is:

   ```
   b:  131072  262144  unused  1024  8192  #  (Cyl. 164*- 402)
   ```

   There is no need to modify cylinder information; cylinder information automatically is modified when you save and exit the file.

9. Search for the `a` partition:

   /a:

   Information similar to the following is displayed:

   ```
   a:   131072  0    unused   0    0   # (Cyl. 0 -164*)
   ```

10. Because the size of the `b` partition was reduced by 131072 sectors, the size of the `a` partition should be increased by 131072 sectors. Change the size of the `a` partition from 131072 sectors to 262144 sectors:

    s/131072/262144/p

    This increases the size of the `a` partition from 64 MB to 128 MB.

11. Display the disk label again to verify your changes:

    1,$p

12. Save your edits and quit the editor:

    wq

13. Apply the new disk label to the disk:

    # **disklabel -R dsk0 /tmp/old_label**

14. Display the newly customized disk label:

    # **disklabel -r dsk0**

15. Exit the shell or restart the Full Installation.

3.9 Returning to the Installation Procedure from the UNIX Shell

The procedure for returning to Full Installation from the UNIX shell depends upon whether you are using the text-based or graphical interface.

3.9.1 Text-Based Interface

If you entered the UNIX shell from the text-based interface or from the Quit button on the Summary window, enter the following commands to restart the Full Installation interface from the UNIX shell:

cd /
restart

If you have a system console with graphics capability and you want to restart the installation procedure with the text-based interface instead of the graphical user interface, enter the following command:

cd /
restart nogui

Any installation selections you made up to the point you exited to the UNIX shell are lost. When you restart the Full Installation, you have to make your selections again.

3.9.2 Graphical Interface

If you entered the UNIX shell from the File menu in the graphical interface, you can resume the Full Installation at any time just by clicking back in the Full Installation window. You do not have to exit the shell window; however, if you want to exit the shell window, enter `exit` at the # prompt within the shell window.

If you entered the UNIX shell from the Quit button on the Summary window, use the procedure shown in Section 3.9.1 to restart the Full Installation.

4

Disk Space Planning

This chapter contains information about the standard UNIX file systems and provides disk space planning guidelines should it be necessary for you to manually plan disk space. The Full Installation process has built–in automatic disk space planning features, so it is unlikely that you would have to manually plan disk space.

Topics in this chapter include:

- Overview of file systems and disk space (Section 4.1)

- Descriptions of the two supported file system types: AdvFS and UFS (Section 4.3)

- Where to find information about planning disk space for clusters (Section 4.2)

- Disk space planning that is done automatically for you by the Full Installation process (Section 4.4)

- The circumstances which may make it necessary for you to manually plan disk space (Section 4.5)

- Considerations for customizing disk partitions and file system layout (Section 4.6)

- How to determine existing disk and partition sizes (Section 4.7)

- How to determine the disk space needed to install the operating system (Section 4.8)

- The contents of the standard UNIX file systems (Section 4.9)

- Swap space allocation guidelines (Section 4.12)

_____ **Note** _____

See the *Glossary* in the *Installation Guide* for definitions of the terms used in this chapter.

_____

4.1 Overview

A Full Installation of the operating system creates several basic file systems: / (root), /usr, and one area for swap space. The var area can be a directory

under the `/usr` file system, or it can be created as its own file system. The Full Installation process is designed to simplify your decision making with respect to where to place these file systems on a disk and how big they have to be in order to install the operating system. This chapter describes the built-in disk planning features of the Full Installation and then gives you the information you need to know if your computing environment requires a more customized approach.

4.2 Disk Planning Considerations for Clusters

The information in this chapter applies to disk planning for single system installations. See the TruCluster Server *Cluster Installation* and *Cluster Administration* manuals for information about disk space planning considerations for cluster file systems, quorum disks, and general information about configuring systems as cluster members.

4.3 Overview of File System Types

This operating system supports two types of file systems, the Advanced File System (AdvFS) or the UNIX File System (UFS).

You can choose either AdvFS or UFS as the file system type for the `/`, `/usr`, `/var`, and `/usr/i18n` file systems. If you are customizing the file system layout, all file systems do not have to use the same file system type. If you are using the default file system layout, you only can use one file system type for all file systems.

The Logical Storage Manager (LSM) is available to install and configure for both file system types.

The information in this section is intended to aid you in making the decision with respect to which file system type to choose.

4.3.1 The Advanced File System (AdvFS)

AdvFS is the default file system type. The AdvFS file system is required for running a cluster configuration.

AdvFS is a log-based file system that provides flexibility, compatibility, data availability, high performance, and simplified system management. AdvFS takes advantage of the 64-bit computing environment and is designed to handle files and filesets as large as almost 16 terabytes.

The configuration of AdvFS differs from the traditional UNIX file system. In AdvFS, the physical storage layer is managed independently of the directory layer. System administrators can add and remove storage without unmounting the file system or halting the operating system. As a result, configuration planning is less complicated and more flexible.

From a user's perspective, AdvFS behaves like any other UNIX file system. End users can use the `mkdir` command to create new directories, the `cd` command to change directories, and the `ls` command to list directory contents. AdvFS logical structures, quota controls, and backup capabilities are based on traditional file system design.

Without taking an AdvFS system off line, system administrators can perform backups, reconfigure file systems, and tune file systems. End users can retrieve their own unintentionally deleted files from predefined trashcan directories or from clone filesets without assistance from system administrators.

AdvFS supports multivolume file systems, which enables file-level striping (spreading data to more than one volume) to improve file transfer rates for I/O intensive applications. Logical Storage Manager (LSM), which allows volume-level striping, can be incorporated into AdvFS configurations.

AdvFS Utilities, which are licensed separately from the operating system, provide additional file management capabilities and a graphical user interface (GUI) to simplify system administration. The AdvFS GUI, which runs under the Common Desktop Environment (CDE), features menus, graphical displays, and comprehensive online help that make it easy to perform AdvFS operations.

With the exception of the / file system, AdvFS file system size can be modified at any time (with the `addvol` command). Increases or decreases to file system size are transparent to the user.

If you plan to use AdvFS as the file system type and you install the optional AdvFS Utilities, which are available on a separate CD-ROM distribution and require a special license, modifying file system space is simplified. After the installation, the AdvFS utilities let you add or remove volumes from the AdvFS file systems with no changes to the directory structure and with no user interruption. There is no need to over allocate file system space for AdvFS file systems.

See the *AdvFS Administration* manual if you need more information about AdvFS.

4.3.2 The UNIX File System (UFS)

UFS is the older, more traditional file system type for UNIX operating systems. AdvFS, a log based file system, is more robust, stable, and less likely to corrupt data than UFS. AdvFS can have files from one file system on more than one disk or partition, but UFS has a more rigid hierarchy. In a UFS file system, each disk or disk partition contains one separate file system; all files in a file system are located on one disk in the assigned disk partition. The UFS file system is characterized by a hierarchical

structure, the ability to create and delete files, dynamic growth of files, and the protection of file data.

UFS is compatible with the Berkeley 4.3 Tahoe release. UFS allows a pathname component to be 255 bytes, with the fully qualified pathname length restriction of 1023 bytes. This implementation of UFS supports a maximum file size equivalent to the largest supported file system (128 GB).

_____ **Note** _____

The Logical Storage Manager (LSM) can be used with the UFS file system type. Clusters cannot use the UFS file system type.

_____

See the *System Administration* manual for more information about UFS.

4.4 Automatic Disk Space Planning Features of the Full Installation

The Full Installation process gives you the option to use a predefined default file system layout, which means:

- The basic UNIX file systems and one swap area are installed on one disk of your choice. This disk must be 1 gigabyte (GB) or greater in size.

- The file systems are installed as follows:

  - The / (root) file system is on the a partition.

  - The /usr file system is on the g partition and var is a directory in /usr. If AdvFS is chosen as the file system type, var is located in the usr_domain.

  - A single swap area is located on the b partition.

- The choice of AdvFS or UFS as the one file system type for all file systems. AdvFS is the default file system type.

- The recommended partition table creates disk partitions large enough for all file systems and base operating system software subsets. Partition size is calculated based upon the size of the disk as shown in Table 4–1.

Table 4–1: Recommended Partition Table Based on Disk Capacity

File System Name and Location	File System Size Based on Disk Capacity		
	1 GB Disk	2 GB Disk	3 GB or Larger Disk
/ (root) partition a	128 MB	256 MB	384 MB
/usr partition g	745 MB	1490 MB	2235 MB
swap space partition b	128 MB	256 MB[a]	384 MB[a]

[a] The swap space could be even greater depending on the size of the b partition in the default disk partition table

The design of the default file system layout along with the recommended disk partition table allows the entire operating system to fit on a single disk that is 1 GB or greater in size.
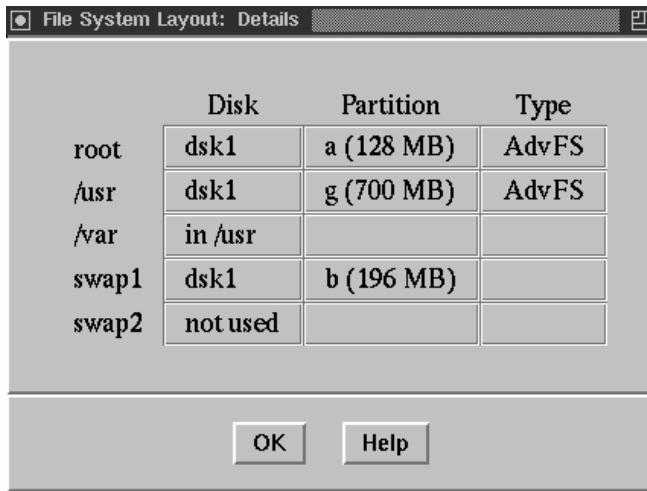
_____ **Note** _____

While possible, a Full Installation to a single 1 GB disk is not advisable if you have more than one disk. Apart from degraded performance, it is not possible to store crash dumps, and it is very difficult to recover the root disk in the event of a disk crash.

The recommended strategy is to place the / (root) file system on one disk and place the /usr file system and swap area on another disk. User data should be located on other disks.

If you are comfortable with the defaults and your system configuration does not have any special considerations, there is no reason to perform additional disk planning exercises. The details of a default file system layout are shown in Figure 4–1. This sample installation uses dsk1 as the single disk on which to install the operating system.

Figure 4–1: Default File System Layout: Details Window

	Disk	Partition	Type
root	dsk1	a (128 MB)	AdvFS
/usr	dsk1	g (700 MB)	AdvFS
/var	in /usr		
swap1	dsk1	b (196 MB)	
swap2	not used		

File System Layout: Details

OK   Help

Section 4.5 describes the situations when you should consider manually planning the file system layout and disk partition sizes.

4.5 When Should I Manually Plan Disk Space?

There are certain circumstances when you should manually plan disk space rather than accepting the defaults or recommendations.

It should only be necessary to plan disk space if any of the following is true:

- You plan to use the system as a remote installation services (RIS) server or as a dataless management server (DMS) server. These two utilities put essential data in the /var file system and as a result, it is recommended to place /var on its own partition, and perhaps its own disk. The RIS and DMS utilities are documented in the *Sharing Software on a Local Area Network* manual. Disk planning considerations for these two utilities are described briefly in Section 4.5.1 and Section 4.5.2.

- You want to install associated or layered products that have significant / (root) file system space requirements. The *Release Notes* contain software subset size information for all associated and layered products. It is recommended that you read the *Release Notes* to make sure there are no extraordinary disk space requirements associated with any additional software products.

- If you plan to install Worldwide Language Support (WLS) software subsets after the Full Installation as a separate installation task, you may need to size the /, /usr, and /var file systems accordingly and consider the size of the /usr/i18n directory. See the *Installation Guide* and the *Release Notes* for a description of software subset names, sizes,

and descriptions within each language group. Chapter 2 in this manual documents WLS installation procedures.

- You want to install the file systems over several disks.
- You want to create two swap areas instead of one, or you want to create swap areas that are larger than the default size.

4.5.1 Considerations for Remote Installation Services (RIS)

RIS is a utility that is used to set up a central server to serve software over the network to registered client systems. If you are planning to set up your system as a RIS server, you have the option to extract software subsets from the distribution media into the `/var/adm/ris` directory in the `var` area instead of using locally mounted media as the source of the RIS area. Extracting software takes up a considerable amount of disk space because you are copying the contents of the operating system CD-ROM onto the disk. You also may need to serve more than one version of the operating system, which requires the extraction and storage of yet more software subsets. One extracted RIS area for the base operating system requires approximately 600 MB of disk space. Disk space requirements for individual layered products are shown in the *Release Notes*.

Another option when using RIS is to create a symbolically linked RIS area. In this type of area, the total contents of the CD-ROM is not copied but is linked to the RIS area. The only disadvantage for a symbolically linked area is that the device holding the distribution media is now used exclusively for this purpose. As a guideline, you can assume a symbolically linked RIS area will take about 10 MB of space for the network bootable kernel and other support files.

You must reserve enough space in the `/var/adm/ris` directory in the `var` file system for the software you want to store in each RIS environment. How much space you reserve depends upon how many RIS environments you intend to create. See the *Installation Guide* for a description of each software subset and the names of other subsets or kernel configuration file options related to RIS operation.

4.5.2 Considerations for Dataless Management Services (DMS)

In a Dataless Management Services (DMS) environment, a server system maintains the `/` and `/usr` file systems for all client systems. The server maintains one copy of `/` for each client. The `/usr` file system is exported read-only and is shared by all clients registered to the environment. Client systems have their own `/var` file system.

Each DMS environment is located in a `/var/adm/dms/dmsn.alpha` directory. Depending upon the server and client relationships at your site,

you may have several *n*.alpha areas. Each area must have at least the base operating system mandatory subsets installed as well as other optional software subsets. Space must be reserved for associated or layered products plus an additional 10 percent for file system administration tasks and file system information. For more information about the size requirements of a dataless environment, see the *Sharing Software on a Local Area Network* manual.

If you want the system to serve a dataless environment, you must decide whether you want /var on a separate file system or whether you want to reserve a partition to mount under the /var/adm/dms directory.

You also should consider the space requirements for the /clients directory, which holds the / file system for each dataless client. You can make the /clients directory a separate file system or keep it in the server's / file system in which case you should allocate 64 MB of space for each client.

4.6 Considerations for Customizing Disk Partitions and File System Layout

The following sections describe what you need to know if you want to create your own disk partition sizes or define a different file system layout. The following topics are discussed:

- Whether you should choose AdvFS or UFS as the file system type (Section 4.3)

- How to determine the size and partitioning on existing disks (Section 4.7)

- How to determine how much space the software itself will consume (Section 4.8)

- The basic contents of each file system to use for planning considerations (Section 4.9)

4.7 Determining Existing Disk and Partition Sizes

A system that already is running a previous version of the operating system may already have a customized disk partition table. To check the disk layout and partition sizes, you have to examine the existing disk label. A disk label contains information about the disk such as the disk type, physical parameters, and partition sizes. Without a disk label, a disk is not bootable.

From the text-based Full Installation interface, use the disklabel command to view the disk label as described in Chapter 3. See disklabel(8) for more information.

If you already have started a Full Installation with the graphical interface, view the current disk partition information by clicking on the Edit

Partitions... button on the Custom File System Layout dialog box to open the Disk Configuration application. Figure 4–2 shows a sample disk partition screen from the Disk Configuration application.

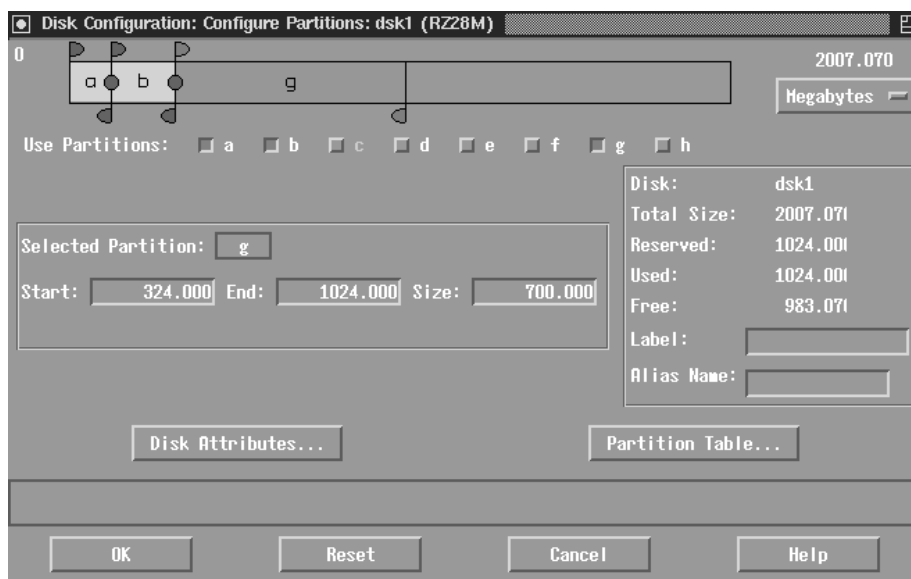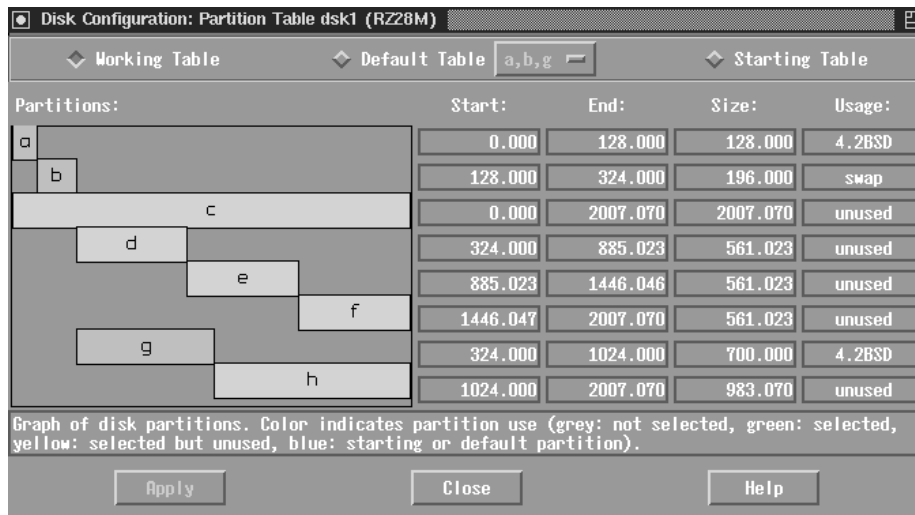**Figure 4–2: Disk Configuration Application: Disk Partition Information**

Figure 4–3 shows a sample disk table screen from the Disk Configuration application.

Figure 4–3: Disk Configuration Application: Disk Partition Table

Disk Configuration: Partition Table dsk1 (RZ28M)

◇ Working Table   ◇ Default Table a,b,g   ◇ Starting Table

Partitions:	Start:	End:	Size:	Usage:
a	0.000	128.000	128.000	4.2BSD
b	128.000	324.000	196.000	swap
c	0.000	2007.070	2007.070	unused
d	324.000	885.023	561.023	unused
e	885.023	1446.046	561.023	unused
f	1446.047	2007.070	561.023	unused
g	324.000	1024.000	700.000	4.2BSD
h	1024.000	2007.070	983.070	unused

Graph of disk partitions. Color indicates partition use (grey: not selected, green: selected, yellow: selected but unused, blue: starting or default partition).

[Apply]   [Close]   [Help]

4.8 How Much Disk Space Does the Software Require?

An important thing you need to know is how much disk space the software
will consume. During a Full Installation you can obtain the amount of space
you need in the `/`, `/usr`, and `/var` file systems in one of two ways:

- If you are using the graphical user interface, from the Software Selection
  window, decide on the type of software you want to install (mandatory,
  optional, or all) and click on the associated Show List... or Edit List...
  button. The bottom of the screen shows the disk space required based
  on your software selections. Use these numbers to determine how big
  the file systems should be. If you intend to install WLS software, choose
  the language to install now because the language software subset sizes
  are calculated into the disk space requirements.

- If you are using the text-based interface, the Full Installation process
  calculates the disk space used as you select optional software subsets. If
  you select a language other than United States English, the language
  software subsets are included in disk space calculations as well. Use
  these numbers to determine how big the file systems should be.

This information helps you decide whether the disk partitions you chose
are large enough to hold the software subsets you want to install. At any
time you can change your disk and partition selections if the partitions are
running out of space. As described in Section 4.8.1, file system overhead is
included in file system space calculations.

_____ **Note** _____

> During a Full Installation, if the partition table on the disk
> chosen for the / file system varies from the recommended table,
> you have the option to use either the recommended disk partitions
> or your customized partitions.

_____

Another manual way to determine software subset sizes is to look in the
software subset size tables in the *Release Notes*.

4.8.1 File System Overhead

When calculating the available disk space for the /, /usr, and /var file
systems, the Full Installation procedure uses approximations for file system
overhead based on the file system type selected for each file system. During
a Full Installation, the amount of disk space that is calculated during
software subset selection includes these overhead requirements.

4.9 Contents of File Systems

The following sections describe the contents of the basic UNIX file systems
and the special space considerations associated with them.

- Section 4.10 describes the contents of the /usr file system including
  space requirements for installing optional software after the Full
  Installation and allocating enough space for user accounts and
  user-created files.

- Section 4.11 describes the contents of the /var file system including
  space considerations to store crash dumps and space for error logger
  and syslog files.

- Section 4.12 describes how swap space is implemented in the operating
  system and provides guidelines about providing enough space for it.

4.10 Contents of the /usr File System

The /usr directory contains the majority of the operating system files,
including libraries, executable programs, and documentation. The directory
structure contains directories such as /usr/sys, /usr/adm, and /usr/bin.
These directories contain required system files and UNIX command binary
files that require a considerable amount of space in the /usr file system.
The general rule is to make the /usr file system as large as possible on
its own disk.

To determine how large the /usr file system might be, consider the following:

- The number of software subsets you plan to install. Most software subsets load a considerable number of system files and commands in the /usr file system. The *Release Notes* document the amount of space consumed by each software subset in each file system.

- The number of user accounts and the amount of space needed by each user if their home directories are in /usr.

_____ **Note** _____

It is recommended that you mount a separate file system or file systems for user accounts so that user information is not overwritten during future Full Installations. It also makes disaster recovery less complicated because you can reinstall the operating system without losing user accounts.

_____

- The size of the /var area if it is on the same partition as /usr.

- The size of /usr/i18n if it is located in the /usr file system.

Over time, you will add files to the /usr file system, and future releases of the operating system may require more space as well. Because of this, the file system can run out of space. Be sure to allow for enough future growth on the /usr file system. A general rule is to make the /usr file system as large as possible.

If you plan to use the Advanced File System (AdvFS) as the file system type and install the AdvFS Utilities (available with a separate license), you do not need to greatly over allocate space for the /usr file system. AdvFS file system space can be increased dynamically without changing directory structures and without system interruption. See the *AdvFS Administration* manual for more information about the AdvFS file system type.

Section 4.10.1 and Section 4.10.2 briefly describe how various items affect the size of the /usr file system and should be considered when planning disk space.

4.10.1 Space for Optional Software Subsets and Associated Products

The /usr file system must be large enough to accommodate the software subsets that will reside within it. A software subset is a collection of executable files and data files needed to perform a specific function or to provide a particular class of services; for example, you need the System Accounting Utilities software subset to perform system accounting.

The *Installation Guide* contains software subset descriptions along with the dependent software subsets and kernel configuration file options related to

each software subset. The *Release Notes* contain tables of software subset sizes.

The mandatory software subsets always are installed. The optional software subsets are not required for the operating system to be fully functional; you can choose none, some, or all of the optional software subsets, depending on your requirements and available disk space.

For WLS installations, the size of the `/usr/i18n` directory must be considered if `/usr/i18n` is not created as a separate file system.

You also may want to allocate space for future installations of associated or layered products that are compatible with this version of the operating system. When planning disk space requirements for `/usr`, allow additional space if you will be adding products in the future. See the *Release Notes* for the associated product for the exact size of the software.

4.10.2 Space for User Accounts and Files

The Full Installation process does not provide an area for user accounts and files; you need to set up this area after the installation. However, you should consider the amount of space needed for user files when planning your system. If you plan to place users' home directories on `/usr`, you should reserve at least 10 to 20 MB of disk space for each user on the system. Multiply that figure by the number of users to get an approximate size of user's home directories.

_____ **Note** _____

It is recommended that you create a separate file system (on another disk if one is available) for users' home directories and mount the new file system perhaps under the `/usr` file system. Placing users' home directories in another file system ensures that the directories will not be overwritten during future Full Installations. It also makes disaster recovery less complicated because you can reinstall the operating system without losing user accounts.

_____

If you intend to set quotas on the user area, multiply the quota for each user by the number of users to determine the amount of user space. See the *System Administration* manual for information on disk quotas.

4.11 Contents of the var File System

The `/var` area contains volatile, machine-specific directories and directories such as `tmp` and `adm`.

You can allocate the /var area either as a file system on its own partition or as a directory under the /usr file system. If system use is heavy, you might want to create a separate /var file system.

To determine the size of the /var area, consider the space requirements of the following:

- Crash dumps (Section 4.11.1)
- Error logger files (Section 4.11.2)
- System accounting files (Section 4.11.3)
- Size of the /var/adm/ris directory, if your system is to be used as a Remote Installation Services (RIS) server (Section 4.5.1)
- Size of the /var/adm/dms directory if your system is to be used as a Dataless Management Services (DMS) server (Section 4.5.2)
- Mail, print, and uucp spooling

If you plan to use AdvFS as the file system type for /var along with the AdvFS Utilities (available with a separate license), you do not need to greatly over allocate space for the /var file system. AdvFS file system space can be increased dynamically without changing directory structures and without system interruption. See the *AdvFS Administration* manual for more information about AdvFS.

4.11.1 Crash Dump Space in the var File System

Two disk areas are used when the system produces a crash dump. As described in Section 4.12, the first area is located in the swap partition and is used to hold the crash dump until the system is rebooted. This area must be large enough to hold a single crash dump.

The second area is where the savecore utility copies the crash dump and a copy of the kernel, /vmunix, when the system is rebooted. This area is located in the /var/adm/crash directory. The disk partition that contains /var/adm/crash must be at least large enough to hold one crash dump and one copy of /vmunix, which is 10 to 13 MB in size. The partition can be made as large as disk resources permit if you want to retain multiple crash dumps.

The crash dump partition must be as large as the size of physical memory on systems configured for full dumps, and can be somewhat smaller on systems configured for partial dumps.

If you want to retain multiple crash dumps, estimate the size of this partition by multiplying the total size required for a single crash dump and a copy of /vmunix by n, where n is the number of crash dumps to retain.

The *System Administration* manual contains a chapter devoted to managing crash dumps and crash dump files, which includes information about how crash dumps are written, choosing partial or full dumps, deciding how much space to reserve for both crash dumps and crash dump files, and much more.

To determine the size and to record the location of the crash dump space, provide the following information:

1. The memory size in MB for your system is _____.

   If you do not know the amount of memory on your system, do one of the following:

   - As superuser or `root`, enter the following command:

     # **uerf | grep -i memory**

     _____ **Note** _____

     The `uerf` command may not be supported on newer EV6 processors. In its place, use Compaq Analyze or DECevent, which can be installed from the Associated Products CD-ROM. See `dia(8)` for more information.

     _____

   - If your system is at the console mode prompt (`>>>`), enter the following command:

     >>> **show mem**

2. You need _____ memory to accommodate your crash dump partition.

4.11.2 Space for Error Logger and syslog Files

The `/var` area requires room to accommodate the log files produced by both `syslog` and the binary error logger. These log files are a record of system events and errors in ASCII text (`syslog`) and binary formats.

The `syslog` utility collects information regarding such system activities as mail, system start-up, system shutdown, rebooting, root account logins, time daemon, printer subsystem, and `syslog` itself. Summary information on hardware errors also is logged. The amount of data logged is related to system activity and the number of users.

The binary error logger records information on hardware errors and system start-up.

If you are creating a new system, estimate your total requirements at about 500 kB per week. There is no limit to how large the `/var/adm/binary.errlog` and the `/var/adm/syslog` files can grow, so they eventually might fill their partition.

4.11.3 Space for System Accounting Files

The `/var/adm` directory contains data files generated by administrative programs such as `acct` and `wtmp`. The data that these programs generate can vary widely from system to system and over time. For example, if you create a `/var/adm/acct` file, it can grow by 50 kB a day for a large system and by 5 kB a day for a workstation.

As a general guideline for system accounting, you should allot 10 kB per day for workstations and 100 kB per day for larger systems. See the *System Administration* manual for more information on the space requirements for system accounting.

4.12 Swap Space Overview

Virtual memory is implemented in the operating system by transparently moving pages back and forth between physical memory and swap space. The amount of virtual address space that can be created is limited only by the amount of swap space. This section discusses some of the factors to consider when configuring swap space on your system. The *System Configuration and Tuning* manual provides additional information about optimizing the use of swap space.

During a Full Installation, you can configure two swap areas: a primary swap partition named `swap1` and an optional swap partition named `swap2`. Additional swap partitions can be configured after the installation is complete by using the procedures described in the *System Administration* manual.

During a Full Installation, you are asked to choose which disk partition to use for `swap1`. The default choice is partition `b` of the system disk.

_____ **Note** _____

It is recommended that you create swap space equal to twice the size of physical memory.

_____

To optimize the use of your swap space, spread out your swap space across multiple devices and use the fastest disks for swap devices. To ensure the best performance, place swap areas on different disks instead of placing multiple swap areas on the same disk. The amount of swap space you allocate also depends on the virtual memory requirements of the applications you plan to install.

Although you cannot choose swap strategy modes during the installation procedure, there are two strategies for swap allocation: immediate and over-commitment. By default, the swap strategy mode used for the operating

systems is immediate mode, which means that swap space is allocated when modifiable virtual address space is created. This mode requires more swap space than over-commitment mode because it guarantees that there will be enough swap space if every modifiable virtual page is modified. See the *System Administration* manual for more information about swap allocation strategies and how to switch from one swap allocation mode to the other after the installation.

Also keep in mind that by default, crash dumps temporarily are stored on the swap partition. This area is used to hold the crash dump until the system is rebooted and must be large enough to hold a single crash dump. This area is referred to as the crash dump partition. In the event of a system crash, the kernel writes the contents of physical memory to the swap partition. The amount of information written, and hence the size of the crash dump, depends on several factors:

- If the system is configured to produce full dumps as described in the *System Administration* manual, the size of the crash dump will be the same as the size of the system's physical memory.

- If the system is configured to produce partial dumps, the crash dump might be considerably smaller.

The factor that determines the size of a partial crash dump is the amount of physical memory in use at the time of the crash by various kernel data structures that define the state of the system. The more tasks and threads that are active, the more kernel data structures that will be in use, and the larger the resulting partial crash dump.

Be prepared to add more swap space later if the system issues warning messages indicating that swap space is approaching exhaustion.

5

Customizing the Installation Process

This chapter contains information about the advanced features that are available for you to automate and customize the installation process. Topics covered in this chapter include:

- An overview of the features that are available for you to automate and customize the installation and system configuration processes (Section 5.1)

- An introduction to configuration description files (CDF) and how they are used for system cloning (Section 5.2)

- The point at which user-supplied files are invoked during a Full Installation or Update Installation to perform customizations you want to make on a target system (Section 5.3)

- The relationship between user-supplied files and CDFs (Section 5.4)

- The high level tasks that are performed by the administrator to set up user supplied files and CDFs (Section 5.5)

- The theory of operation behind how the installation process invokes user-supplied files and CDFs (Section 5.6)

- How to create and position user-supplied files and CDFs for use by the Full and Update Installation processes (Section 5.7)

5.1 Overview

The Full and Update Installation processes have the built-in capability to look for certain files at predefined times. If you copy these files to the right locations, the following can be achieved:

- Installation Cloning

  An `install.cdf` file contains the installation characteristics from an installed system and is used to duplicate the installation to one or more target systems.

- Configuration Cloning

  A `config.cdf` file contains the system configuration characteristics from a configured system and is used to duplicate the configuration on one or more target systems. Configuration is the process of setting up the

network, mail system, internet access, and printers so that the system can communicate with other systems and users.

- Additional customizations through user-supplied files

  You can create files to perform specific tasks on a target system at specific times during a Full or Update Installation. These files must be named `preinstall`, `update_preinstall`, `postload`, `update_postload`, and `postreboot`.

This chapter provides a brief introduction to the concept of configuration description files. The remainder of this chapter is devoted to the task of creating files to perform customized tasks during an installation.

5.2 Overview of Configuration Description Files

Configuration description files (CDF) are used to clone systems. Two distinct CDFs store the installation and configuration characteristics of an installed and configured system. Installation information is stored in the `install.cdf` file, and configuration characteristics are stored in a `config.cdf` file. To briefly describe these two files:

- An `install.cdf` file is generated whenever a system is installed with the current version of the operating system by the Full Installation process. This file is located in the `/var/adm/smlogs` directory. It contains a record of the file system layout, host- and site-specific information, and the software that was installed during a Full Installation. The information in the file is used to clone the same installation on other systems with similar hardware. Because Installation Cloning is an extensive subject that has many options, procedures are documented in Chapter 6. The theory of operation described in Section 5.6 shows where in the Full Installation process the `install.cdf` file is invoked.

- The `config.cdf` file contains network, internet, printer, and mail configuration information that has been saved from a fully installed and configured system. The `config.cdf` file is created manually with the `sysman -clone -save` command whenever you want to save configuration information. The `config.cdf` file can be applied to a target system during a Full Installation, or it can be applied manually to a running system. Because Configuration Cloning is an extensive subject that has many options, procedures are documented in Chapter 7. The theory of operation described in Section 5.6 shows where in the Full Installation process the `config.cdf` file is invoked.

Table 5–1 shows at what points the Full Installation process searches for CDFs and the file names it looks for.

Table 5–1: Invocation Points of Configuration Description Files

Invocation Point During Full Installation	Full Installation Process Searches for a File Named	Action Taken If Found
Before the installation user interface is displayed	install.cdf	The information stored in the install.cdf file is used to clone the same installation on the target system.
After the system reboots but before the tailored kernel build	config.cdf	The information stored in the config.cdf file is used to configure the target system.

CDFs can be used in combination with user-supplied files as described in Section 5.4 to perform additional tasks on the target system.

5.3 Overview of User-Supplied Files

User-supplied files can contain scripts, executables, or programs and are a way to extend and customize the installation process. Table 5–2 shows the invocation points in the installation process, the file names that are searched for, and the type of installation that searches for them. The Full Installation and Update Installation processes always look for these files, and they are executed if found. Except for the postreboot file, if a user-supplied file is executed and returns a non-zero status (which indicates a failure), the installation stops.

Table 5–2: Invocation Points of User-Supplied Files

Invocation Point	Installation Process Searches for a File Named	Searched for During Which Installation Type?
Before the Full Installation user interface is presented	preinstall	Full and Cloned
Before the Update Installation user interface is presented	update_preinstall	Update
After software is installed but before the first reboot of the generic kernel	postload	Full and Cloned
After software is updated but before the system reboots	update_postload	Update
After the first reboot but before the tailored kernel build	postreboot	Full and Cloned

Table 5–3 lists some typical uses of user-supplied files.

Table 5–3: Typical Uses of User-Supplied Files

File Name	Possible Use
preinstall	Define a customized disk label to eliminate the need to do so during the Full Installation.
postload	Dynamically modify host-specific information in the `config.cdf` file so that the target systems are uniquely defined on the network as soon as the cloning process is done. Appendix B shows a sample script that performs this function.
postreboot	Install additional optional software to simplify the software selection process.
update_preinstall	Automatically perform a backup of the operating system before the actual update process starts.
update_postload	Reinstall a layered product that you removed because it prevented the update installation from continuing.

The remainder of this chapter is dedicated to the relationship between CDFs and user-supplied files, how the installation process searches for user supplied files, and how to create and position the files.

5.4 The Relationship Between CDFs and User-Supplied Files

CDFs and user-supplied files can be used independently or in combination. The CDFs and user-supplied files can be located on different sources. For example, either the `install.cdf`, the `config.cdf` or both CDFs can be on a diskette, the `preinstall` file can be on the RIS server, and the `postload` file can be located in the `/isl` directory of the RIS server. However, if the `postload` file is intended to operate on a `config.cdf` file, both files should reside in the same location. User-supplied files can modify CDFs dynamically.

5.5 Summary of Administrator Tasks

Figure 5–1 shows the high level system administrator tasks required to set up CDFs and user-supplied files. To execute user-supplied files during a Full or Update Installation, a system administrator performs Tasks 3 and 4 only.

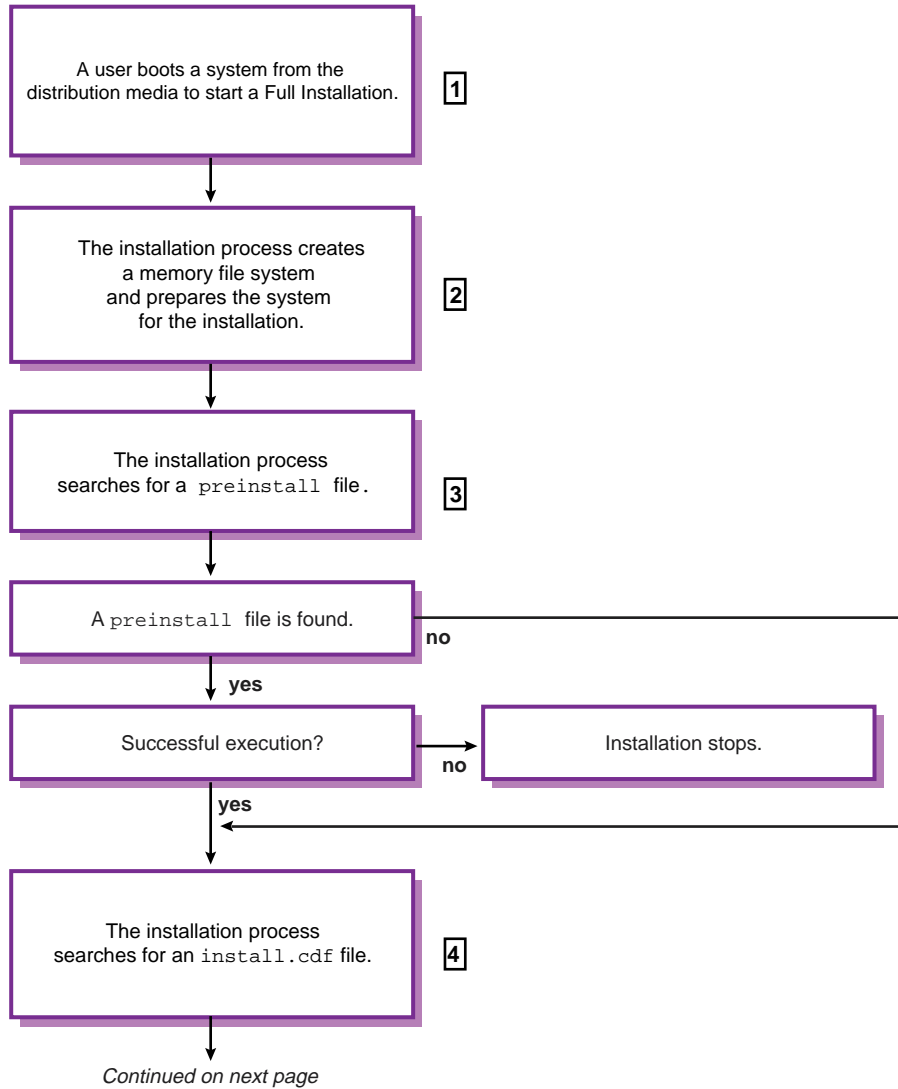**Figure 5–1: Summary of Administrator Tasks**
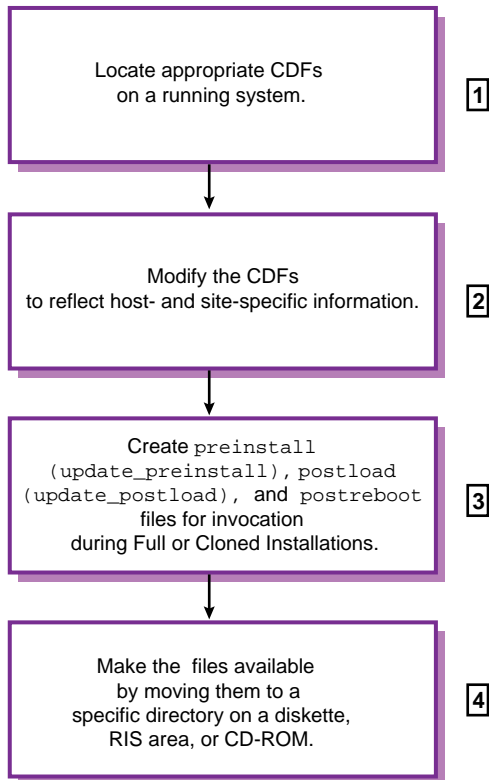
1. An administrator generates or locates CDFs suitable for cloning. On systems that are installed with the current version of the operating system, the `install.cdf` is located in the `/var/adm/smlogs` directory. The `config.cdf` can be saved to any directory by using the `sysman -clone -save` command, but the default directory is `/var/adm/smlogs`. Installation and Configuration Cloning procedures are documented in Chapter 6 and Chapter 7 respectively.

2. The administrator copies original CDFs to a working area for modification. At a minimum, host-specific information must be modified so that the cloned system has a unique identity. Original CDFs should be retained in the `/var/adm/smlogs` directory because they contain information about the initial system installation or configuration that could be valuable for future troubleshooting.

3. The administrator optionally creates scripts or programs to be executed at three predefined points in the Full or Cloned Installation processes (two points in the Update installation process). The administrator determines the actions performed by these files. See Section 5.7 in this chapter for more information about creating these files.

4 The administrator moves the modified CDFs and any
user-supplied files to the `/` (root) directory on a diskette, to the
`/var/adm/ris/clients/sets/`*`profile_set`* directory on a RIS
server, or to the `/isl` directory on a CD–ROM if the distribution media
is being repackaged. The files also can be copied to the `/isl` directory
within an extracted RIS area. See Section 5.8 in this chapter for more
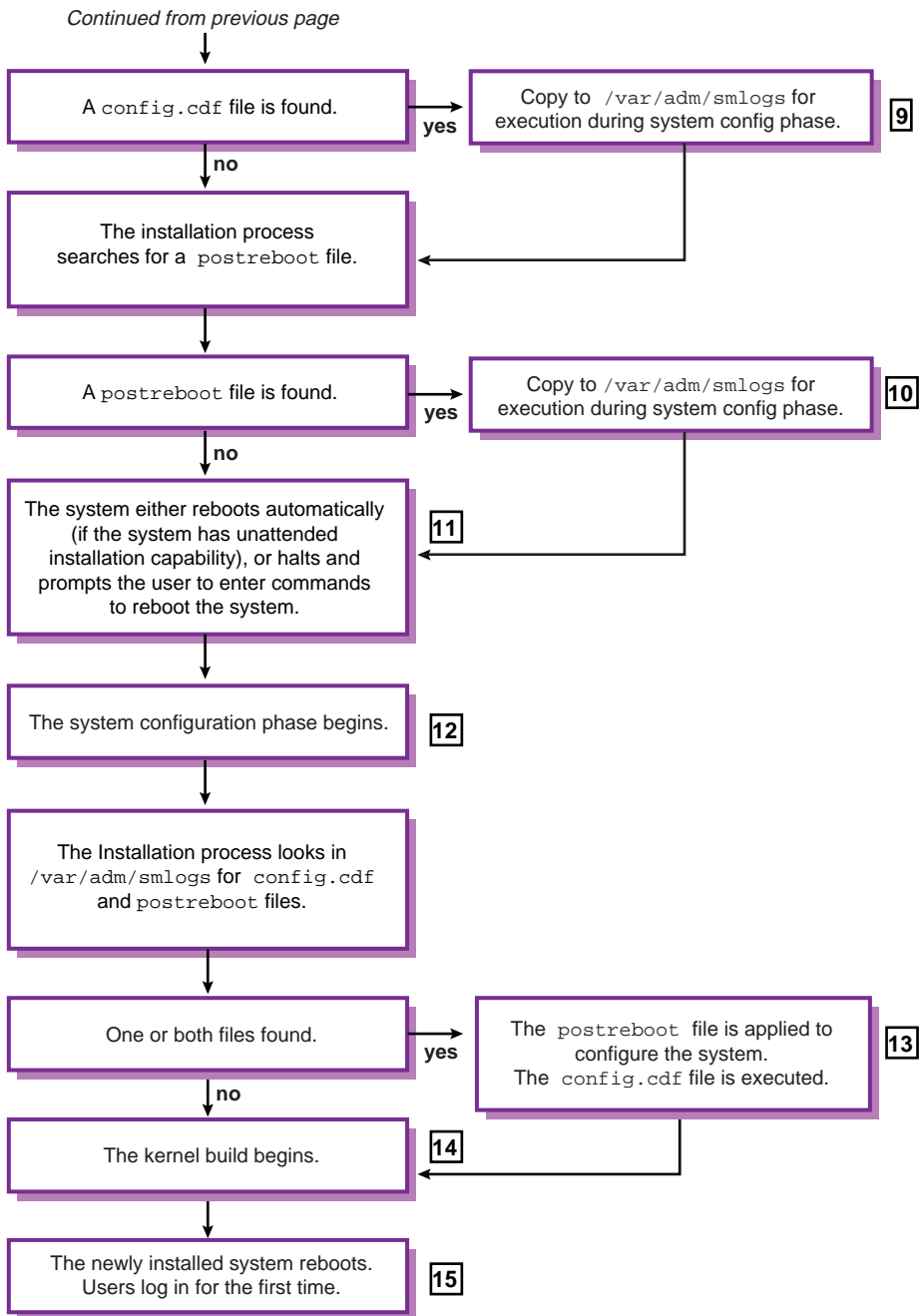information about moving the files to the correct location.

5.6 Theory of Operation

Figure 5–2 contains a summary of how user-supplied files and CDFs are
invoked during a Full Installation.

Figure 5–2: Theory of Operation: User Supplied Files and CDFs

```
┌──────────────────────────────────────┐
│                                      │
│      A user boots a system from the  │   ┌─┐
│  distribution media to start a Full  │   │1│
│             Installation.            │   └─┘
│                                      │
└──────────────────────────────────────┘
                    │
                    ▼
┌──────────────────────────────────────┐
│                                      │
│      The installation process creates│   ┌─┐
│         a memory file system         │   │2│
│          and prepares the system     │   └─┘
│            for the installation.     │
│                                      │
└──────────────────────────────────────┘
                    │
                    ▼
┌──────────────────────────────────────┐
│                                      │
│         The installation process     │   ┌─┐
│     searches for a preinstall file.  │   │3│
│                                      │   └─┘
└──────────────────────────────────────┘
                    │
                    ▼
┌──────────────────────────────────────┐
│      A preinstall file is found.     │────────── no ──────────┐
└──────────────────────────────────────┘                        │
                    │ yes                                        │
                    ▼                                            │
┌──────────────────────────┐      ┌──────────────────────────┐  │
│   Successful execution?  │──no─▶│    Installation stops.   │  │
└──────────────────────────┘      └──────────────────────────┘  │
                    │ yes                                        │
                    ▼◀──────────────────────────────────────────┘
┌──────────────────────────────────────┐
│                                      │
│         The installation process     │   ┌─┐
│     searches for an install.cdf file.│   │4│
│                                      │   └─┘
└──────────────────────────────────────┘
                    │
                    ▼
         *Continued on next page*
```

Continued from previous page

```
An install.cdf file is found.  ──no──▶  The Full Installation interface is displayed.
        │                                              │
       yes                                             ▼
        ▼                                   Users enter information required
The install.cdf file is loaded   [5]       for the Full Installation process.   [6]
       and validated.                                  │
        │                                              │
        ▼                                              │
Successful validation?  ──no──▶  Installation stops.   │
        │                                              │
       yes                                             │
        ▼                                              │
Installation Cloning begins.                           │
        │                                              │
        ▼                                              │
File systems are created.  ◀───────────────────────────
        │
        ▼
The setld command is invoked
 to load software subsets.
        │
        ▼
The Installation process searches   [7]
   for a postload file.
        │
        ▼
A postload file is found.  ──no──▶  Installation continues.
        │                                      │
       yes                                     │
        ▼                                      │
Successful execution?  ──no──▶  Installation stops.   │
        │                                              │
       yes                                             │
        ▼                                              │
The Installation process                               │
searches for a config.cdf file.  ◀─────────────────────
   [8]
        │
        ▼
```

Continued on next page

```
A config.cdf file is found.  ──yes──>  Copy to /var/adm/smlogs for    9
                                       execution during system config phase.
          │
          no
          │
          ▼
The installation process           <───────────────┐
searches for a postreboot file.                     │
          │                                         │
          ▼                                         │
A postreboot file is found.  ──yes──>  Copy to /var/adm/smlogs for    10
                                       execution during system config phase.
          │
          no
          │
          ▼
The system either reboots automatically    11   <───────┘
(if the system has unattended
installation capability), or halts and
prompts the user to enter commands
to reboot the system.
          │
          ▼
The system configuration phase begins.   12
          │
          ▼
The Installation process looks in
/var/adm/smlogs for config.cdf
and postreboot files.
          │
          ▼
One or both files found.  ──yes──>  The postreboot file is applied to    13
                                    configure the system.
          │                         The config.cdf file is executed.
          no
          │
          ▼
The kernel build begins.   14   <──────────┘
          │
          ▼
The newly installed system reboots.   15
Users log in for the first time.
```

1 The system is booted from the distribution media to start a Full Installation.

2 The Memory File System (MFS) provides writable space required by the installation process.

3 The Full Installation process searches for a file named `preinstall`. The search order and file locations are described in Table 5–4. This file is a user-supplied script, program, or executable containing specific actions to be carried out before the installation process begins. If this file is found, it is executed. If execution is successful, the installation process begins. If execution is not successful (that is, an exit status of non-zero is returned), the installation process stops. If a `preinstall` file is not found, the Full Installation process begins the search for an `install.cdf` file.

4 If an `install.cdf` file is found in one of the locations described in Table 5–4, it drives the rest of the installation and begins an Installation Cloning process on the target system. If an `install.cdf` file is not found, a regular Full Installation process begins.

5 The installation process validates the `install.cdf` file before cloning the target system. Validation includes ensuring that the disk names and disk types specified in the CDF exist on the system to be cloned. For RIS installations, validation includes comparing the versions of the software subsets included in the CDF with the software subset versions that are installed in the RIS environment. A CDF validation failure causes the installation to stop. Diagnostic messages display the reason for validation failures. Upon successful CDF validation, the Installation Cloning process continues.

6 If an `install.cdf` file is not found, the installer responds to the questions asked during a regular Full Installation.

7 After software subsets are loaded, the Full Installation process searches for a file named `postload`. The search order and locations are described in Table 5–4. This file is a user-supplied script, program, or executable containing specific actions to be carried out after software subsets are loaded. If this file is found, it is executed. If execution fails, the installation process stops.

8 The Full Installation process searches for a `config.cdf` file in the search order and locations described in Table 5–4.

9 If the `config.cdf` file is found, it is copied to the `/var/adm/smlogs` directory where it will be applied to clone the configuration of the target system later during the system configuration phase of the installation process.

10. The Full Installation process searches for a file named `postreboot`. This file is a user-supplied script, program, or executable containing specific actions to be carried out after the installation process boots the installed system disk for the first time. If this file is found, it is copied to the `/var/adm/smlogs` directory for execution during the system configuration phase.

11. The system automatically reboots after the software subsets are loaded. If your system does not have the capability for automatic reboots, or if the `install.cdf` was not modified so that no user invention is necessary, the installation process halts and prompts the user to enter commands to reboot the system from the newly installed disks. If the system is not set up for automatic reboots, the screen displays the boot commands that must be entered to reboot the system.

12. The system configuration phase begins automatically after the system reboots. Configuration refers to the process of tailoring the software subsets, setting the host name, `root` password, date, time, location and area, tuning the system, and building a kernel. If values are not defined for these attributes or if the installer did not enter a response during the Full Installation, the installation process becomes interactive to request it.

13. The Full Installation process looks in the `/var/adm/smlogs` directory to see if a `config.cdf` or `postreboot` file was moved there. This is where the installation process moved the file; it is not a supported location for users.

If a `config.cdf` file is found, it is used to configure the target system. The Configuration Cloning performs validation checking, which is described in Section 7.8. If the `config.cdf` file fails validation checking, a configuration cloning will not take place, but the installation process continues. To apply a `config.cdf` file to clone another system, the CDF must pass validation.

If a `postreboot` file is found, it is executed. If execution is not successful, the Full Installation continues; execution failure of the `postreboot` file does not prevent the next step in the installation process: the kernel build.

14. For cloned installations, the type of kernel build is defined by the `kernel_options=` attribute in the `install.cdf` file. For regular Full Installations, the type of kernel build depends upon the type of kernel components you chose to build into the kernel: all, mandatory only, or a combination of mandatory and optional kernel components.

15. The last phase of any installation is the final system reboot after which users can log in for the first time. If this is a Full or Cloned Installation, `root` is the only user that may log in.

5.7 Creating User-Supplied Files

The contents of any user-supplied file depends upon what tasks you want to perform, but all user-supplied files must have read and execute permissions. When creating your files, be aware of the environment in which they will be run. For instance, `preinstall` and `postload` files only can call commands and utilities that are available on the distribution media because a full operating system environment is not yet installed. The `postreboot` file can call any command or utility that is available on an installed operating system.

The distribution media is comprised of file systems that are laid out just as the software would be installed. However, for performance reasons and space considerations, the distribution media contains a combination of uncompressed and compressed software subsets. Only those commands and utilities in the uncompressed software subsets are available during an installation. To view the contents of the distribution media, mount it and use a combination of the `cd` and `ls` commands to determine what commands are available to use.

Section 5.7.1 contains more information about creating `preinstall` files; Section 5.7.2 contains more information about creating `postload` files, and Section 5.7.3 contains information about creating `postreboot` files.

5.7.1 Creating preinstall Files

The first invocation point of user supplied files during Full and Update Installations is before the user interface is displayed, and in the case of a Full Installation, before the search for an `install.cdf` file.

At this point, the Full Installation process searches for a file named `preinstall`, which is a user-supplied script, program, or executable. It contains specific actions to be carried out before the installation user interface is presented. The Update Installation looks for a file named `update_preinstall`.

Actions to be carried out before file systems are created and software subsets are loaded might include writing a customized disk label to one or more disks. Another use would be to dynamically modify a generic `install.cdf` file to include host-specific attributes and put that file in `/var/tmp` so the Full Installation process finds it.

You do not want the `preinstall` file to execute any function that requires installed file systems and software to be available because these phases of the installation have not yet been completed; however, you can assume that file systems and software are available during an Update Installation.

The preinstall and update_preinstall files and any files they call require read and execute permission.

It is not necessary that the preinstall file be in the same location as other user-supplied files and CDFs (unless they are being modified by the script).

The installation process queries the return status from the execution of the preinstall and update_preinstall files and terminates the installation process if a non-zero return status is received. The preinstall and update_preinstall files are responsible for supplying their own status or error messages. The installation process does not guarantee the results of executing the script or program but does guarantee that upon successful completion, the installation process proceeds.

The sample preinstall script shown in the following example applies a customized disk label to an RZ26 disk.

Example 5–1: Sample preinstall Script That Calls Another File

```
#!/sbin/sh

#
# Write a custom disk label to the
# system disk before starting the installation.
#

# NOTE:  THIS FILE ASSUMES A DISK NAME OF dsk0 AND DISK TYPE OF RZ26

#
# First, zero the label
#
2>/dev/null disklabel -z dsk0

#
# Next, restore the label
#
disklabel -Rr dsk0 ./DiskLabelSave RZ26 ||                          1
{
 echo "\nError restoring disklabel on dsk0\n"
 exit 1
}

echo "\nThe disklabel that has been applied is:\n"
disklabel -r dsk0 | tail -10
exit 0
```

1 The DiskLabelSave file called by the preinstall script must reside in the same directory as the preinstall script and must have read permissions. The sample DiskLabelSave file is shown in Example 5–2.

The DiskLabelSave file called by the preinstall script contains a disk label that was created by reading the disk label of the disk at dsk0 and redirecting the output into a file. To create this file, enter commands similar to the following:

```
# disklabel -r dsk0 > DiskLabelSave
```

Example 5–2: DiskLabelSave File Called by the Sample preinstall Script

```
# /dev/rdisk/dsk0a:
type: SCSI
disk: rz26
label:
flags:
bytes/sector: 512
sectors/track: 57
tracks/cylinder: 14
sectors/cylinder: 798
cylinders: 2570
sectors/unit: 2050860
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0   # milliseconds
track-to-track seek: 0 # milliseconds
drivedata: 0

8 partitions:
#       size   offset   fstype [fsize bsize   cpg]
 a:   131072        0   4.2BSD   1024  8192    16  # (Cyl.     0 - 164*)
 b:   262144   131072   unused   1024  8192        # (Cyl.   164*- 492*)
 c:  2050860        0   unused   1024  8192        # (Cyl.     0 - 2569)
 d:   552548   393216   unused   1024  8192        # (Cyl.   492*- 1185*)
 e:   552548   945764   unused   1024  8192        # (Cyl.  1185*- 1877*)
 f:   552548  1498312   unused   1024  8192        # (Cyl.  1877*- 2569*)
 g:  1210000   393216   4.2BSD   1024  8192    16  # (Cyl.   492*- 2009*)
 h:   447644  1603216   4.2BSD   1024  8192    16  # (Cyl.  2009*- 2569*)
```

At this point, after creating the DiskLabelSave file, edit the file to create the custom partition sizes you want.

5.7.2 Creating postload Files

Upon completion of the file system creation, software subset load, and the preparation of the configuration environment for the pending software configuration phase, the Full and Update Installation processes search for files named postload and update_postload before the first system reboot.

Actions to be carried out after software subsets are loaded might include creating additional file systems or dynamically modifying a config.cdf file that will be applied to more than one target system. Appendix B contains a sample postload script, which shows how to set host-specific attributes in a config.cdf file.

The postload file and any files that postload calls require read and execute permission. It is not necessary for the postload file to be in the same location as the user-supplied scripts and install.cdf file. However,

if the `postload` file is intended to operate on a `config.cdf` file, both files should reside in the same location.

The installation process queries the results of the execution of the `postload` file and terminates the installation process upon a non-zero return status. The `postload` file is responsible for supplying its own status or error messages. The installation process does not guarantee the results of executing the script or program but does guarantee that upon successful completion, the installation process proceeds.

It is important to know that at this point of a Full Installation, the newly created /, `/usr`, and `/var` file systems on the magnetic media are mount-relative with respect to the `/mnt` directory until the system is rebooted from the newly installed system disk. That is, the / file system is `/mnt`, the `usr` file system is `/mnt/usr`, and so on.

The sample `postload` script shown in Example 5–3 is creating a new file system called `users` and is then adding the entry into the `/etc/fstab` file to mount the new file system upon every reboot.

Example 5–3: Sample postload Script

```
#!/sbin/sh
#
# postload - script which is invoked after the subset load of a full
# installation.  The script creates a new file system and
# adds an entry in the fstab file.  Doing this will make the
# file system available as soon as the installation completes.


#
# Create a new file system on dsk2c which is to be mounted at /usr/users
#

echo "postload:  creating new file system on dsk2c\n"

# Create the UFS file system on dsk2c, an RZ26L disk.

/usr/sbin/newfs -F /dev/rdisk/dsk2c RZ26L ||
{
    echo "postload:  failed to create a new file system on dsk2c\n"

    # We consider this a nonfatal error and allow the install to
    # continue.  This is done by returning 0.  Otherwise, exit with a
    # non-zero value.

    exit 0
}
# Next, add an entry to fstab so that this new file system is
# automatically mounted when the system boots.

# NOTE:  the actual installed file systems are mounted at /mnt.
# Therefore, we want to add the entry to /mnt/etc/fstab and
# not /etc/fstab.

echo "/dev/disk/dsk2c /usr/users ufs rw 1 2" >> /mnt/etc/fstab

# Finally, make sure the mount point is created.  Again, create it
# relative to /mnt.
```

Example 5–3: Sample postload Script (cont.)

```
/bin/mkdir /mnt/usr/users

# Process complete!

exit 0
```

5.7.3 Creating postreboot Files

To provide a place for users who already have written scripts to configure
their systems as well as to allow supported scripting capabilities from a
configured system, a third invocation point is available.

This user-supplied file is called `postreboot` to signify its relative location
in the installation process. This file is searched for and is invoked during
the software configuration phase of a Full Installation. The software
configuration phase occurs after subsets have been loaded and the system is
rebooted with the generic kernel. More specifically, the `postreboot` script
is invoked after the check for a `config.cdf` file so that the `postreboot`
script can take advantage of a network-configured system.

In addition, the `postreboot` script is invoked before the tailored kernel
build so that additional layered software (for example, those with kernel
dependencies) could be installed and the required kernel components will be
satisfied by the tailored kernel build. The `postreboot` script is searched
for and executed if found during a Full Installation; the Update Installation
process does not look for this file.

The installation process queries the results of the execution of the
`postreboot` file but does not terminate the installation process upon a
non-zero return status. The `postreboot` file is responsible for supplying its
own status or error messages. The installation process does not guarantee
the results of executing the script or program but does guarantee that
whether or not the script succeeds or fails, the installation process proceeds.

Section B.5 contains a sample `postreboot` script.

5.8 Copying User-Supplied Files and CDFs to the Right Location

CDFs and user-supplied files and all additional files they require must be
located in the right directories so the installation process can find them.

Both the Full and Update Installation process search for the user-supplied
files and CDFs in the order shown in Table 5–4. As soon as a file is found, the

installation process stops looking in the remaining locations. For example, if the installation process finds a `preinstall` file on diskette, it does not look on the RIS server.

Table 5–4: Acceptable Locations of User-Supplied Files and CDFs

Search Order	Location	Copy Instructions Located In
1	In the / (root) directory of diskette drive `floppy0` or `floppy1`.	Section 5.8.1
2	In the *profile_set* subdirectory of the `/var/adm/ris/clients/sets/` directory on the RIS server to which the client system is registered.	Section 5.8.2
3	In the `/var/tmp` memory file system (MFS) on the system to be cloned.	Section 5.8.3
4	In the `/isl` directory on the distribution media (local CD–ROM or extracted RIS area).	Section 5.8.4

5.8.1 Copying Files to a Diskette

Before you can copy user-supplied files and CDFs to the diskette, you may have to format the diskette, write a new disk label, and then create a new file system using the following command syntax:

fddisk -fmt *raw_diskette_device*

disklabel -wr *diskette_drive disk_type*

newfs *raw_diskette_device_partition*

Follow this procedure to format the diskette in diskette drive `floppy0`, write a new disk label specifying the `rx23` type of diskette (standard 3.5-inch diskette), and create a new file system on the entire diskette (partition c):

1. Format the diskette in drive `floppy0`:

    ```
    # fddisk -fmt /dev/rdisk/floppy0c
    ```

2. Write a new disk label to a standard 3.5-inch diskette:

    ```
    # disklabel -wr floppy0 rx23
    ```

3. Create a new file system on the entire diskette, the `c` partition:

    ```
    # newfs /dev/rdisk/floppy0c
    ```

If either the `preinstall`, `postload` or `postreboot` files are located on the diskette, all files called by the `preinstall`, `postload` or `postreboot` files must be located on the diskette as well.

Follow this procedure to mount the diskette drive and copy the files to the diskette:

1. Mount the diskette drive on the /mnt mount point:

    # **mount /dev/disk/floppy0c /mnt**

2. Assuming that you are in the directory in which the files are located, enter the following commands to copy the files to the diskette:

    # **cp ./install.cdf /mnt/install.cdf**
    # **cp ./preinstall /mnt/preinstall**
    # **cp ./postload /mnt/postload**
    # **cp ./config.cdf /mnt/config.cdf**
    # **cp ./postreboot /mnt/postreboot**
    # **cp ./**_file_name_ **/mnt/**_file_name_

3. Enter the chmod command to ensure all files have execute permissions:

    # **chmod 755 /mnt/***

4. Unmount the diskette drive:

    # **umount /mnt**

5. Remove the diskette from the drive and insert it into the diskette drive of the target system.

6. Start a Full installation on the target system (as described in the *Installation Guide*).

5.8.2 Copying Files to a RIS Server Profile Set Directory

A Remote Installation Services (RIS) server stores CDFs and user-supplied files in logically organized subdirectories that are created by the RIS administrator. These subdirectories, known as profile sets, are located in the /var/adm/ris/clients/sets directory. When a system is registered as a RIS client, you also can register the system to the profile set that contains the CDFs or user supplied files you want to execute during the installation.

For more information about profile sets and RIS administration, see the guide to *Sharing Software on a Local Area Network*.

After you or the RIS administrator have established naming conventions and a structure for the profile set directory on the RIS server, use procedures similar to the following to copy CDFs, user-supplied files, and any related files to a profile set directory:

1. Log in to the RIS server as root.

2. Change to the /var/adm/ris/clients/sets directory:

    # **cd /var/adm/ris/clients/sets**

3. Using the naming scheme of your choice, create a profile set directory with a meaningful name. This example is creating a profile set directory for the Engineering department:

    # **mkdir engineering**

4. Change to the new profile set directory to ensure that files are copied to the correct directory:

    # **cd engineering**

5. Copy the right CDFs, user-supplied files, and all other related files from your working area to the new `engineering` profile set directory with your preferred copy tool (`ftp`, `dcp`, or `rcp`).

6. Enter the `chmod` command to ensure all files have read and execute permissions:

    # **chmod 755 ***

7. Invoke the `ris` utility to register the target system (or systems) to the correct RIS software environment and profile set directory:

    # **/usr/sbin/ris**

8. Start the Full Installation on the target system (as described in the *Installation Guide*).

5.8.3 Copying Files to the /var/tmp Directory

The `/var/tmp` directory is a writable directory created during the installation process and, therefore, it cannot be used to ship the CDFs and user-supplied files. However, if a `preinstall` script is used, it can dynamically copy the CDFs, `postload`, `postreboot`, and any files needed by `postload` and `postreboot` into `/var/tmp` during the installation process. The `preinstall` file itself cannot be invoked from `/var/tmp` as it is the only mechanism available to move files into `/var/tmp`.

This feature is valuable if you are repackaging the operating system and you are providing CDFs and user-supplied files on the CD-ROM. This feature is also useful for dynamically modifying a generic CDF that might exist in the RIS area to which many client systems are registered. User-supplied files could be used to modify the CDF as appropriate for the client being installed and configured and deposit the resulting file into the `/var/tmp` directory so it is found by the installation process.

The `/var/tmp` location is third in the search order for user-supplied files. If you are using this location as a writable area, the first two locations (that is, a diskette drive or RIS area) must not contain the same user-supplied files because when a file is found, the search ends.

When there is a need to modify or select an `install.cdf`, `config.cdf`, `postload`, or `postreboot` file as part of the installation process, a writable location is needed because the system cannot write to the CD-ROM. For example, assume that several CDFs are shipped on the CD-ROM for the purpose of supporting different hardware or configurations from one distribution media. In this case, you can create a `preinstall` file that examines the system on which the installation is being executed, and based on the examination, select the right CDF file from among those shipped. The `preinstall` file can copy this CDF to `/var/tmp/install.cdf` where it later will be read by the installation process. Similarly, the `preinstall` file could choose from among several `postload` files and copy the right one to `/var/tmp/postload`.

The `preinstall` script should ensure that files copied to `/var/tmp` have the correct permissions. Issuing the `chmod 777 *` command is the safest way to ensure correct permissions.

5.8.4 Copying Files to CD–ROM

You can repackage the operating system CD–ROM to include CDFs and user-supplied files in the `/isl` directory.

_____ **Note** _____

Copying software may be done only for the purpose of licensed use of the operating system. A valid license agreement must be present for all instances of use of the copied operating system.

_____

Use the method you usually use to create a CD–ROM (that is, write to a CD-ROM) if you plan to provide the `install.cdf`, `config.cdf`, `preinstall`, `postload`, and `postreboot` files to the `/isl` directory on a CD–ROM. The method you use depends upon the type of CD–ROM burner you have.

Follow this basic procedure to create an image on a CD–ROM:

1. Mount the Tru64 UNIX Version 5.1B CD–ROM to determine how much disk space is required on the magnetic disk to which you will be copying the contents of the CD–ROM. For example, to mount the CD–ROM in drive `/dev/disk/cdrom0c` on the directory `/mnt`, enter commands similar to the following:

    ```
    # mkdir /mnt
    # mount /dev/disk/cdrom0c /mnt
    # cd /mnt
    ```

2. Enter the following command to determine disk space in kilobytes:

```
# df -k
```

Remember this number and make sure you have a disk large enough to meet the space requirement.

Record it here: _____

3. Erase the disk label of the magnetic disk before creating the image:

    ```
    # disklabel -z /dev/disk/dsk2
    ```

4. Apply the default disk label to the same disk:

    ```
    # disklabel -rw /dev/disk/dsk2
    ```

5. Change directory to the mount point of the CD-ROM:

    ```
    # cd /mnt
    ```

6. Using the figure obtained in Step 2, make sure that the file system in which you will be creating the tar file in the next step has enough space to hold the file. You may need to mount another disk and create a new temporary file system. Optionally, the /spare directory used in the example could be a directory in another existing file system such as /var/spare or /usr/spare if those file systems have enough free disk space. Modify the remaining steps accordingly depending upon how you solve the disk space situation.

7. Create a tar file containing the contents of the CD-ROM:

    ```
    # tar cf /spare/OS1.tar .
    ```

8. Change out of the mounted directory and unmount the CD-ROM after the tar file is created:

    ```
    # cd /
    # umount /mnt
    ```

9. Create a new file system on dsk2 and mount it:

    ```
    # newfs /dev/disk/dsk2c
    # mkdir /cdimage
    # mount /dev/disk/dsk2c /cdimage
    ```

10. Copy the OS1.tar file to the new disk:

    ```
    # cd /cdimage
    # tar xf /spare/OS1.tar
    ```

11. Change to the directory where your files are located, and use the cp command to copy the install.cdf, preinstall, postload, config.cdf, and postreboot files and any files called by these files into the /cdimage/isl directory of the image.

    Files for a Full Installation:

    ```
    # cd file_location
    # cp ./preinstall /cdimage/isl/preinstall
    ```

```
# cp ./install.cdf /cdimage/isl/install.cdf
# cp ./postload /cdimage/isl/postload
# cp ./config.cdf /cdimage/isl/config.cdf
# cp ./postreboot /cdimage/isl/postreboot
# cp ./filename /cdimage/isl/filename
```

Files for an Update Installation:

```
# cd file_location
# cp ./update_preinstall /cdimage/isl/update_preinstall
# cp ./update_postload /cdimage/isl/update_postload
# cp ./filename /cdimage/isl/filename
```

12. Depending upon the type of CD–ROM burner you have, use the recommended method to burn a CD–ROM from the modified image on the disk. The label of the CD-ROM to be burned must match the label of the operating system CD-ROM; otherwise the process will fail. Use the `disklabel -r` command and look for `label: string` to determine the label on the operating system CD-ROM.

_____ **Note** _____

To ensure that you have a valid, bootable operating system image, it is recommended that you verify the ability to boot from the image on the disk before burning the CD–ROM.

_____

6

Installation Cloning

This chapter discusses the following topics:

- An overview of the Installation Cloning process (Section 6.1)
- How an Installation Cloning happens (Section 6.2)
- An overview of the installation configuration description file (CDF), `install.cdf` (Section 6.3)
- The format and contents of the `install.cdf` file (Section 6.3.1)
- A sample `install.cdf` file (Section 6.3.2)
- Step by step procedures that describe how to generate a suitable CDF, edit the CDF, and start the Installation Cloning process (Section 6.4)

See Chapter 5 for information about the other customizations that can be made to a Full Installation process.

6.1 Overview

Installation Cloning lets you duplicate the installation characteristics (that is, the file systems and installed software) from a running system onto one or more systems with the same or similar hardware configuration.

When you install the current version of the operating system on a machine, the installation process automatically generates a configuration description file (CDF) that contains a record of the installation setup data you specified. This file is created in the `/var/adm/smlogs` directory under the file name `install.cdf`. The `install.cdf` file contains all the installation information required to perform the same installation on a target system.

_____ **Note** _____

If you want to clone Version 5.1B of the operating system onto a target system, the CDF must be created by a Version 5.1B Full Installation. Installation cloning is not supported between different releases of the operating system because CDFs created by other versions of the operating system are not compatible with the current version.

_____

Systems that are installed by the cloning process must have the same disk configuration as the system where the CDF was generated. This means that the disks used for the `/ (root)`, `/usr`, `/var`, `/usr/i18n` file systems and `swap` areas on both systems must have the same disk type and the same device name. It is possible, however, to accommodate slight differences in configuration. Section 6.5.1 describes these acceptable differences.

Attributes in the `install.cdf` file can be modified so that there is no user intervention required at the target system. The `install.cdf` file also contains host- and site-specific attributes that you should modify in order to give the cloned system a unique identity. You most likely will have to change the host name entry for any system you want to clone.

Installation Cloning can be combined with Configuration Cloning, which is described in Chapter 7, and user-supplied scripts, which are described in Chapter 5, to totally clone and customize one or more target systems from a single installed and configured system.

Using Installation Cloning to mass-install systems has the following benefits:

- You can produce identical installations with less effort.

- You can set up the Installation Cloning process to run with minimal user intervention.

- You can save time and reduce the chance of error in environments because Installation Cloning eliminates the need to manually perform duplicate installations on all systems.

- You can administer software centrally, instead of attempting concurrent installations with locally-mounted removable media such as CD-ROMs.

6.2 How Does It Happen?

To clone a system, you copy the `install.cdf` file from the original model system to one of the four locations shown in Table 6–1. When you begin a Full Installation on a system you want to clone, the installation process looks for the `install.cdf` file in the order shown.

Table 6–1: Search Order for the install.cdf File

Search Order	Location
1	On a diskette in diskette drive `floppy0` or `floppy1`.
2	In the `/var/adm/ris/clients/sets/`*profile_set* subdirectory on the RIS server. During the RIS client registration process, the target system must be registered to the *profile_set* directory with the `install.cdf` file you want to use.

Table 6–1: Search Order for the install.cdf File (cont.)

Search Order	Location
3	In the `/var/tmp` memory file system (MFS) on the system to be cloned.
4	In the `/isl` directory on the distribution media (local CD–ROM or extracted RIS area).

If the installation process finds an `install.cdf` file in any of the locations shown in Table 6–1, an Installation Cloning begins on the target system. As soon as the file is found, the installation process stops looking in the remaining locations. For example, if the installation process finds the `install.cdf` file on diskette, it does not look on the RIS server. If an `install.cdf` file is not found in any of the four locations, a regular Full Installation begins on the target system.

A majority of the remainder of this chapter is provided to assist you in editing the `install.cdf` file. If you are performing the Full Installation from CD-ROM (regardless of where the CDF is located), read Section 6.6.3, which describes the importance of modifying host- and site-specific attributes.

If you want to learn more about the format and contents of the CDF, read Section 6.3, Section 6.3.1, and Section 6.3.2. Go directly to Section 6.4 for step by step instructions for performing an Installation Cloning.

6.3 Installation CDF Overview

An `install.cdf` file contains the following information about an installation:

- Where file systems were created: / (root), `/usr`, `/var`, and `/usr/i18n`
- Where swap space was created
- The disklabel definition for each device
- Disk types and disk names where file systems reside
- File system layout (the specific partitions where file systems reside)
- File system types: UNIX File System (UFS) or Advanced File System (AdvFS)
- Logical Storage Manager (LSM) configuration
- Host-specific information such as host name and encrypted `root` password and site-specific information such as location and area (also known as time zones)
- Type of distribution media (CD–ROM or RIS) from which the installation took place

- Software subsets that were installed

Section 6.3.1 describes the format and contents of the installation CDF, and Section 6.3.2 shows a sample installation CDF.

6.3.1 CDF Format and Contents

The `install.cdf` file is organized as groupings of attribute-value pairs. An equal sign (`=`) separates each attribute-value pair. An _item= defines each logical grouping of attribute-value pairs. See `stanza(4)` for more information about stanza file format.

Table 6–2 describes each _item= in the `install.cdf` file. Appendix A provides detailed descriptions of the attribute-value pairs in each item.

Table 6–2: Items in the install.cdf File

_item=	Contents
Inst_islinfo	Information about the media used for the installation (CD-ROM, RIS, or clone) and other system information that conveys the state of the system before the start of the installation process.
Inst_disklabel	Disk configuration information such as partition sizes and offsets.
Inst_filesystem	File system information such as the number and type of file systems that were created on the system. There is one Inst_filesystem item for every file system and swap area that was created. At a minimum, there are four Inst_filesystem items in the CDF to describe the / (root), /usr, and /var file systems and the swap device.
Inst_subsets	A list of the installed base software subsets. If you installed Worldwide Language Support (WLS) subsets, up to two additional Inst_subsets item were created. If you are cloning a system that was installed with a hardware release, another Inst_subsets item contains a list of hardware-specific base subsets.
Inst_cinstall	Target system configuration information, which is conveyed to the installation process. All of the attributes specified in the Inst_cinstall item are optional. If values are not provided for these attributes, the installation process becomes interactive to request this information during the system configuration phase.
Inst_lsm_disks	Logical Storage Manager (LSM) private region partition information.
Inst_lsm_global	Global LSM information including the size of the private region and the LSM host name.

6.3.2 Sample Installation CDF

Example 6–1 shows the contents of an `install.cdf` file. Appendix A
provides descriptions of each attribute and its valid values. The order of the
attributes within each _item is not relevant.

Example 6–1: Sample Installation CDF

```
install:
 _item=Inst_islinfo
 _action=create
 media_type=CDROM
 srcloc=/ALPHA/BASE

install:
 _item=Inst_disklabel
 g_size=1433600
 c_offset=0
 e_offset=1812528
 b_size=401408
 g_offset=663552
 d_size=1148976
 b_offset=262144
 f_size=1148976
 name=dsk1
 h_size=2013328
 d_offset=663552
 a_size=262144
 f_offset=2961504
 c_size=4110480
 _action=create
 h_offset=2097152
 e_size=1148976
 a_offset=0

install:
 _item=Inst_filesystem
 disk_number=1
 disk_name=dsk1
 controller_type=SCSI
 name=root
 partition=a
 controller_number=0
 disk_type=RZ28M
 file_system_type=AdvFS
 _action=create

install:
 _item=Inst_filesystem
 disk_number=1
```

Example 6–1: Sample Installation CDF (cont.)

```
 disk_name=dsk1
 controller_type=SCSI
 name=usr
 partition=g
 controller_number=0
 disk_type=RZ28M
 file_system_type=AdvFS
 _action=create

install:
 _item=Inst_filesystem
 disk_number=1
 disk_name="in usr_domain"
 controller_type=SCSI
 name=var
 partition=g
 controller_number=0
 disk_type=RZ28M
 file_system_type=AdvFS
 _action=create

install:
 _item=Inst_filesystem
 disk_number=1
 disk_name=dsk1
 controller_type=SCSI
 name=swap1
 partition=b
 controller_number=0
 disk_type=RZ28M
 file_system_type=swap
 _action=create

install:
 _item=Inst_filesystem
 disk_number=1
 disk_name="in usr_domain"
 controller_type=SCSI
 name=i18n
 partition=g
 controller_number=0
 disk_type=RZ28M
 file_system_type=AdvFS
 _action=create

install:
 _item=Inst_subsets
```

Example 6–1: Sample Installation CDF (cont.)

```
volume_name=DISC1
name=BASE
ss_names=OSFADVFS540,OSFADVFSBIN540,OSFBASE540,OSFBIN540,
   OSFBINCOM540,OSFCDEDT540,OSFCDEMAIL540,OSFCDEMIN540,
   OSFCLINET540,OSFCMPLRS540,OSFFONT15540,OSFHWBASE540,
   OSFHWBIN540,OSFHWBINCOM540,OSFJAVA540,OSFKBDPCXAL540,
   OSFMITFONT540,OSFNETCONF540,OSFNETSCAPE540,OSFNFS540,
   OSFNFSCONF540,OSFOLDX11540,OSFPRINT540,OSFSER540,
   OSFSERPC540,OSFSYSMAN540,OSFTCLBASE540,OSFTKBASE540,
   OSFX11540,OSFXADMIN540,OSFXPRINT540,OSFXSYSMAN540
advflag=1
_action=create

install:
_item=Inst_subsets
volume_name=DISC2
name=Worldwide_Language_Support
ss_names=IOSPLCDEDT540,IOSPLCDEMAIL540,IOSPLCDEMIN540,
   IOSPLOLDX11540,IOSPLUCSBASE540,IOSPLX11540,IOSWWBASE540,
   IOSWWLAT2FONT100M540,IOSWWLAT9FONT100M540,IOSWWPRINT540,
   IOSWWSYSMAN540,IOSWWUCSBASE540,IOSWWX11540
advflag=1
_action=create

install:
_item=Inst_cinstall
kernel_option=interactive
timeset=yes
 lang_env=C
password=Bp2xAe46zVpUo
timezone=New_York
locality=America
_action=create
hostname=taurus
```

6.4 Summary of Installation Cloning Procedures

This section summarizes the tasks to set up and perform an Installation
Cloning on a target system:

1. Create or select a suitable installation CDF (Section 6.5).

2. Modify the installation CDF to set host- and site-specific attributes and
    include certain attributes to eliminate the need for user intervention at
    the cloned system (Section 6.6).

3. Optionally create user-supplied scripts or a `config.cdf` file to execute during the Full Installation as well (Section 6.7).

4. Copy the installation CDF to the right location depending upon your distribution needs (Section 6.8).

5. Start a Full Installation on the target system (Section 6.9).

The following sections provide more detail about each step in the Installation Cloning process.

6.5 Step 1: Creating or Selecting a Suitable CDF

Whether you are intentionally installing a model system to generate a CDF or you are selecting an existing CDF to use to clone target systems, you must consider the disk configuration, graphics adapter, font sizes and keyboard types of the systems to be cloned. Ideally, you should clone systems with identical hardware configurations.

During a regular Full Installation of a model system (not during cloning), the installation process automatically determines the mandatory software subsets required to support the graphics adapters, font sizes, and keyboard types resident on the system. All other software subsets are considered optional and are not installed unless you specifically select them.

When cloning a system, the CDF defines the software subsets to be installed on the target system. Therefore, if the target system has a different graphics adapter, font size, or keyboard type from the model system on which the CDF was created, the right software subsets will not be installed, and the cloned system may not be usable.

To generate an installation CDF that is versatile enough for use across differing systems, you may want to consider performing a Full Installation on a model system so that the CDF generated from that installation is usable by systems with different graphics adapters, font sizes, and keyboards. You do this by installing the software subsets to support all graphics adapters, font sizes, and keyboard types required by the systems to be cloned even though they are not required by the model system. Installing these optional software subsets will result in additional software being loaded on each system, but it will create a generic CDF that can be used to clone target systems with different configurations.

The following sections describe acceptable differences between the model system and target systems with respect to disk configuration, graphics adapters, font sizes, and keyboard types.

6.5.1 Acceptable Differences in Disks

Target systems should have the same hardware configuration as the model system where the CDF was generated. However, it is possible to support slight differences.

The disks on which /, /usr, swap1, /var, /usr/i18n (if it is not a directory under /usr) and swap2 (if allocated) must have the same configuration on both the target system and the model system from which the CDF was generated. The same disk configuration means that the disk type (for example RZ1BB) and the device name (for example dsk0) must match. If the partition tables for these disks are not identical on both systems, the cloning process will reconfigure the target system's disks as described in the CDF. It does not matter if disks not touched during the cloning process are different on the target system.

_____ **Note** _____

If the cloning process reconfigures a disk or disks on the target system, be aware that this action may destroy user data on the reconfigured disk or disks.

_____

The install.cdf file contains the disk type and disk name for the model system. To determine the disk type and disk names on the target system, issue the show dev command when the target system is in console mode (>>>).

Table 6–3 illustrates one example of acceptable differences in disk configuration between a CDF generated from a model system and a target system. Both systems have two disks, but the second disk on the target system is an RZ1BB disk instead of an RZ1CB disk. This difference does not matter because the cloning is taking place on the first RZ1BB with device name dsk0, which both systems have in common.

Table 6–3: Acceptable Differences in Disks Between a Model System and a Target System

System	Disk Type	Device Name
model system	RZ1BB	dsk0[a]
	RZ1CB	dsk1
target system	RZ1BB	dsk0
	RZ1BB	dsk1

[a] The / (root) and /usr file systems and swap1 space are located on the dsk0 device on the model system.

Assuming there are no other differences in disk configuration, the target system can use the CDF generated from the model system. As long as the

/, /usr, and /var file systems and swap space are not on dsk1, the disk type at device name dsk1 can be different. If the disk device at dsk0 were different, however, the cloned installation would fail.

6.5.2 Differences in Graphics Adapters

If any of the target systems have different graphics adapters from the model system, the software subsets required to support the graphics options needed by those systems must be installed on the model system or must be added manually to the install.cdf file before starting the cloning process on the target system.

When selecting software subsets during the installation of the model system, look in the Windowing Environment category for software subsets starting with the words X Servers for <name>. Replace <name> with the name that describes the graphics options supported by the software subset. The following graphics software subsets are available:

- X Servers Base – Device independent X Server support (always installed)

- X Servers for PCbus – Supports EISA bus and PCI bus graphics adapters

_____ **Note** _____

All graphic adapters supported by this release of the operating system are listed in the *Software Product Description* (SPD). If you have a third party graphics adapter attached to the target system, you will have to install the support for the adapter after the Installation Cloning take place.

_____

Table 6–4 shows the graphics adapters on a model system and a target system.

Table 6–4: Graphics Adapters on a Model System and a Target System

System	Graphics Adapter
model system	Device independent X server
target system	QVision (PCbus)

Based on the model system's configuration, the installation process automatically installs the software subset X Servers for Base as a mandatory subset for the model system. Therefore, the X Servers for PCbus software subset is optional for the model system. To ensure the availability of the right software for the cloned system, you must install the

`X Servers for PCbus` software subset onto the model system; otherwise, the target system will not have graphics capability.

_____ **Caution** _____

Do not use the CDF from a system that does not have graphics capabilities to clone systems that have the hardware to support graphics. There are several software subsets, most notably those associated with the common desktop environment (CDE), that will not be loaded on systems without graphics capabilities that are mandatory for systems with graphics capabilities.

You can ensure that you provide the right graphics by installing all of the graphics software subsets on the model system. Be aware that installing all of the software subsets requires more disk space than loading only selected graphics software subsets.

6.5.3 Differences in Font Size

When generating the CDF through the Full Installation of a model system, you must consider the font sizes required by the target systems. If the target systems require different size fonts from those defined in the CDF, load the right font software subset when installing the model system.

The need for DECwindows 75dpi Fonts or DECwindows 100dpi Fonts depends on the resolution of the graphics adapter being used. On a system already installed with the operating system, this value can be determined by entering the following command:

`# /usr/sbin/sizer -gr`

When the resolution is 1024x768 or less, the DECwindows 75dpi Fonts are required. When the resolution is greater, the DECwindows 100dpi Fonts are required. If you are unsure of the resolution available on the systems to be cloned, select both font software subsets to ensure that the correct font is available.

Systems with multiple graphics adapters may require both the DECwindows 75dpi Fonts and DECwindows 100dpi Fonts if the adapters include those with 1024x768 or less resolution and those with greater resolution.

While there are other software subsets that contain fonts, only the DECwindows fonts are packaged separately by size.

Table 6–5 shows the different font sizes required on a model system and a target system.

Table 6–5: Font Sizes on a Model System and a Target System

System	Graphics Resolution	Required Font Size
model system	1024x768	DECwindows 75dpi Fonts
target system	1280x1024	DECwindows 100dpi Fonts

During the installation of the model system, the DECwindows 75dpi Fonts software subset is mandatory and is installed automatically; the DECwindows 100dpi Fonts software subset is optional. You should install the optional software subset to provide the necessary fonts for the Installation Cloning of the target system.

You can ensure that you provide the right fonts by installing all of the font software subsets on the model system. Installing all of the font software subsets will require more disk space than loading selected fonts.

6.5.4 Differences in Keyboard Type

When generating the CDF through the installation of a model system, you must consider the keyboard type of the systems that will be cloned using the CDF. If the systems that will be cloned have different keyboard types, load the right keyboard support software subset when installing the model system.

To determine the keyboard type on a system already installed with the current release of the operating system, use the following command:

```
# /usr/sbin/sizer -wk
```

Table 6–6 shows an example of the keyboard types on a model system and a target system.

Table 6–6: Keyboard Types on a Model System and a Target System

System	Keyboard Type
model system	PXCAL
target system	LK411

Based on the model system's configuration, the installation process automatically installs the software subset PCXAL Keyboard Support as a mandatory subset for the model system. Therefore, the software subset for LK411 Keyboard Support is optional. Installing this optional software subset results in some unnecessary software being loaded on the model system but allows the CDF to be suitable to clone the target system.

You can ensure that you provide the right keyboard type by installing all of the keyboard software subsets on the model system. Be aware that loading

all keyboard software subsets requires more disk space on the model system than loading selected keyboard software subsets.

6.6 Step 2: Modifying the CDF

You have the option to modify the `install.cdf` file so that the Full Installation bypasses all user responses usually required during a Full Installation process. It is recommended to modify host- and site-specific attributes so that the target system has a unique identity when the cloning process is complete.

_____ **Caution** _____

Typographical errors and inserting attribute-value pairs into the wrong item may result in a failure during the Installation Cloning process and may render the cloned system unusable.

Attribute-value pairs cannot contain blank spaces because blank spaces cause data validation errors. Be very careful to remove all blank spaces, especially at the end of a line. When you want to give an attribute a null value, make sure there is nothing (null) after the equal sign (`=`).

Do not modify or remove attributes that are prefixed with an underscore (`_`). These attributes, for example `_action=create`, are internal variables required by the Full Installation and Installation Cloning processes.

_____

It is recommended that you do not modify the original `install.cdf` file located in the `/var/adm/smlogs` directory of an installed system. Instead, make a copy of `install.cdf` and modify the copy. The original CDF should be retained in the `/var/adm/smlogs` directory because it contains information about the initial system installation that could be valuable for future troubleshooting.

The next three sections describe how to set the confirmation and kernel option attributes to run the cloning in an unattended fashion and how to modify host- and site-specific attributes to achieve uniqueness of the cloned system. Section 6.6.4 describes a common error to avoid when modifying the CDF.

6.6.1 Setting the CDF Confirmation Attribute to Eliminate User Intervention

By default, the installation process prompts the user to confirm whether or not the `install.cdf` file should be applied. To turn off this confirmation

and eliminate the need for user intervention, the CDF confirmation attribute determines whether user confirmation is required before the CDF is used to start an Installation Cloning process. This feature is set with the `prompt=` attribute-value pair in the `Inst_islinfo` item in the CDF. You must always manually add this attribute to the CDF because the installation interfaces do not provide the ability to set this value. Valid values are shown in Table 6–7.

Table 6–7: Valid Values for the prompt= Attribute

Value	Description
`prompt=yes`	The user will be asked to confirm the use of the `install.cdf` file to drive the installation process. This is the default value if this attribute-value pair is not defined or is null.
`prompt=no`	Bypasses the CDF confirmation question and begins an Installation Cloning process automatically when the Full Installation process finds an `install.cdf` file. This eliminates any user intervention at the system to be cloned.

A portion of an `install.cdf` file in Example 6–2 shows you where to include the `prompt=` attribute-value pair in the `Inst_islinfo` item:

Example 6–2: Adding the CDF Confirmation Attribute to the install.cdf File

```
install:

_item=Inst_islinfo
prompt=no
media_type=CDROM
server=cosmos
_action=create
srcloc=/ALPHA/BASE
```

6.6.2 Setting the Kernel Option Attribute to Eliminate User Intervention

The `kernel_option` attribute in the `Inst_cinstall` item controls the type of kernel components that are built into the tailored kernel and controls whether or not user intervention is required.

_____ **Note** _____

The tailored kernel build on the target system does not automatically include the optional kernel components that

were in the model system unless the kernel build type is set to `interactive` and the user intentionally selects them.

Valid values for the `kernel_option` attribute are shown in Table 6–8.

Table 6–8: Valid Values for the kernel_option= Attribute

Value	Description
`kernel_option=interactive`	Provides an interactive kernel build, which stops the cloning process to let the user select optional kernel components from a menu. You should set this value to `interactive` in order to select the same optional kernel components that are on the model system.
`kernel_option=mandatory`	Builds a kernel with all mandatory kernel components; no user intervention is required.
`kernel_option=all`	Builds a kernel with all mandatory and all optional kernel components; no user intervention is required.

A portion of the `install.cdf` file in Example 6–3 shows you where to include the `kernel_option=` attribute-value pair in the `Inst_cinstall` item. This example sets the value to `mandatory`, which automatically builds a kernel with mandatory components, which will not require any user intervention.

Example 6–3: Setting the Type of Kernel Build in the install.cdf File

```
install:
 _item=Inst_cinstall
 kernel_option=mandatory
 timeset=yes
  lang_env=C
 password=Bp2xAe46zVpUo
 timezone=New_York
 locality=America
 _action=create
 hostname=taurus
```

6.6.3 Setting Host- and Site-Specific Attributes

You must read this section if you are performing the Full Installation from CD-ROM; it is recommended to read this section if you are performing the Full Installation from a RIS server.

Setting host- and site-specific information such as the host name, geographic location and area, date, and time are not necessary for a RIS installation because these values are obtained from the RIS server automatically during the installation even if they are defined in the CDF. This statement is true for Full Installations from RIS and for Installation Cloning from RIS.

In the case of a stand-alone system installed from a CD–ROM, however, setting these values must be determined from the CDF that drives the Installation Cloning. If the CDF does not define these attributes, the values must be entered interactively during the software configuration phase of the Installation Cloning process that occurs after software has been loaded. If you want to eliminate the need for user intervention during the cloning process, you should define values for these attributes.

The host-specific attributes to be considered are:

- Host Name

  A system's host name is contained in the `hostname=` attribute-value pair in the `Inst_cinstall` item. Host names for model and target systems that exist on the same network must be unique, so you should change this value. If the `hostname=` attribute does not exist in the CDF, or if the value associated with this attribute is null, the installation process becomes interactive during the software configuration phase of the Installation Cloning process to request this information. If LSM configuration data is contained in the CDF, it is recommended that you set the `lsm_hostname=` attribute in the `Inst_lsm_global` item to the same host name specified by the `hostname=` attribute. See the *Installation Guide* if you need guidelines for choosing a proper host name.

- Password

  Be aware that an encrypted value in the `password=` attribute means that all systems to be cloned share the same `root` password with the model system. You may want to consider leaving this value null so that the installation process becomes interactive to request a `root` password. For security reasons, sharing passwords among systems is not recommended. If you choose to retain the encrypted password in the CDF, remember that the password came from the model system and you should change the password on that model system to protect it from unauthorized users. Because the value of the `password=` attribute must be encrypted, this value cannot be set manually. If you need to change

the password on the model system, the *Installation Guide* contains guidelines for choosing correct passwords.

The site-specific attributes to be considered are:

- Geographic Location and Area

  A system's geographic location and time zone are contained in the `locality=` and `timezone=` attribute-value pairs in the `Inst_cinstall` item. On a system already installed with the current version of the operating system, valid values for these attributes are located in the `/etc/zoneinfo` directory.

  Recent changes to locations and time zones to make them comply to industry standards have resulted in changes to this file. For example, `locality=US` and `timezone=Eastern` is now represented as `locality=America` and `timezone=New_York`. You can use either representation, but the new style is recommended. When you specify a value for a locality that is divided into time zones, you must choose a valid time zone for that location. The *Installation Guide* contains more information about locations and areas.

  If the `locality=` and `timezone=` attributes do not exist in the CDF, or if the value associated with these attributes is null, the installation process becomes interactive during the software configuration phase to request this information. A `locality=` attribute can be present without a `timezone=` attribute because not all geographic locations are divided into time zones. For example, the geographic location `Japan` does not have time zones. In that situation, the installation process recognizes the fact that Japan does not have time zones and bypasses the request for a time zone.

- Date and Time

  It is not possible to specify dynamic values such as date and time in an `install.cdf` file and still retain accuracy at the cloned system. The ability does exist, however, for the `install.cdf` file to indicate that the date and time have been set previously either by one of the installation interfaces or through a RIS installation cloning process. The method used to indicate whether or not the date and time have been set is the `timeset=` attribute-value pair in the `Inst_cinstall` item.

  Valid values for the `timeset` attribute are shown in Table 6–9.

Table 6–9: Valid Values for the timeset Attribute

Value	Description
`timeset=no`	The system date and time have not been set previously. The Installation Cloning process becomes interactive to request this information.
`timeset=yes`	The system date and time have been set previously. Therefore, it is possible by using the `timeset=` attribute set to `yes` to continue the installation in an unattended fashion, even if the system time had not been set. The value of date and time on the target system is undetermined until the first user logs in and sets the date and time to the proper values using the `date` command.

6.6.4 Common Error: Trailing Blank Space

While modifying a CDF, a common error is to include a trailing blank space after an attribute-value pair. If the validation process detects a trailing blank space in the CDF, a message similar to the following is displayed:

```
----------------------------------------
Some errors occurred:
SetItmAttr: invalid attribute value kernel_option=all
----------------------------------------
```

This error causes the installation process to stop. In the previous example, the validation process found a trailing blank space after the word `all` in the `kernel_option=all` attribute-value pair. The corrective action is to edit the CDF, remove the blank space, place the CDF back in the same location, and restart the installation process on the target system.

6.7 Optional Step 3: Creating and Positioning Other User-Supplied Files

If you want to go a step further and recreate customized features of the model system on the cloned system, create `preinstall`, `postload`, and `postreboot` files. The Full Installation has the ability to invoke user-supplied files to perform customizations during a Full Installation. To learn more about invoking these files during a Full Installation, see Chapter 5.

You can also clone the configuration from a model system onto a target system. If you want to fully configure as well as install a target system, see Chapter 7 for information about how to create and apply a `config.cdf` file during a Full Installation.

The user-supplied files and `config.cdf` file are searched for in the same locations and order as the `install.cdf` file, which makes combining these features easy to do. If you want to take advantage of configuration cloning and invoke user supplied scripts to further customize the cloned system, see Chapter 5 and Chapter 7, respectively.

To continue with the Installation Cloning procedure, go to Section 6.8.

6.8 Step 4: Copying the CDF to the Right Location

The next step in the process is to copy the modified CDF to the right location. Because the `install.cdf` file must be located in and is searched for in the same locations as the user-supplied files and the `config.cdf` file for configuration cloning, the actual steps in the copy process are not duplicated here.

Based on your media requirements, decide where you want to move the `install.cdf` file and go to the referenced section shown in Table 6–10 for step-by-step instructions for copying the file there.

Table 6–10: Acceptable Locations of the install.cdf File

Search Order	Location	Copy Instructions Located In
1	On a diskette in diskette drive `floppy0` or `floppy1`.	Section 5.8.1
2	In the *profile_set* subdirectory of the `/var/adm/ris/clients/sets` directory on the RIS server.	Section 5.8.2
3	In the `/var/tmp` memory file system (MFS) on the system to be cloned.	Section 5.8.3
4	In the `/isl` directory on the distribution media (local CD–ROM or extracted RIS area).	Section 5.8.4

6.9 Step 5: Beginning a Full Installation on the Target System

When you have verified the contents of the `install.cdf` file and copied it to the right location based on your media requirements, begin a Full Installation on the target system as described in the *Installation Guide*. When the installation process finds the `install.cdf` file, an Installation Cloning process begins on the target system. If the Full installation process finds correctly named and placed user-supplied files or a `config.cdf` file, those files are executed as well.

If you start a Full Installation on the target system and the Full Installation graphical or text-based interface is displayed instead, the `install.cdf` file was not located in one of the four supported locations. In that case, copy the `install.cdf` file to one of the locations shown in Table 6–1, and restart the Full Installation on the target system.

7

Configuration Cloning

This chapter provides the following information:

- An overview of Configuration Cloning (Section 7.1)

- What you have to do to make it happen (Section 7.2)

- A description of the format and contents of the `config.cdf` file (Section 7.3)

- A sample `config.cdf` file (Section 7.3.1)

- Restrictions pertaining to the system where the `config.cdf` file was generated and the type of systems that are suitable to be cloned from it (Section 7.4)

- Step by step procedures, which describe how to save configuration information to a CDF, edit the CDF, and start the cloning process (Section 7.5)

7.1 Overview

Network services, naming services, printers, internet services, and electronic mail delivery must be configured to make users of a newly installed system productive and able to communicate with other systems and users. On a newly-installed system, system configuration tasks are performed from the Quick Setup application, the Custom Setup checklist, or the SysMan Menu. Before Configuration Cloning was available, configuration tasks had to be performed at each system individually.

Configuration Cloning lets you duplicate the configuration from an already-configured system onto one or more target systems, eliminating the need to perform the configuration tasks as a separate operation.

It is recommended that you use Configuration Cloning to clone systems with similar hardware and functions. Configuration cloning is ideal for cloning the same class or type of machine (for instance, all workstations or all servers). The criteria for determining whether cloning is suitable for your situation is if the systems are of the same type and have the same number and type of network adapters. Instead of configuring each system individually, you configure a single system, then replicate that model configuration among other systems of the same class or type.

You can also use the Configuration Cloning process to restore the configuration of a system if the current configuration becomes corrupted or if you want to go back to a previous configuration.

To achieve a fully automated installation and configuration of another system, Configuration Cloning can be combined with Installation Cloning to completely eliminate the need to perform these tasks at the target system itself. Installation cloning requires a separate `install.cdf` file that is created automatically when a system is installed. See Chapter 6 for more information about Installation Cloning.

7.2 How Does it Happen?

When a system you designate to be a model system has been fully configured the way you want it, you use the `sysman -clone -save` command to save a snapshot of the system configuration data into a configuration description file (CDF). This file is named `config.cdf`, and it is saved in the `/var/adm/smlogs` directory by default. The information saved in the `config.cdf` file is used to duplicate the same configuration on similar systems. You may edit the `config.cdf` file and change any value (except the checksum), but there are certain host-specific attributes (such as host name and IP address) that must be edited to retain a unique network identity for the cloned systems.

The `config.cdf` file can be applied automatically to another system during a Full Installation process or can be applied manually after the system to be cloned is installed but is not yet configured. The following list summarizes your options:

- To manually clone one system at a time, edit the `config.cdf` file to set host- and site-specific attributes, copy the CDF to the target system, and manually apply the CDF to an installed target system using the `sysman -clone -apply` command.

- To clone one system during a Full Installation, edit the `config.cdf` file to set host- and site-specific attributes, and position the `config.cdf` file in one of four acceptable locations.

  When you begin a Full Installation on a system you want to clone, the installation process looks for the `config.cdf` file in the order shown in Table 7–1.

Table 7–1: Search Order for the config.cdf File

Search Order	Location
1	On a diskette in diskette drive `floppy0` or `floppy1`.

Table 7–1: Search Order for the config.cdf File (cont.)

Search Order	Location
2	In the `/var/adm/ris/clients/sets/`*`profile_set`* subdirectory on the RIS server. During the RIS client registration process, the target system must be registered to the *`profile_set`* directory that contains the `install.cdf` file you want to use.
3	In the `/var/tmp` memory file system (MFS) on the system to be cloned.
4	In the `/isl` directory on the distribution media (local CD–ROM or extracted RIS area).

- To clone multiple systems during a Full Installation, write a `postload` script to define each system to be cloned and set values for host- and site-specific attributes in the `config.cdf` file. The `config.cdf` file and the `postload` script are executed automatically by a Full Installation when they are placed in the right locations.

_____ **Note** _____

Configuration Cloning is not just available at installation time; Section 7.11.2 has instructions on how to apply a configuration to an already-running system.

_____

If you want more information, Figure 5–2 in Chapter 5 contains a theory of operation, which describes the points at which user-supplied files and CDFs are invoked during a Full Installation.

7.3 Configuration CDF Format and Contents

The `config.cdf` is organized by configuration component. Each component contains several groups in which there are one or more attribute-value pairs to define the configuration data. An equal sign (`=`) separates each attribute and its associated value.

Table 7–2 describes the components in the `config.cdf` file.

Table 7–2: Components in the config.cdf File

Component Name	Description
bindconfig	Contains Domain Name Service (DNS) configuration data. Cloning of DNS clients is supported. DNS servers cannot be cloned.
internetServices	Starts and stops the process that manages participation on the Internet.
mail	Contains information about the mail server configuration. Only cloning of a mail client is supported.
networkAdapters	Defines the type of network adapter that is configured.
networkServices	Specifies whether the system is a member of a cluster.
networkedSystems	Defines, among other things, the hosts in the `/etc/hosts` database.
networks	Defines the data in the `/etc/networks` file, which contains information about known networks that comprise the DARPA.
nfs_export	Defines whether the system is configured to export local file systems across the network.
nfsconfig	Defines the Networked File Systems (NFS) client and server settings.
nisconfig	Contains information about the Network Information Services (NIS) configuration.
ntpconfig	Contains Network Time Protocol (NTP) server information.
printcap	Contains information about defined printers similar to the information found in the `/etc/printcap` file.
remoteWhoServices	Manages the `rwhod` daemon (the server that maintains the databases used by the `rwho` and `ruptime` daemons).
routing	Defines the routing configuration.

7.3.1 Sample Configuration CDF

Example 7–1 shows the bindconfig component in a `config.cdf` file. Comment lines begin with a number sign (#). The last piece of information within a `config.cdf` file component is Group: componentid, which contains internal information required by the cloning process for validation purposes.

For space considerations, only a small portion of a `config.cdf`
file is shown in the example. If you want to see an entire
`config.cdf` file, use the `sysman -clone -save` command
on a configured system to save configuration information to the
default location `/var/adm/smlogs/config.cdf`. Then, use a
text editor or another method of your choice to view the file.

Example 7–1: Portion of a config.cdf File

```
CHECKSUM=56823

#
# CDF Created: Fri Aug 23 11:34:30 EST 2002
#
#
# Component: bindconfig
#
#
# Group: bind
#
/bindconfig/bind:
        configured=YES
        bindtype=CLIENT
#
# Group: resolver
#
/bindconfig/resolver:
        change_hostname=NO
        domain=mydomain.com
        precedence=first
#
# Group: ns
#
/bindconfig/ns:
        cdf_record=00000001
        ipaddress=16.29.221.1
        hostname=libra.mydomain.com
#
# Group: search
#
/bindconfig/search:
#
# Group: componentid
#
/bindconfig/componentid:
        manufacturer=Hewlett-Packard Company
        product=Domain Name Service Configuration
        version=DNS-1.1.4.17
        serialnumber=1.1.4.17
        installation=19990203184929.000000-300
        verify=7
#
# CDF Created: Fri Aug 23 11:34:47 EST 2002
#
```

7.4 Configuration Cloning Restrictions

It is recommended to adhere to the following restrictions when selecting or generating a `config.cdf` file to clone other systems:

- Configuration cloning is not supported between different releases of the operating system. If you want to clone a configuration onto a system running Version 5.1B, the `config.cdf` file you use must be created from a model system that is installed and configured with Version 5.1B.

- Software patches are not cloned. The reason for this restriction is that installed patches change serial numbers, which will cause validation errors when the `config.cdf` file is applied to the target system. The `config.cdf` file you create must be created from a model system that has not been patched.

  If the model system has already been patched, there are two alternatives you can follow, although they are not recommended:

  - Create the `config.cdf` from the patched system anyway. Use the `-f` option when applying the `config.cdf` file to the target system to force the `config.cdf` onto the target system. For example, at the target system, enter:

    ```
    # sysman -clone -apply -f config.cdf
    ```

    Then, at the newly-cloned target system, create another `config.cdf` to use for cloning additional systems. The new `config.cdf` will contan the correct checksums, and cloning other systems from this file will be valid.

    _____ **Caution** _____

    The `-f` option to force a CDF onto a target system is typically used for internal debugging purposes. Be aware that should the `config.cdf` contain additional problems other than the patched model system issue, forcing the CDF could cause problems on the target system from which there may be no recovery.

    _____

  - Use the Update Installation process to update the patched model system to the next version of the operating system, if the next version has been released, thereby bypassing the patch issue.

- Software licenses are not cloned. The product authorization codes (PAKs) that you installed and registered on the original system are not cloned on the target system. You must run the License Manager application or the `lmfsetup` script on the cloned system to install and register software licenses.

- A `config.cdf` file that was created on a system with multiple network interfaces (adapters) should not be used to clone the configuration on a system with only one network adapter. If you decide to do this, be careful to remove the extra adapter information from the CDF or the CDF will fail validation.

- Configuration cloning is best suited for cloning similar class machines. It is recommended to use a CDF created from a server class system to clone servers and to use a CDF from a workstation class system to clone other workstations.

  You do not want to clone a workstation class system with a CDF that was taken from a machine that is configured to be any type of server because the CDF contains too much server setup information that is not necessary on a workstation. This unnecessary information can prove harmful to the network. For example, if a system is configured to be the primary IP router for your site, you do not want to clone another one.

- Configuration Cloning is not supported in a cluster environment. Do not save the configuration of a cluster member because individual cluster members share disks, and the information saved in the CDF relates to the entire cluster, not to individual members.

7.5 Summary of Configuration Cloning Procedures

The following list summarizes the steps required to clone the configuration from a model system to one or more target systems.

1. Use the `sysman -clone -save` command to save system configuration information from a configured model system to a `config.cdf` file (Section 7.6).

2. Modify the `config.cdf` file to set host-specific attributes, and optionally, site-specific attributes (Section 7.7).

3. Validate the CDF before you use it to clone another system (Section 7.8).

4. To set up Configuration Cloning of multiple systems during a Full Installation process, optionally create a `postload` script to dynamically set host-specific values (Section 7.9).

5. Copy the `config.cdf` file to the right location depending upon how it will be applied to the target system or depending upon your distribution media requirements (Section 7.10).

6. Use the Full Installation process or the `sysman -clone -apply` command to apply the `config.cdf` file to the target system (Section 7.11).

7.6 Step 1: Saving Configuration Information to a Configuration CDF

Unlike Installation Cloning, where the `install.cdf` file is created automatically, you must intentionally save configuration data into a `config.cdf` file. The reason the process is not automated is because without human intervention there is no automated way to be certain when a system configuration is complete. You can save configuration data into a CDF at any time after a system is installed, as often as you like. When configuration data is saved, any attribute that has no associated value will not be defined in the `config.cdf` file.

If you intend to clone the configuration of many systems using one `config.cdf` file, you should consider installing and configuring one system to use as the model system. That way, except for host-specific attributes, there is very little CDF editing required.

On a system that is configured the way you want it, use the `sysman` command with the following flags to save configuration data into a CDF:

sysman -clone -save [*filename*]

This command saves the current system configuration to `/var/adm/smlogs/config.cdf` unless you specify a different path and file name. Be aware that if you intend to clone a system configuration during a Full Installation process, the installation process searches for a file called `config.cdf`. A configuration CDF with any other name is not recognized.

The `sysman -clone -save` command also validates the CDF which is evidenced by the checksum number at the top of the `config.cdf` file. When you apply the CDF to clone a target system, the checksum number determines the integrity of the `config.cdf` file to protect it from being rendered unusable by incorrect modifications. It is very important that you do not modify this checksum.

To list the status of all configured components, use the `sysman` command as follows:

sysman -clone -list

7.7 Step 2: Modifying Host- and Site-Specific Attributes in the Configuration CDF

The `config.cdf` file contains host- and site-specific attributes that must be edited to ensure each cloned system is uniquely configured on the network. It is suggested that you do not modify the original `config.cdf` file located in the `/var/adm/smlogs` directory. Instead, make a copy of the file and modify the copy. The original CDF should be retained in the

`/var/adm/smlogs` directory because it contains information about the model system configuration that could be valuable for future troubleshooting or restoring the original configuration. You should save a backup of the modified `config.cdf` file as a record of your modifications.

_____ **Note** _____

The top of the file contains a checksum number, which is used for validation purposes. Do not change this number because the cloning process will not be able to validate the CDF after which it cannot be used to clone another system.

_____

If you plan to edit this file in any way, it is recommended that you do not modify any attribute or value that you do not understand. If you see a host- or site-specific value that you know is incompatible with the target system, for instance, the host name of the model system, you should change it. It is recommended that you only modify the attributes and values that are specifically described in this chapter.

Table 7–3 lists the host-specific attributes that must be modified before the configuration is applied so that cloned systems maintain a unique identity on the network. The third attribute should be considered if the hardware or processor type on the system to be cloned does not exactly match the system where the `config.cdf` file was generated.

Table 7–3: Host-Specific Attributes in the config.cdf File

Host-Specific Attribute	Description
`systemName=`	Sets the name of the system as it is known on the network. This attribute must be modified because each system must have a unique name. This attribute is located in the `networkAdapters` component in the `interface` group, and depending upon which other components are configured, the name may be specified in other attributes. Make sure to search for and change all occurrences of the system name.

Table 7–3: Host-Specific Attributes in the config.cdf File (cont.)

Host-Specific Attribute	Description
`networkAddress=`	Sets the unique internet protocol (IP) address of the client system as it is identified on a network. This attribute is located in the `networkAdapters` component in the `interface` and is located in the `networkedSystems` component in the `hostMappings` group.
`devName=` and `type=`	Defines the network adapter attached to the system. This value is modified only if the network adapter on the system to be cloned is different from the one defined in the `config.cdf` file. Valid values for this attribute include `tu0` for Tulip devices, `ln0` for Lance devices, `fddi0` for FDDI devices, and `ee0`. To determine the network adapters available on the system to be cloned if the system is not yet running the operating system, issue the `show dev` and `show config` commands at the console mode prompt (>>>). If the system to be cloned is running the operating system, use the following command to determine the network adapter device (NETDEV_0): `rcmgr mget	grep NET`

Section 7.7.1 describes the site-specific attributes you may want to modify; Section 7.7.2 describes how to use the CDFMODE attribute to modify component groups with more than one record.

7.7.1 Optional Step: Modifying Site-Specific Attributes in the Configuration CDF

The `config.cdf` file contains many network related attributes that you may want to consider editing if the systems you are cloning have different requirements. For example, you may be cloning all systems in a particular department or site that require a special network configuration. Attributes to consider modifying are the values for DNS server, netmask, domain name, type of network routing, network time protocol (NTP) server name, default printer, and printer and mail client configuration.

7.7.2 Using the CDFMODE Attribute for Component Groups with Multiple Records

The CDFMODE attribute applies only to component groups containing multiple records. CDFMODE is a global attribute that applies to all `cdf_record` labels within a component group throughout the entire CDF. An example of a component group containing multiple records is the `hostMappings` group within the `networkedSystems` component. This group represents the data in the `/etc/hosts` file.

The configuration CDF begins each record within a multiple-record group with the label `cdf_record=` to show that there is more than one record within the group. Example 7–2 shows an example of a host group with multiple records.

Example 7–2: Example of a Component Group with Multiple Records

```
# Component: networkedSystems
#
# Group: hostEquivalencies
#
/networkedSystems/hostEquivalencies:
#
# Group: hostMappings
#
/networkedSystems/hostMappings:
        cdf_record=00000001
        networkAddress=127.0.0.1
        systemName=localhost
        cdf_record=00000002
        aliases=aries
        networkAddress=16.29.221.1
        systemName=aries.mydomain.com
        cdf_record=00000003
        networkAddress=16.29.221.15
        systemName=pluto
        cdf_record=00000004
        aliases=virgo
        networkAddress=16.29.221.27
        systemName=virgo.mydomain.com
#
#
```

Table 7–4 lists and describes valid values for the CDFMODE attribute.

Table 7–4: Values for the CDFMODE Attribute

Value	Description
MERGE	Merges the component/group data with existing data on the target system. Duplicate data is ignored. This is the default behavior for the entire configuration CDF if CDFMODE is not specified.
APPEND	Appends the component and group data to existing configuration data on the target system. Duplicate data is not ignored.
REPLACE	Replaces the component and group data on the target system with the data in the configuration CDF. Data existing on the target system, but not in the CDF will be removed. Data that exists in the configuration CDF, but not on the target system will be applied. Data existing in both the CDF and the target system will be modified accordingly.

The CDFMODE attribute can be placed outside of any component contained within a config.cdf file.

––––––––––––––––––––––– **Note** –––––––––––––––––––––––

As shown in Example 7–3, the CDFMODE= attribute must start in the first column of the line on which it is located; do not indent it by inserting spaces or tab characters.

––

The CDFMODE value is in effect until another CDFMODE= statement is reached. Example 7–3 shows the use of the CDFMODE attribute. In this example, the default mode MERGE is in effect until CDFMODE=REPLACE is encountered. This causes the entire /etc/hosts file to be replaced with the contents of the CDF. After that, CDFMODE=MERGE (which is the default mode) is encountered, and the remaining attributes are merged onto the system to be cloned.

Example 7–3: Inserting the CDFMODE Attribute into a CDF

```
CDFMODE=REPLACE
#
# Component: networkedSystems
#
# Group: hostEquivalencies
#
/networkedSystems/hostEquivalencies:
#
# Group: hostMappings
#
/networkedSystems/hostMappings:
        cdf_record=00000001
        networkAddress=127.0.0.1
        systemName=localhost
        cdf_record=00000002
        aliases=host1
        networkAddress=16.29.221.2
        systemName=host1.mydomain.com
        cdf_record=00000003
        networkAddress=16.29.221.16
        systemName=host2
        cdf_record=00000004
        aliases=host3
        networkAddress=16.29.221.28
        systemName=host3.mydomain.com
#
CDFMODE=MERGE
```

7.8 Step 3: Validating the Modified CDF

After you have modified the `config.cdf` file, it is recommended that you validate its contents by entering the following command:

sysman -clone -validate [*filename*]

If you do not supply a file name, the command assumes the file name is `/var/adm/smlogs/config.cdf`. This command uses the checksum at the top of the file to determine whether modifications you made have rendered the file unusable to clone other systems. The validation process attempts to isolate the component group in which the error occurred and a message is displayed. If validation errors occur, obtain an unmodified original copy of the `config.cdf` file and begin again.

7.9 Optional Step 4: Creating a Script to Clone Multiple Systems During a Full Installation

To use a `config.cdf` file during a Full Installation to clone many systems at the same time, it is recommended that you create a single, representative `config.cdf` file with the values of host-specific values set to variables. Then, manually enter a value for the variables before applying them to a system, or you can create a script to be invoked during a Full Installation process to dynamically modify the variables in the `config.cdf` file before it is applied to the target system.

The Full Installation process is designed to look for specific scripts during a Full Installation in the same places where the `config.cdf` file is located. If the installation process finds a script named `postload` in the right location, the `postload` script is executed after software subsets are loaded but before kernel build takes place. This is also the phase where the `config.cdf` file is applied. Appendix B contains an example of just such a `postload` script, which shows how to create variables for host-specific values. The `postload` file and the `config.cdf` file must reside in the same location.

Section 7.10.3 describes the four locations where the installation process looks for the `postload` and `config.cdf` files.

7.10 Step 5: Copying the Configuration CDF to the Right Location

The location of the `config.cdf` file depends on whether you are cloning the system configuration during a Full Installation or if you are cloning the configuration on a system that is already installed but not yet configured:

• If you intend to apply the `config.cdf` file manually on a single system or to one system at a time, copy the `config.cdf` file to a diskette as

shown in Section 7.10.1 or set up a temporary network to move the
`config.cdf` file to the system to be cloned as shown in Section 7.10.2.

- If you want to clone a system configuration during a Full Installation, see
  Section 7.10.3 for the list of supported locations that are searched. These
  locations are the same places where the installation process searches for
  user-supplied scripts and the `install.cdf` file.

7.10.1 Copying the Configuration CDF to a Diskette

Use the following procedure to copy the `config.cdf` file to a diskette:

1. Format, write a new disk label, and create a new file system on a
   diskette:

   ```
   # fddisk -fmt /dev/rdisk/floppy0a
   # disklabel -wr floppy0 rx23
   # newfs /dev/rdisk/floppy0c
   ```

2. Mount the diskette drive on the `/mnt` mount point:

   ```
   # mount /dev/disk/floppy0 /mnt
   ```

3. Save the configuration data to the diskette using the file name
   `config.cdf`:

   ```
   # sysman -clone -save /mnt/config.cdf
   ```

After the `config.cdf` file is copied to the diskette, insert and mount
the diskette into the diskette drive of the system to be cloned. Go to
Section 7.11.2 for instructions about applying the `config.cdf` file manually
to automatically configure the system.

7.10.2 Copying the Configuration CDF to a System That Is Not Connected to the Network

If the system you want to clone is installed, but is it not configured and
does not have a diskette drive, there is no way to copy the `config.cdf` file
there unless a temporary network connection is created. Use the following
procedure to set up a temporary network and then use the file transfer
protocol (FTP) to copy the `config.cdf` file to the system to be cloned. You
will need to know the Internet Protocol (IP) address of the system from
which you are copying the CDF.

The following procedure is performed on the system that is already installed
but is not yet configured and assumes that you have root privileges on
both machines:

1. On the system to be cloned, issue the following command to determine
   the network adapter and IP address:

```
# ifconfig -a
```

Look for the line that contains the IP address of the machine to know which network interface is the one to configure.

2. Temporarily connect the network adapter to the network:

```
# ifconfig adapter_name IP_address
```

3. Check the network connection you just set up to make sure your system recognizes the model system where the CDF is located:

```
# /usr/sbin/ping -c2 model_system_name
```

4. Change to the following directory, which is the default location of the `config.cdf` file:

```
# cd /var/adm/smlogs
```

5. Use the file transfer protocol (FTP) to connect to the model system where the `config.cdf` file was generated. Because the domain name service (DNS) is not running, specify the IP address of the model system rather than the system name:

```
# ftp model_system_IP_address
```

6. Log in to FTP as the user `root`, and enter the root password.

7. Change to the directory where the `config.cdf` file is located. This example assumes the file is in the default location:

```
ftp> cd /var/adm/smlogs
```

8. Copy (get) the `config.cdf` file to move it to the system to be cloned:

```
ftp> get config.cdf
```

9. Close the connection and exit out of FTP:

```
ftp> bye
```

The `config.cdf` file is now located in the `/var/adm/smlogs` directory of the system to be cloned. Go to Section 7.11.2 for instructions about applying the `config.cdf` file manually to configure the system.

7.10.3 Copying the Configuration CDF to Distribution Media

To clone a configuration during a Full Installation, copy the `config.cdf` file from the original model system to one of the four locations shown in Table 7–5. When you begin a Full Installation on a system you want to clone, the installation process looks for the `config.cdf` file in the order shown.

Because the `config.cdf` file is located in and is searched for in the same locations as the user-supplied files and the `install.cdf` file for Installation Cloning, the actual steps in the copy process are not duplicated here. Based on your media requirements, decide where you want to move the

`config.cdf` file and go to the section shown in column 3 for step-by-step procedures for copying the file.

Table 7–5: Acceptable Locations of the config.cdf File

Search Order	Location	Copy Instructions Located In
1	On a diskette in diskette drive `floppy0` or `floppy1`.	Section 5.8.1
2	In the *profile_set* subdirectory of the `/var/adm/ris/clients/sets` directory on the RIS server.	Section 5.8.2
3	In the `/var/tmp` memory file system (MFS) on the system to be cloned.	Section 5.8.3
4	In the `/isl` directory on the distribution media (local CD–ROM or extracted RIS area).	Section 5.8.4

When you have copied the `config.cdf` file to the right location, go to Section 7.11.1 to apply the CDF to a target system.

7.11 Step 6: Applying the Configuration CDF to a Target System

When applying a `config.cdf` file to a target system, any attribute not defined in the CDF will remain unchanged or unconfigured on the target system.

Depending upon your needs, you can apply the `config.cdf` file to a system in the following ways:

- The Full Installation applies the CDF if the file is in the right place to begin a Configuration Cloning (Section 7.11.1).

- You can manually apply the CDF to the target system to begin a Configuration Cloning (Section 7.11.2).

- You can manually apply the CDF to restore the configuration on a system with a corrupted configuration (Section 7.11.3)

7.11.1 Applying the Configuration CDF During a Full Installation

When the Full Installation process finds a `config.cdf` file in any one of the locations shown in Section 7.10.3, the CDF is applied to the system after file systems are created, software is loaded, and the system is rebooted off the newly installed system disk. The newly installed system will be configured exactly as defined in the CDF.

7.11.2 Applying the Configuration CDF Manually to a Running System

When a `config.cdf` file is copied to a diskette, which has been inserted into the diskette drive and mounted or has been copied to the `/var/adm/smlogs` directory on a system to be cloned, use the `sysman` command with the following syntax to apply the configuration to the system:

sysman -clone -apply [*filename*]

If you do not specify a full path and file name, the `sysman` command assumes the default location and file name `/var/adm/smlogs/config.cdf`. If you placed the `config.cdf` file in a location other than `/var/adm/smlogs` or called the file something other than `config.cdf`, you must specify the path and exact file name on the command line.

7.11.3 Restoring a System Configuration Using the Configuration CDF

To restore a corrupted system configuration or to simply go back to a previous configuration that is stored in a `config.cdf` file, make sure the right `config.cdf` file is in the `/var/adm/smlogs` directory, and issue the following command:

sysman -clone -apply [*filename*]

If the configuration CDF is in a location other than `/var/adm/sm-logs/config.cdf`, you must specify the full path in the command line.

A

Installation CDF Attribute–Value Pairs

The attribute-value pairs within individual items differ as a result of the distribution method (CD–ROM or RIS) that was used to perform the initial Full Installation of the model system. This appendix describes the contents of the following items in the `install.cdf` file:

- `Inst_islinfo` (Section A.1)
- `Inst_filesystem` (Section A.2)
- `Inst_disklabel` (Section A.3)
- `Inst_lsm_global` (Section A.4)
- `Inst_lsm_disks` (Section A.5)
- `Inst_subsets` (Section A.6)
- `Inst_cinstall` (Section A.7)

_____ **Caution** _____

Only experienced system administrators should modify the attributes-value pairs in the `install.cdf` file. Typographical errors and inserting attribute-value pairs into the incorrect item may result in serious corruption on the cloned system and may render the system unusable.

In addition, attribute-value pairs cannot contain blank spaces because blank spaces cause data validation errors. Be very careful to remove all blank spaces especially at the end of a line. When you want to give an attribute a null value, make sure there is nothing (null) after the equal sign (`=`).

Do not modify or remove attributes that are prefixed with an underscore (`_`). These attributes, for example `_action=create`, are internal variables required by the Full Installation and Installation Cloning processes.

_____

A.1 Attributes in the Inst_islinfo Item

Table A–1 defines the attributes in the `Inst_islinfo` item in the CDF. The `Inst_islinfo` item is used to convey the system state before the start of the installation process.

Table A–1: Attribute Definitions in the Initial Subset Load (Inst_islinfo) Item

Attribute	Valid Values	Description
`client=`		**Do not modify**. This attribute is valid only for RIS Full Installations (not Installation Cloning) and specifies the RIS client name of the system that was cloned. The RIS client name is determined automatically as a result of the `bootp` request to the server.
`clone=`		**Do not modify**. This attribute is inserted automatically into the CDF as a result of the Installation Cloning process and is valid only during the Installation Cloning process.
`force_ccii=`	yes no	Setting this attribute to `yes` invokes a text-based (character cell) installation user interface on systems that have graphics capabilities. The default value is Null, which corresponds to `no`.
`media_type=`	REMOTE CDROM	This attribute is used by the Full Installation and Installation Cloning processes to indicate the type of distribution media for the current installation. This is the only required entry in the `Inst_islinfo` item. Edit this attribute when the type of distribution media used for the initial installation is different from the Installation Cloning that is to take place.

Table A–1: Attribute Definitions in the Initial Subset Load (Inst_islinfo) Item (cont.)

Attribute	Valid Values	Description
prompt=	yes no	This attribute is used by the Installation Cloning process to indicate whether the start of the cloning requires a confirmation response from the user. This attribute must be entered manually into the CDF for an Installation Cloning process because the installation interfaces do not provide the ability to insert this attribute into the CDF. A value of yes indicates that the process should prompt for confirmation to use the CDF. A value of no indicates that the Installation Cloning process automatically should use this CDF and bypass the confirmation question. If this attribute is not included in the CDF, the default is prompt=yes. Setting the attribute to no should be used with caution because the Installation Cloning begins as soon as the installation process detects a CDF. If you wanted to boot the system from the distribution media and perform system management or disk maintenance tasks, for example, you would not want the Installation Cloning to begin immediately.
risdir=		**Do not modify**. This attribute is specific to RIS Full Installations and is set automatically to the base RIS directory of the product environment to which the client system is registered.
server=		**Do not modify**. This attribute is specific to RIS full and cloning installations and identifies the RIS server to which the client system is currently registered.

Table A–1: Attribute Definitions in the Initial Subset Load (Inst_islinfo) Item (cont.)

Attribute	Valid Values	Description
server_locality=		**Do not modify**. This attribute is specific to RIS Full Installations and specifies to the installation interfaces the current geographic location. This attribute is controlled by the `locality=` attribute in the `Inst_cinstall` item.
server_timezone=		**Do not modify**. This attribute is specific to RIS Full Installations and specifies to the installation interfaces the current geographic time zone. This value is set automatically during a RIS Full Installation.
srcloc=	/ALPHA/BASE *RISserver:*	This attribute is not used by either the Full Installation or Installation Cloning processes; it is used by the operating system for internal purposes. This attribute identifies the location of the software to load. For RIS installations, this value specifies the server name (appended with a colon). For CD–ROM installations, this value is the directory path `/ALPHA/BASE`. Do not modify this attribute unless the `media_type` attribute is changed because this value must be consistent with the value of `media_type`.
timeset=	0 1	**Do not modify**. This attribute applies to Full Installations and indicates to the installation interfaces whether the date and time on the target system have been set successfully and whether the date and time can be displayed during the installation. If the date and time have not been set and will not be displayed during the installation process, this attribute is set to `0`. If the date and time have been set successfully and will be displayed during the installation process, this attribute is set to `1`. If you modify this entry, it will revert back to its original setting because it is controlled by internal installation processes.

A.2 Attributes in the Inst_filesystem Item

Table A–2 defines the attributes in the `Inst_filesystem` item in the CDF. The `Inst_filesystem` item conveys information about the number and type of file systems that are to be created on the cloned system. At a minimum, there must be at least four file system items to describe the `/`, `/usr`, and `/var` file systems and one `swap` area. Except where noted, you can modify all attribute-value pairs in this item, although it is not recommended.

Table A–2: Attribute Definitions in the File System (Inst_filesystem) Item

Attribute	Valid Values	Description
`name=`	root usr var i18n swap1 swap2	**Required**. Specifies the name of the file system to be made. There can only be one item each for `/`, `usr`, `var`, `i18n`, `swap1`, and `swap2`.
`file_system_type=`	ufs advfs swap	**Required**. Specifies the file system type to be created for the named file system. If the value of the `name=` attribute is `swap1` or `swap2`, the value of this attribute must be `swap`. The value `swap` is not allowed as a value for any other `name=` attributes.
		Caution: Be aware that changing this value from `ufs` to `advfs` may cause errors on the cloned system because the software subsets necessary to support an Advanced File System (AdvFS) may not be defined in the CDF and will not be installed on the cloned system. Therefore, the file system will be unreadable.
		Do not change this value to `advfs` unless other file systems have been set by the installation process to `advfs` or the required AdvFS software subsets are present in the `ss_names=` attribute in the `Inst_subsets` item. Subset descriptions are documented in the *Installation Guide*.

Table A–2: Attribute Definitions in the File System (Inst_filesystem) Item (cont.)

Attribute	Valid Values	Description
disk_name=	dsk*n*	**Required**. Specifies the disk name for the named file system as it is known to the operating system (for example, dsk0). The value in this attribute must be consistent with the value in the disk_type= attribute. If you change this attribute, you must validate the change with respect to the disk_type= attribute. Enter the following command to determine the disk type: disklabel dsk*n*|grep disk For example, if you change this value to disk_name=dsk1, you must determine the type of disk at dsk1. If it is an RZ58 type of disk, make sure the value of the disk_type= attribute is RZ58.
disk_number=	*n*	This attribute is set by the Full Installation process and is not used by the cloning process.
disk_type=	RZ*nn[n][A-Z]* RA*nn* HSZ*nn*	**Required**. Indicates the type of disk for the specified disk_name (for example RZ26). The value in this attribute must be consistent with the disk_name= attribute. See the disk_name= attribute for more information.
partition=	[a-h]	**Required**. Specifies the disk partition on which the named file system will be created. Valid values are the letters a through h inclusive. The / (root) file system must always be located on partition a. _____ **Caution** _____ If you change the value in this attribute for any file system other than /, make sure the partition you choose does not overlap another partition that contains an active file system. _____

Table A–2: Attribute Definitions in the File System (Inst_filesystem) Item (cont.)

Attribute	Valid Values	Description
controller_type=		This attribute identifies the controller type to which the specified disk for the named file system is connected. During a Full Installation, this value is provided automatically for informational purposes. During an Installation Cloning process this attribute is not used, and can be omitted from the CDF.
controller_number=		This attribute identifies the controller number to which the specified disk for the named file system is connected. During a Full Installation, this value is automatically provided for informational purposes. During an Installation Cloning process this attribute is not used, and can be omitted from the CDF.

A.3 Attributes in the Inst_disklabel Item

Table A–3 defines the attributes in the Inst_disklabel item in the CDF. The Inst_disklabel item conveys the size and offset of each partition to be created on the cloned system. An Inst_disklabel item is created for each disk on which you installed a file system or swap area. For example, if you installed / on dsk0, and /usr on dsk1, an Inst_disklabel item will be created for dsk0 and dsk1. All other disks on the target system will be unaffected during cloning.

Table A–3: Attribute Definitions in the Disklabel (Inst_disklabel) Item

Attribute	Valid Values	Description
name=	dsk*n* in usr_domain in usr	**Required. Do not modify.** This attribute specifies the disk associated with the size and offset definitions within the Inst_disklabel item. An Inst_disklabel attribute is created for each disk configured in the model system. A value of in usr_domain means that the var and i18n directories with an AdvFS file system type are in the usr_domain. A value of in usr means that the var and i18n directories with a UFS file system type are in the /usr file system.

Table A–3: Attribute Definitions in the Disklabel (Inst_disklabel) Item (cont.)

Attribute	Valid Values	Description
`[a-h]_size=`	*n* (blocks)	**Required. Do not modify.** This attribute specifies the size of every partition on configured disks. All partitions are defined, whether or not they are used for a file system. The `Inst_filesystem` item defines the file systems.
`[a-h]_offset=`	*n* (blocks)	**Required. Do not modify.** This attribute specifies the offset of every partition on configured disks. All partitions are defined, whether or not they are used for a file system. The `Inst_filesystem` item defines the file systems.

A.4 Attributes in the Inst_lsm_global Item

Table A–4 defines the attributes in the `Inst_lsm_global` item in the CDF. The `Inst_lsm_global` item conveys the size of the Logical Storage Manager (LSM) private region to be created on all configured disks and defines an LSM host name. The `Inst_lsm_global` item is created only if you chose to install into LSM volumes.

Table A–4: Attribute Definitions in the LSM Global (Inst_lsm_global) Item

Attribute	Valid Values	Description
priv_reg_size=	*n* (blocks)	**Do not modify**. This attribute conveys the size of the LSM private region to be created on all configured disks in the target system.
lsm_hostname=	*nnnnn:hh:mm:ss* existing host name	This attribute defines the host name that LSM uses when defining the rootdg disk group. LSM host names must be unique for systems that exist on the same network. If a host name cannot be determined from an existing LSM configuration or LSM does not exist on the system, the Full Installation process generates an LSM host name in the format of a five-digit number followed by a time stamp (hour;minutes;seconds). It is recommended that when you modify the hostname= attribute in the Inst_cinstall item, you make the same change here as well so that your system maintains a unique identity on the network but has the same host name and LSM host name.

A.5 Attributes in the Inst_lsm_disks Item

Table A–5 defines the attributes in the Inst_lsm_disks item in the CDF. The Inst_lsm_disks item conveys the private region partition and disk information. An Inst_lsm_disks attribute is created for each configured disk. The Inst_lsm_disks item is created only if you chose to use LSM.

Table A–5: Attribute Definitions in the LSM Disks (Inst_lsm_disks) Item

Attribute	Valid Values	Description
name=	dsk*n*	**Do not modify**. This attribute specifies the disk associated with the LSM private region partition information. An Inst_lsm_disks attribute is created for each disk configured in the model system.
priv_reg_part=	[a-h]	**Do not modify**. This attribute conveys the partition on which the LSM private region is to be created for the disk defined by the name= attribute.

A.6 Attributes in the Inst_subsets Item

Table A–6 defines the attributes in the Inst_subsets item in the CDF. The Inst_subsets item is used to convey information to the Installation Cloning process about the base operating system software subsets that are

to be installed on the system to be cloned. One `Inst_subsets` item always will exist for the base operating system software. There can be up to two additional `Inst_subsets` items for WLS software.

Table A–6: Attribute Definitions in the Software Subsets Load (Inst_subsets) Item

Attribute	Valid Values	Description
`advflag=`	1	**Do not modify**. This attribute informs the subset load procedure, the `setld` utility, that this is a Full Installation (as opposed to an Update Installation or after the installation). It is always set to 1 by the Full Installation process.
`name=`	BASE WLS[a]	**Required. Do not modify.** This attribute specifies the name of the product associated with the subsets listed in the `ss_names` attribute. There will be a `name` attribute defined for each product installed. `BASE` represents the base operating system; `WLS` represents Worldwide Language Support software.
`ss_names=`	List of subset names	**Required**. This attribute specifies the list of base operating system and worldwide language support software subsets to be installed. Each software subset name is separated by a comma (,) and must be on one continuous line (let the line wrap). If you add software subset names to this attribute, you must consider available disk space and dependencies upon other software subsets. See the *Installation Guide* for software subset dependency information, and see the *Release Notes* for software subset size information.
`volume_name=`	DISC1 DISC2	**Required. Do not modify.** This attribute specifies the distribution on which the product defined by the `name=` attribute resides. For example, the base operating system CD-ROM has a `volume_name` value of `DISC1`; the CD-ROM containing the WLS software has a `volume name` value of `DISC2`.

[a] The literal value of this attribute is `Worldwide_Language_Support`. `WLS` is shown because it uses less space in the column.

A.7 Attributes in the Inst_cinstall Item

Table A–7 defines the attributes in the `Inst_cinstall` item in the CDF. The `Inst_cinstall` item is used to convey target system configuration

information to the Installation Cloning process. All of the attributes specified in the installation configuration item are optional. If values are not provided for these attributes, the installation process becomes interactive to request this information during the system configuration phase.

To use a single CDF to clone many systems, consider leaving the system-specific attributes such as host name and password null, but provide attributes for site-specific attributes such as kernel option, time zone, geographic location, and date and time.

Table A–7: Attribute Definitions in the Installation Configuration (Inst_cinstall) Item

Attribute	Valid Values	Description
hostname=		This attribute specifies the target system's host name to the installation process. Host names for target systems that exist on the same network must be unique. See the *Installation Guide* for guidelines on choosing a proper host name. During a RIS Installation Cloning process, this value is set automatically to the host name of the target system. For CD–ROM installations, make sure this value is set correctly or is null. A null value means that the installation process becomes interactive during the system configuration phase to request a host name.
kernel_option=	mandatory all interactive	This attribute specifies to the installation process whether the tailored kernel build should be interactive or non-interactive. If you want the Installation Cloning process to run without user intervention, do not set this attribute value to `interactive`. In an `interactive` kernel build session, a kernel components menu is presented that allows the user to select optional kernel components to build into the kernel. It is important to know that the kernel build on the target system does not automatically include optional kernel components that were on the model system unless the kernel option type is set to `interactive` and the user intentionally selects the optional kernel components.

Table A–7: Attribute Definitions in the Installation Configuration (Inst_cinstall) Item (cont.)

Attribute	Valid Values	Description
		The `mandatory` value builds a kernel with the kernel components that are mandatory for the installed software. The `all` value builds a kernel with all mandatory and all optional kernel components. Setting this attribute to `mandatory` or `all` builds the tailored kernel without any user intervention.
lang_env	See `i18n_in-tro(5)`	This attribute sets the language environment. Valid values are the same as the `LANG` environment variable. The default is United States English.
locality=	See the `/etc/zoneinfo` directory	This attribute specifies the geographic location of the target system. During a RIS Installation Cloning process, this value is set automatically to the geographic location of the RIS server. A null value means that the installation process becomes interactive during the system configuration phase to request a geographic location.
password=		**Do not modify**. This attribute specifies to the installation process the encrypted `root` password for the target system. The presence of a value here means that all cloned systems and the model system share the same `root` password. A null value means that the installation process becomes interactive during the system configuration phase to request a password. Because this entry is encrypted, either leave it alone or make it null. If you insert a value, you will set a password that you will not be able to decipher.
		After the target system is cloned, you should change this password for security reasons.
timeset=	yes no	This attribute specifies to the installation process the status of the system date and time on the target system. In the case of a RIS Full Installation or RIS Installation Cloning, this value is always set to `yes`, which means that the system date and time have been set (because it comes from the RIS server).

Table A–7: Attribute Definitions in the Installation Configuration (Inst_cinstall) Item (cont.)

Attribute	Valid Values	Description
		If this value is set to no, it means that the system date and time have not been set. The installation process becomes interactive to request the date and time.
		For CD–ROM installations, the installation process becomes interactive to request the date and time.
timezone=	See the /etc/zoneinfo directory	This attribute specifies the time zone within a specific geographic location (if applicable). During a RIS Installation Cloning process, this value is set automatically to the time zone of the RIS server. The value of timezone must be a valid time zone for the geographic location defined in the locality= attribute. For example, if locality=US, only time zones in the United States are valid. If the geographic location does not have a time zone, leave this value null. The installation process recognizes geographic locations that do not have time zones, and will not request a time zone during the configuration phase.
		If the geographic location has valid time zones, a null value means that the installation process becomes interactive during the system configuration phase to request a time zone.

B

Sample User-Supplied Files

This appendix provides samples of the following user-supplied files that can be invoked during a Full or Update Installation:

- `preinstall` file (Section B.1)
- `update_preinstall` file (Section B.2)
- `postload` file (Section B.3)
- `update_postload` file (Section B.4)
- `postreboot`file (Section B.5)

B.1 Sample preinstall File

Example B–1 contains a sample `preinstall` script, which sets site-specific attributes in an `install.cdf` file for an Installation Cloning from CD-ROM. The `preinstall` script is invoked just before the file system creation and software subset load phases of a Full Installation.

Example B–1: Sample preinstall Script

```
#!/sbin/sh
#
# This script takes a generic install.cdf file which has
# variables defined for the attributes "hostname",
# "locality", and "timezone", and substitutes real values
# for the system being installed.
#
# This script assumes the name of the generic CDF is
# install.cdf.generic.  By using this namespace, the
# installation process will not find, nor attempt to use
# this generic version.  The resulting install.cdf file is saved to the
# file /var/tmp/install.cdf so that the installation process
# will use the modified version.
#
# The relevant portion of the file install.cdf.generic is:
#
#       install:
#       _item=Inst_cinstall
#       kernel_option=mandatory
#       timeset=yes
#       password=gyq\Qy7xgNJZqXk
#       timezone=TIMEZONE
#       locality=LOCALITY
#       _action=create
#       hostname=HOSTNAME
#
# Set the real values for the variables
```

Example B–1: Sample preinstall Script (cont.)

```
HOSTNAME=aries
LOCALITY=America
TIMEZONE=New_York

#
# Substitute for the variables
#
sed -e "s/HOSTNAME/$HOSTNAME/" -e "s/LOCALITY/$LOCALITY/" \
    -e "s/TIMEZONE/$TIMEZONE/" ./install.cdf.generic > /var/tmp/install.cdf

if [ "$?" = "0" ]
then
        #
        # If the CDF was properly created, display a success
        # message, and exit with good status.
        #
        echo "/var/tmp/install.cdf successfully created"
        exit 0
else
        #
        # The CDF could not be created successfully.
        # Cause the installation process to stop.
        #
        echo " /var/tmp/install.cdf could not be created"
        exit 1
fi
```

B.2 Sample update_preinstall File

Example B–2 shows a sample update_preinstall script, which is
executed just before the initial window is displayed during the analysis
phase of an Update Installation.

Example B–2: Sample update_preinstall File

```
#!/bin/sh

#
# This is a sample script that demonstrates
# the type of operations that can be performed using
# the update_preinstall script.  Creating
# backup files of any file shipped with the operating
# system product is not strictly necessary because any
# user customizations are either merged automatically or
# saved to a .PreUPD extension.
#
BACKUP_LIST="/etc/passwd \
/etc/fstab \
/etc/group"

for FILE in $BACKUP_LIST
do
        #
        # Save each file to a .BACKUP extension if the
        # file exists (-f $FILE) and a backup file does
```

Example B–2: Sample update_preinstall File (cont.)

```
          # not already exist (! -f $FILE.BACKUP)
          #
          [ -f $FILE -a ! -f $FILE.BACKUP ] &&
          {
                  cp $FILE $FILE.BACKUP
          }
done
```

B.3 Sample postload File

Example B–3 contains a sample `postload` script, which sets multiple
host-specific attributes in a generic `config.cdf` file to perform configuration
cloning of many systems. The script is identifying each host name and
associated IP address because each system must have a unique identity
on the network. The `postload` script is invoked after the file systems
have been created and software subsets have been loaded during a Full
Installation process.

Example B–3: Sample postload Script

```
#!/usr/bin/posix/sh
#
# This is a generic postload script that expects
# to operate on a generic configuration cdf file named "config.cdf.generic"
# also found in the same directory.  This script will
# query the host name of the system being installed and dynamically
# modify the hostname and IP address in the config.cdf.generic file,
# and place the file in the /var/tmp directory as config.cdf.  The
# installation process will then find the resulting CDF file in
# /var/tmp.
#
# The relevant portion of the config.cdf.generic file is:
#
#
#         debug=false
#         devName=tu1
#         hasDynamicNetAddr=false
#         hasRCInfo=true
#         hopCountMetric=0
#         ifaceNum=1
#         maxTransUnit=1500
#         networkAddress=IPADDRESS
#         netMask=255.255.252.0
#         operational=true
#         receiveAll=false
#         receiveMulticast=false
#         speed=not_applicable
#         systemName=HOSTNAME
#         type=ETHERNET
#         useArp=true
#
# All other host name and IP address references FOR THE CLIENT BEING
# INSTALLED should be replaced by variables in the file
```

Example B–3: Sample postload Script (cont.)

```
# config.cdf.generic.
#

MOUNT=/mnt
RCMGR=/usr/sbin/rcmgr
#
# Must use the mount-relative version of ./etc/rc.config, because
# file systems are still mounted relative to /mnt.
#
RC_CONFIG=$MOUNT/etc/rc.config; export RC_CONFIG

#
# Define a table of hostname to IPaddress to network adapter mapping
#
        aries=0; IP[aries]=16.69.56.100; DEV[aries]=tu0
        taurus=1; IP[taurus]=16.69.56.111; DEV[taurus]=tu0
        gemini=2; IP[gemini]=16.69.56.222; DEV[gemini]=ln0
        cancer=3; IP[cancer]=16.69.56.105; DEV[cancer]=tu0
        leo=4; IP[leo]=16.69.56.123; DEV[leo]=tu0
        virgo=5; IP[virgo]=16.69.56.75; DEV[virgo]=tu0
        libra=6; IP[libra]=16.69.56.50; DEV[libra]=ln0
        scorpio=7; IP[scorpio]=16.69.56.55; DEV[scorpio]=tu0
        sagittarius=8; IP[sagittarius]=16.69.56.60; DEV[sagittarius]=tu0
        capricorn=9; IP[capricorn]=16.69.56.66; DEV[capricorn]=ln0
        aquarius=10; IP[aquarius]=16.69.56.70; DEV[aquarius]=tu0
        pisces=11; IP[pisces]=16.69.56.77; DEV[pisces]=tu0


GetIPAddress()
{

        eval host=\$$1

        echo ${IP[host]}
}


GetDevName()
{

        eval host=\$$1

        echo ${DEV[host]}
}

Main()
{

        #
        # Get the host name of the system being installed.  Use
        # the host name to index into a table of IP addresses, and pull
        # the correct IP address for this system.  Then, dynamically
        # update the host name and IP address in the config.cdf file.
        #
        HOSTNAME=`$RCMGR get HOSTNAME`

        IPADDRESS=`GetIPAddress $HOSTNAME`

                echo "Host is $HOSTNAME; IP is $IPADDRESS"

        #
```

Example B–3: Sample postload Script (cont.)

```
        # Now modify the version of config.cdf.generic that exists in the
        # current working directory, and copy it to the /var/tmp
        # directory so that it is found by the installation process.
        #

        sed -e "s/HOSTNAME/$HOSTNAME/g" -e "s/IPADDRESS/$IPADDRESS/g" \
                ./config.cdf.generic > /var/tmp/config.cdf

        [ -s /var/tmp/config.cdf ] &&
                echo "/var/tmp/config.cdf successfully created"

        #
        # Always exit with good status; the process is too far
        # to exit the installation at this point.
        #
        exit 0
}

 Main "$@"
```

B.4 Sample update_postload File

Example B–4 contains a sample script, which archives all *.PreUPD and
*.PreMRG files left over from an Update Installation. This operation is
meant as an example of what can be done during the update process. It is
recommended that you use the Update Installation Cleanup application
(/usr/sbin/updadmin) to archive these types of files instead of archiving
them using the update_postload script. The update_postload script
is executed after the operating system software is loaded but before the
first reboot.

Example B–4: Sample update_postload File

```
#!/bin/ksh
#
LOGDIR=/var/adm/smlogs
BACKUPDIR=/mybackups
PREMRG_FILE=$LOGDIR/upd_PreMRG_files
PREUPD_FILE=$LOGDIR/upd_custom_files

if [ ! -d $BACKUPDIR ]
then
        mkdir -p $BACKUPDIR
fi

cd /
for FILE in `cat $PREMRG_FILE $PREUPD_FILE`
do
        cp $FILE $BACKUPDIR
done
```

B.5 Sample postreboot File

Example B–5 contains a sample postreboot script which, among other
things, loads additional software subsets from a RIS server, adds entries to
the /etc/fstab file, adds users to the .rhosts file and allows remote root
logins. The postreboot script is invoked after the system reboots after a
Full Installation.

Example B–5: Sample postreboot File

```
#!/usr/bin/posix/sh
#
# This script is executed during the c-install phase of the
# full installation process.  At the time of execution, all network
# services will be available assuming that the system was configured
# using the 'sysman -clone' capability.
#

        echo "Executing postreboot script"
        #
        # Load reference page software subsets from the
        # RIS server.
        #
        SERVER=`rcmgr get INST_SERVER`

        2>&1 /usr/sbin/setld -l $SERVER: OSFMANOS540 OSFCDEMANOS540 \
        OSFMANWOS540 OSFCDEMANOP540 OSFMANWOP540 OSFDCMTEXT540

        #
        # Add an entry to the /etc/fstab file to provide an
        # NFS-mount from an exporting NFS-server
        #
        cp /etc/fstab /etc/fstab.ORIG
        echo "/mnt1@giants   /nfs-mount nfs ro,bg 0 0" >> /etc/fstab

        #
        # Make a local mount-relative directory on which to mount
        # the NFS-mounted directory
        #
        mkdir -p /nfs-mount

        #
        # Add the Engineering team members to the .rhosts file on this system.
        #

cat <> /.rhosts
aries.company.com jsmith
aries jsmith
libra.company.com mwang
libra mwang
virgo.company.com rhurley
virgo rhurley
leo.company.com jcruz
leo jcruz
taurus.company.com gwilliams
taurus gwilliams
$(hostname)
EOF

        #
```

Example B–5: Sample postreboot File (cont.)

```
        # Allow root logins from remote systems.
        #
        echo ptys >> /etc/securettys

        #
        # Use the Korn shell as the default shell for root.
        #
        sed "s@/bin/sh@/bin/ksh@" /etc/passwd > /tmp/passwd
        [[ -s /tmp/passwd ]] && mv /tmp/passwd /etc/passwd
        chmod 644 /etc/passwd

        #
        # Make changes to .profile to change default editor to vi
        #
        echo "EDITOR=vi; export EDITOR" >> /.profile

#
# Additions to sysconfigtab that help our testing.
#
cp /etc/sysconfigtab /etc/sysconfigtab.ORIG
cat <> /etc/sysconfigtab
streams:
        nstrpush=15

advfs:
        AdvfsPanicLevel=1

proc:
        maxusers=1024
        max-proc-per-user=256
        max-threads-per-user=512

vfs:
        revoke_tty_only=0

kdebug:
        kdebug_escape = iseeme
        kdebug_stop_on_panic = 0
EOF

        exit 0
```

Index

X