Installing Software with Apollo's Release and Installation Tools

Order No. 008860-A02

First Printing: June 1988
Latest Printing: October 1989

10 9 8 7 6 5 4 3 2 1

Preface

Installing Software with Apollo's Release and Installation Tools describes how to use the installation tools accompanying the Domain/OS operating system. It describes how to manage installations of the operating system and optional software products.

This manual contains general installation instructions for all Apollo software products. The release document that is shipped with the release media contains product–specific and release–specific installation information. Information contained in the release document includes the list of media required for the installation, the size of the software, and software compatibility notes.

The manual is organized as follows:

Chapter 1	Discusses things you should know and consider before attempting to install Domain/OS and optional software products.
Chapter 2	Describes how to get Domain/OS installed and running for the first time on a node in a network without any other nodes running Domain/OS. The procedures also create an Authorized Area on the first Domain/OS node. The Authorized Area can be used as a source for subsequent installations.

Chapter 3	Describes how to install Domain/OS across the network from an Authorized Area to a pre–SR10 node.
Chapter 4	Discusses the concepts underlying the Apollo installation model. Explains Authorized Areas and their structure; the installation tools and how they work together; and the use of selection, override, and configuration files.
Chapter 5	Describes how to manage an Authorized Area, constrain available software product configurations, modify an Authorized Area, and create a distributed Authorized Area.
Chapter 6	Describes how to establish product configurations for subsequent installation.
Chapter 7	Describes how to install customized product configurations on multiple nodes on a network. Included are tips on installing software unattended at large sites.
Chapter 8	Describes how to modify the protection model after software installation and gives information about protection issues related to installation.
Appendix A	Details instructions on initializing disk volumes with **invol**.
Appendix B	Details instructions on setting the date and time on node volumes, using **calendar**.
Appendix C	Describes many of the tools used in this book in a reference format. The commands are arranged in alphabetical order. They are: **cfgsa, config, distaa, fix_10_1_ri, inprot, install, install++, minst,** and **mrgri**.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Appendix D	Describes the environment that third-party solution suppliers can count upon when creating their own installation scripts.

Related Manuals

Before attempting to use this manual, you should be familiar with the basics of operating an Apollo workstation. We assume you know how to start up shells (command line interpreters), change your working directory, and manipulate files and directories, and are familiar with command line syntax and conventions. If you are not, please consult *Getting Started with Domain/OS (002348)* before proceeding.

For more information on the three user environments provided by Domain/OS, you may also want to consult the following:

- *Using Your SysV Environment (011022)*
- *Using Your BSD Environment (011020)*
- *Using Your Aegis Environment (011021)*

The file **/install/doc/apollo/os.v.***latest_software_release_number*_**manuals** lists current titles and revisions for all available Apollo manuals. For example, at Software Release 10.2 (SR10.2) refer to **/install/doc/apollo/os.v.10.2_manuals** to check that you are using the correct version of the manuals. You may also want to use this file to check that you have ordered all of the manuals that you need. (If you are using the Aegis environment, you can access the same information through the help system by typing **help manuals**.)

Refer to the *Apollo Documentation Quick Reference* (002685) and the *Domain Documentation Master Index* (011242) for a complete list of related documents.

For more information on how Domain/OS differs from pre-SR10 versions of the operating system, refer to the following documents:

- *Domain System Software Release Notes (005809)*
- *Making the Transition to SR10 Operating System Releases (011435)*
- *Making the Transition to SR10 TCP/IP (011717)*

For more information on Domain/OS commands, refer to the following documents:

- *Aegis Command Reference (002547)*
- *BSD Command Reference (005800)*
- *SysV Command Reference (005798)*

For more information on system administration topics, refer to the following documents:

- *Managing Aegis System Software (010852)*
- *Managing BSD System Software (010853)*
- *Managing SysV System Software (010851)*
- *Managing NCS Software (011895)*
  (formerly *Managing the NCS Location Broker)*
- *Network Computing Architecture (010201)*
  (formerly *Network Computing Architecture (NCA) Protocol Specifications)*
- *Network Computing System Reference Manual (010200)*
  (formerly *Network Computing System (NCS) Reference)*
- *Configuring and Managing TCP/IP (008543)*

You can order Apollo documentation by calling **1-800-225-5290**. If you are calling from outside the U.S., you can dial **(508) 256-6600** and ask for **Apollo Direct Channel**.

Problems, Questions, and Suggestions

We appreciate comments from the people who use our system. To make it easy for you to communicate with us, we provide the

Apollo Product Reporting (APR) system for comments related to hardware, software, and documentation. By using this formal channel, you make it easy for us to respond to your comments.

For more information about how to submit an APR, consult the appropriate command reference manual for your environment (Aegis, BSD, or SysV). Refer to the **mkapr** (make apollo product report) shell command description. You can view the same description online by typing:

man mkapr (in a UNIX* environment)

help mkapr (in the Aegis environment)

Alternatively, you may use the Reader's Response Form at the back of this manual to submit comments about the manual.

Documentation Conventions

Unless otherwise noted in the text, this manual uses the following conventions.

entering text	The instruction to *enter* a text string, command, or command line means to type the text then press the RETURN key. Although command lines may appear in the manual on more than one line, you always enter the command line as a single line.
literal values	Bold words or characters in formats and command descriptions represent commands or keywords that you must use literally. Pathnames are also in bold. Bold words in text indicate the first use of a new term.
user-supplied values	*Italic words or characters in formats and command descriptions represent values that you must supply.*

*UNIX is a registered trademark of AT&T in the USA and other countries.

sample user input	In examples, information that the user enters appears in color.
`output`	`Information that the system displays appears in this typeface.`
[ ]	Square brackets enclose optional items in formats and command descriptions.
{ }	Braces enclose a list from which you must choose an item in formats and command descriptions.
|	A vertical bar separates items in a list of choices.
< >	Angle brackets enclose the name of a key on the keyboard.
CTRL/	The notation CTRL/ followed by the name of a key indicates a control character sequence. Hold down <CTRL> while you press the key.
. . .	Horizontal ellipsis points indicate that you can repeat the preceding item one or more times.
. . .	Vertical ellipsis points mean that irrelevant parts of a figure or example have been omitted.
	This symbol indicates the end of a chapter.

Table of Contents

Chapter 3 Installing Domain/OS on Subsequent pre-SR10 Nodes

Chapter 4 Installation Concepts

Chapter 5 Managing Authorized Areas

Chapter 6 Configuring Products

Chapter 7 Installing Products

Chapter 8 Protecting Software

Appendix A Running invol

Appendix B Running calendar

Appendix C Command Reference

Appendix D Installing Third–Party Software

Glossary

Index

Figures

Tables

Chapter 1

Before You Install

This chapter explains what you need to know before installing Domain/OS or optional products. We have kept it short in the hope that you won't skip it.

Introducing Domain/OS

Domain/OS is the operating system that runs on Apollo workstations beginning at Software Release 10 (SR10). It provides a common set of networked operating system services via three environments (Aegis, BSD, and SysV). Each environment provides a different set of files that can be installed and used independently, but the underlying libraries and operating system services are consistent across all three environments.

Domain/OS runs on all Apollo nodes except **saul** (DN100, DN400, DN420, and DN600) machines, and Domain/OS nodes can share data across the network with nodes running earlier versions of the operating system. Although Domain/OS nodes can read or write files on nodes running earlier Apollo operating systems, the compatibility release for Domain/OS is SR9.7, and only SR9.7 nodes can read and write files on Domain/OS nodes. Therefore, if you are operating an Apollo network running pre-SR9.7 software, we urge you to update as many nodes as possible to SR9.7 before installing Domain/OS on the network. We provide tools to make the operation of mixed networks of SR9.7 and Domain/OS nodes as simple as possible.

This manual attempts to provide all the information you need to install Apollo software. You should also consult *Making the Transition to SR10 Operating System Releases* to help determine the best installation strategy for your site, which environments to install, and the best way to migrate from pre-SR10 software to Domain/OS.

The Apollo Installation Model

Installation is usually defined as the task of moving software off distribution media and into place in a new or existing file system. The Apollo installation model recognizes that, for a network-based distributed operating system, the installation task just begins with getting the software off the distribution media. It continues with propagating software through the network, and maintaining a consistent file set as the operating system is updated and new software is added to software already in place.

For that reason, the first step in an Apollo installation is the creation of an **Authorized Area** (often abbreviated as **AA**). New software is first loaded from the distribution media (usually tape) into an Authorized Area (usually on a fixed disk volume). An Authorized Area is not an operational configuration of software, but rather a source area for subsequent installations. An Authorized Area also includes all the tools and data files necessary to administer the Authorized Area and manage installations of the software loaded into it.

Once the new software is loaded off the distribution media and into an Authorized Area, it can be installed on the node containing the Authorized Area, and on other nodes on the network. See Figure 1-1.

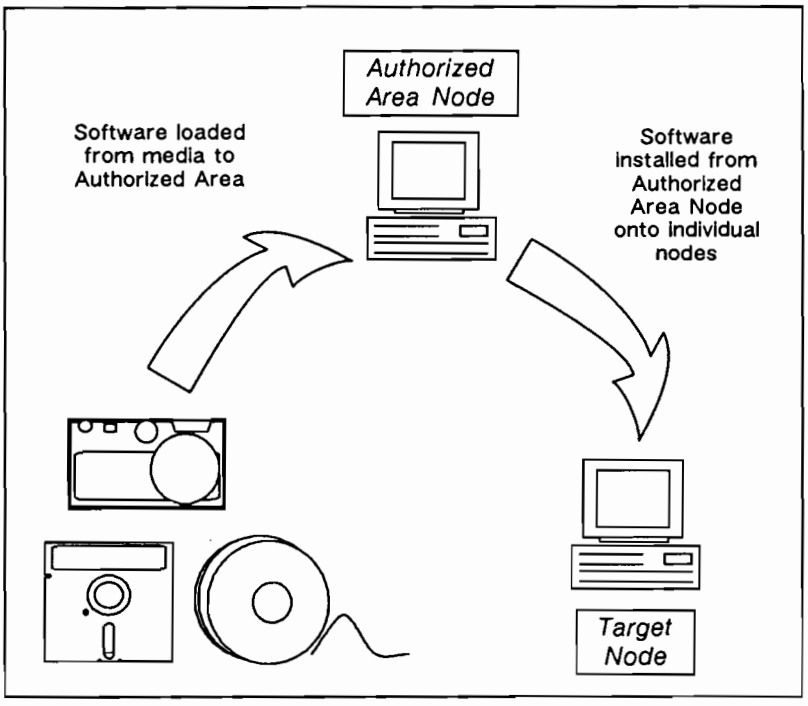


Figure 1–1. Software Installation Path

The Installation Tool Set

The Authorized Area contains a set of tools to handle various installation tasks. The set includes six *basic,* command–line utilities: **distaa, cfgsa, config, install, inprot,** and **mrgri**. Each of these tools handles a single, discrete aspect of the installation process. For example, **distaa** loads products from distribution media into an Authorized Area. The command–line interfaces of the tools allow you to combine them with more general–purpose tools, such as shells, to solve any unusual installation problems specific to your site. The basic installation tools provide the most flexible control over the installation process.

We also provide two other installation tools—**minst** and **installl++**—that are easier to use than the basic tools in some

situations, but don't provide as much flexibility. These tools are layered on top of the basic ones; that is, they transparently invoke some of the basic tools to combine discrete installation tasks into a single process. The layered tools have more interactive interfaces and require less prior knowledge to run. But they do not allow you to customize an installation as much as the basic tools do. You use **minst** to load and install the Domain/OS operating system for the first time in a pre–SR10 network.

In addition to installation tools, the Authorized Area contains tools you will need when installing Domain/OS for the first time. These tools do not install software or manage the Authorized Area; they are included in the Authorized Area for your convenience when following the installation procedures detailed in this book.

Installing Domain/OS

Installing Domain/OS on pre–SR10 Nodes

The Domain/OS disk format is different than that of previous operating system releases (pre–SR10). The format difference requires that you initialize each pre–SR10 disk with the SR10.x version of the **invol** utility before you install Domain/OS on the disk. Accordingly, we provide some *special case* procedures for handling Domain/OS installation on pre–SR10 nodes. Use the procedures in Chapter 2 to load and install Domain/OS for the first time in a network where no node is running Domain/OS. Use the procedures in Chapter 3 to install Domain/OS across the network from a Domain/OS node to a pre–SR10 node.

Installing an Updated Version of Domain/OS

To load Domain/OS from distribution media into an Authorized Area on a node already running Domain/OS (for example, to load SR10.2 on an SR10.0 node), use the appropriate procedure in the section "Loading New Products into an Authorized Area" in Chapter 5. If you want to load a configuration of Domain/OS that is too large to fit on a single disk, you can distribute the product among more than one disk when you load it. This procedure is described in the section "Distributing Products when Loading from Distribution Media" in Chapter 5.

After you load Domain/OS into an Authorized Area, you can configure and install an operational configuration of the product on one or more nodes. These procedures are described in Chapter 6 and Chapter 7.

Let us emphasize that in the Apollo installation model, Domain/OS is in principle a product that is handled just like any other optional product. The need for special procedures for installing Domain/OS on pre-SR10 nodes, as we previously mentioned, is because of the disk format change at SR10, not because of any fundamental difference between how Domain/OS and other products are structured in the installation model. Similarly, the procedures for loading and installing an updated version of Domain/OS are essentially identical to those for loading and installing other products; there are only some minor, administrative-type differences in the load procedures.

Installing Optional Products

With respect to optional products, this manual is intended primarily for installing **RAI** optional products on a network with at least one Authorized Area on a node running Domain/OS. An RAI product is a product released at or after the first version of Domain/OS (SR10). Such products have the character string RAI on the label of their distribution media.

To install an optional product, you first load the product from the distribution media into an Authorized Area using the **distaa** tool. This load procedure is described in the section "Loading New Products into an Authorized Area" in Chapter 5. If the product is too large to fit on a single disk, you can distribute the product among more than one disk when you load it. This procedure is described in the section "Distributing Products when Loading from Distribution Media" in Chapter 5.

After you load the product into an Authorized Area, you can configure and install an operational configuration of the product on one or more nodes. These procedures are described in Chapter 6 and Chapter 7.

Installing RAI Optional Products on SR9.7 Nodes

Some optional RAI products can, or must, run on an SR9.7 node. The standard installation tools, however, do not *run* on SR9.7

nodes. If your network has at least one Domain/OS node, this does not present a problem. You can install such products on an SR9.7 node using the standard installation tools and procedures. To do this, you load the product into an Authorized Area on the Domain/OS node. You then run the tools on the Domain/OS node to install the product across the network from the Authorized Area to the SR9.7 node.

If your network does not have *any* Domain/OS nodes, or for some reason you want to run the tools on a SR9.7 node in a mixed network, we provide an equivalent set of tools compiled to run on SR9.7 nodes. (See Chapter 4 for more details on this tool set.) The section "Loading New Products into an Authorized Area" in Chapter 5 explicitly addresses how to load a product from distribution media to a SR9.7 node. Once you load the product into an Authorized Area, you can use the procedures in Chapter 6 and Chapter 7 to install operational configurations of the product on SR9.7 nodes; just make sure you use the SR9.7–compatible version of the tools.

Chapter 2

Installing Domain/OS for the First Time

This chapter describes how to install the Domain/OS operating system (SR10.x) for the first time on an existing, pre-SR10 network — a network where no node is running Domain/OS. Specifically, we describe how to

- Choose and prepare the first node for the Domain/OS installation
- Initialize and boot the first node from the Domain/OS distribution media
- Load Domain/OS into an Authorized Area on the first node, and then install an operational configuration of Domain/OS on the node
- Set up a Domain/OS registry site
- Restore files backed up before the installation

After you complete the procedures in this chapter, you can use the Authorized Area on the first node to install Domain/OS onto other pre-SR10 nodes across the network. This procedure is described in Chapter 3.

Choosing the First Node

Unless your site has only one node, you have to decide which node to use for the first Domain/OS installation. The following sections describe prerequisites and other considerations for the first node.

Current Operating System Version

The first Domain/OS node can be running any version of SR9 (SR9.x), unless the first node is your network's master registry site. If the first node is the master registry site, the node must be running SR9.7. (Registry sites are discussed more fully later in this section.)

Aside from the first node, we recommend that you update as many nodes as possible to SR9.7 before installing Domain/OS, since SR9.7 is the compatibility release for Domain/OS. You can find out the current operating system version by running **/com/bldt** on the node.

Disk Capacity

You create an Authorized Area on the first Domain/OS node. Since an Authorized Area can require a lot of disk space, we suggest that you choose a node with a large-capacity disk. For instance, to load every file included in all three Domain/OS environments (Aegis, SysV, BSD) for all node types requires about 170 MB of disk space. Chapter 2 of the *Domain System Software Release Notes* tells you how much disk space is required for each installable configuration of Domain/OS. You can find out the size of a disk volume by running **/com/lvolfs**.

The first node's disk must have a capacity of at least 80 MB.

Drive Type

The first Domain/OS node must be equipped with either a cartridge tape drive or a floppy disk drive. We strongly recommend that you use a machine with a cartridge tape drive. Booting from floppy disks takes much longer than booting from cartridge tape.

Node Type

Installation is simpler if your first Domain/OS node is a workstation (equipped with a display and keyboard) rather than a Domain Server Processor (not equipped with a display and keyboard). If you don't wish to maintain an Authorized Area on a workstation, you can move the Authorized Area from the workstation to a Domain Server Processor (DSP) after the first installation is complete.

If you must use a DSP for the first Domain/OS node, note the following limitations and requirements:

- Since DSP80s and DSP90s cannot be attached directly to cartridge tape or floppy disk drives, they cannot be used as the first Domain/OS node.
- The DSP must have an internal Winchester disk, even if it is also connected to an SMD drive.
- You must attach a terminal to serial I/O line number 1 (SIO1), an RS-232 port, of the DSP to act as the system console. If there are no terminals available to act as a console, you can run a null-modem cable from SIO1 of the DSP to an RS-232 port of a workstation, and run a terminal emulator on the port at the workstation. The Apollo terminal emulator **emt** will do the job, as will any program that can emulate a dumb terminal.

  NOTE: A null-modem cable is an RS-232 cable wired from pin 7 of connector M to pin 7 of connector F, from pin 2 of connector M to pin 3 of connector F, and from pin 3 of connector M to pin 2 of connector F. In other words, a null-modem cable is a standard RS-232 cable with pins 2 and 3 "crossed" between the connectors.

Registry Sites

We do not recommend using a registry site for the first Domain/OS node. We especially recommend *against* using your master registry site for the first node. The master registry site must be running SR9.7, whether or not you use it for the first node.

If you are not sure whether a node is a registry site, run **/com/lrgy**. If the node's name appears after the string "Registry:" in the pathname *//node_name*/**registry/rgy_master**, it is the master registry site. If the node name appears in the list of "Sites of Registration Data Files," it is a registry site. If the node's name does not appear in the output of **/com/lrgy**, the node is not a registry site.

ns_helper Sites

If your network has an **ns_helper** database, we do not recommend using an **ns_helper** site for the first Domain/OS node. If you must use an **ns_helper** site for your first node, you must reinitialize the **ns_helper** database with **edns** after you install Domain/OS, or back up the **ns_helper** database files before you initialize the first node and restore them after installation. (Both of these procedures are describe later in this chapter.)

If you are not sure whether a node is an **ns_helper** site, run **/com /edns** on the node. Then issue the **lr** command at the **<edns>** prompt. If the node ID appears in the list, it is an **ns_helper** site. Enter **q** to exit **edns**. You may have to be logged in as **root** or **%.locksmith.%** to run **edns** on your system.

Preparing the First Node

Before you can install Domain/OS on a pre-SR10 node, you must initialize the node's disk or storage module, destroying all data on the respective storage device. This section describes procedures you should perform to preserve critical data prior to initialization.

Preparing a Master Registry Site

If the node you select for the first Domain/OS installation is *not* the master registry site for your network, skip this section. If for some reason you *must* use your master registry node as the first Domain/OS node, you must convert the registry to the Domain/OS format and back it up, using the following procedure:

NOTE: The programs **/install/com/cvtrgy** and **/install/com/crpasswd** are required for

this procedure. If these programs are not on your system, and it is imperative that you use the master registry site as the first Domain/OS node, contact your customer service representative.

1. If the master registry node is running Aegis only, log in as **%.locksmith.%.** to the node. Then go to Step 2.

   If the master registry node is running Domain/IX, log in as **root.** Then run **/install/com/crpasswd** to make sure all accounts are in the **passwd** file:

   ```
   /install/com/crpasswd
   ```

2. Back up the registry tree and, if the node is running Domain/IX, the **/etc/passwd** and **/etc/group** files:

   ```
   /com/cpt /registry /registry.old -sacl -pdt
   /com/cpf /etc/passwd /etc/passwd.old -sacl -pdt
   /com/cpf /etc/group /etc/group.old -sacl -pdt
   ```

3. Convert the master registry to the Domain/OS format, using the **cvtrgy** tool. Use the command line

   ```
   /install/com/cvtrgy  -from9to10
               -from source -to destination
               -owner pgo -first
   ```

   where:

   source
   : is the name of the SR9.7 registry, in the form *//first_node*/**registry/rgy_site**; for example, **//color/registry/rgy_site.**

   destination
   : is the name of the Domain/OS registry site to be created, in the form *//first_node*; for example, **//color.** For this procedure, the node name in *source* and in *destination* are identical.

   pgo
   : is the Domain/OS registry owner, in the SID form *person.group.organization*. The *person*, *group*, and

organization names and the actual registry owner account must already exist in the SR9.7 registry.

The **cvtrgy** tool creates a read-only Domain/OS registry and adds a few new accounts to your SR9.7 registry to make it compatible with the Domain/OS registry. Ignore messages warning you about creation of the new accounts.

4. Back up your old SR9.7 registry, your new Domain/OS registry, and the **/etc/passwd**, **/etc/group**, and **/etc/org** files. The SR9.7 registry database is in **/registry**, while the Domain/OS registry database is in **/sys/registry**; archive *both* directory trees.

   Use this command line to back up the registry files:

   ```
   /com/wbak -dev device -l -nhi -f end -fid registry
             /etc/passwd /etc/group /etc/org
             /registry  /sys/registry
   ```

   where *device* is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

For more information about Domain/OS registries and the **cvtrgy** program, see *Making the Transition to SR10 Operating System Releases* and your managing system software manuals.

Preparing an ns_helper Site

If the first node is an **ns_helper** site and you wish to maintain it as an **ns_helper** site after installing Domain/OS, you must either back up the **ns_helper** databases before initialization and restore them after installation, or reinitialize the databases with **edns** after starting **ns_helper** again. The latter is is not that difficult because **edns** can reconstruct the databases with little intervention on your part.

If you choose to back up the databases, change your working directory to **/sys/node_data** and issue the following command:

```
/com/wbak -dev device -l -nhi -f end
          -fid ns_helper 'ns_helper.?*'
```

where *device* is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

Backing Up User Trees

You should back up all user directories and files on the first node to some removable medium (cartridge tape, magnetic tape, or floppy disk). Use this command line to perform the backup:

```
/com/wbak -dev device -l -nhi -f end
          -fid user_trees  pathname1...pathnameN
```

where:

device
is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

pathname1...pathnameN
are user home directories.

Backing Up Other Files

In addition to users' home directory trees, you may want to back up the following types of files for later use or reference.

Startup Files

You may want to back up startup files in the directories **'node_data** and **/sys/dm**. If other nodes use the first node for booting diskless, there may be additional startup files in **/sys/node_data.***node_ID* directories, where *node_ID* is the node ID of a node that boots diskless from the first node.

The Domain/OS installation provides a new set of startup file templates in the appropriate directories. However, the information from your old startup files may help you get the node up and running the way you like it more quickly.

Site-Specific Files

If you added local tools, databases, or other files to this node, you may want to back them up so you can reinstall them later. For example, you may want to back up any local extensions to the

standard UNIX utilities in **/usr/local**, or printer configuration files in **/sys/print**, if the node is attached to a printer.

Customized Font Files

You may have customized font files in **/sys/dm/fonts** that you want to save.

TCP/IP Administrative Files

If the node is running TCP/IP, and especially if the node is a TCP/IP administrative site, you may want to back up the TCP/IP administrative files.

If the node is running Domain/IX TCP/IP (most likely if the node is running Domain/IX), you should back up

- **/etc/hosts**
- **/etc/networks**
- **/etc/gateways**
- **/etc/hosts.equiv**

If the node is running Domain TCP/IP (most likely if the node is not running Domain/IX), you should back up

- **/sys/tcp/hostmap/local.txt**
- **/sys/tcp/hostmap/hosts.txt**

After you install Domain/OS, consult *Making the Transition to SR10 TCP/IP* for the details of configuring TCP/IP in a Domain/OS network.

UNIX System Configuration Files

If the node is running Domain/IX, you may want to archive some UNIX system configuration files for later reference. The Domain/OS installation provides a new set of templates for these files. However, the information from your old files may help you get

the node up and running the way you like it more quickly. The files you may want to back up include

- **/etc/rc**
- **/usr/lib/crontab**
- Configuration files for **uucp**, if the node is a **uucp** administrative site

Initializing the First Domain/OS Node

This section describes how to initialize and boot the first node in preparation for installing the Domain/OS system software. You initialize the node using the Domain/OS version of **invol** and then boot from the Domain/OS distribution media. Before you initialize the node, make sure you perform the backup procedures described in the previous section.

While you can load the Domain/OS system software from cartridge tapes, floppy disks, or magnetic tapes, you must boot the first node from a cartridge tape or floppy disks.

Before Booting the First Node

Before you initialize and boot the first Domain/OS node,

- You must know the capacity of the first node's disk or storage module. You can find out the size of a disk volume by running **/com/lvolfs**.
- If the node is a DSP, the DSP must be running.
- If the first node is a DSP equipped with a storage module, you should know how many units the storage module contains, and which ones you want to initialize with **invol**. You must run **invol** on at least the module's Winchester boot volume. We recommend you run the Domain/OS version of **invol** as soon as possible on all units of the storage module.

Booting from Cartridge Tape

Use the following procedure to boot the node from cartridge tape. (If you want to boot from floppy disks, skip to the next section.)

1. Make sure the target node is in NORMAL (versus SERVICE) mode. This mode is usually controlled by a toggle switch on the back panel of the CPU. See your node's operating guide for more information.

2. Shut down the node by entering the **shut** command at the Display Manager (DM) command line:

   ```
   Command: shut
   ```

   Wait for the message

   ```
   SHUTDOWN SUCCESSFUL
   ```

   and for the Mnemonic Debugger (MD) prompt to appear. The prompt depends on the node firmware, but it will end in a >.

3. Enter a reset command, followed by a carriage return at the next prompt. The reset command for Series 10000 workstations is **RE W**. For all other workstation types, the command is simply **RE**. For example,

   ```
   >RE
   ><RETURN>
   MD8  Rev. 4.2, 1987/04/29.12:48:02
   >
   ```

4. Make sure the Domain/OS boot tape, labeled **crtg_std_sfw_boot_1**, is **write-disabled**.

   To check this, hold the tape cartridge so that the word "SAFE" embossed in the plastic casing is visible in the upper left corner (see Figure 2–1). Immediately to the left of the word "SAFE" is a plastic screw head, half of which is a semicircle and half of which is a triangle. If the apex of the triangle points to the word "SAFE," the tape is write-disabled. If the tape isn't write-disabled, use a tool to turn the screw head to the proper position.

After you ensure the tape is write–disabled, insert the cartridge in the tape drive.

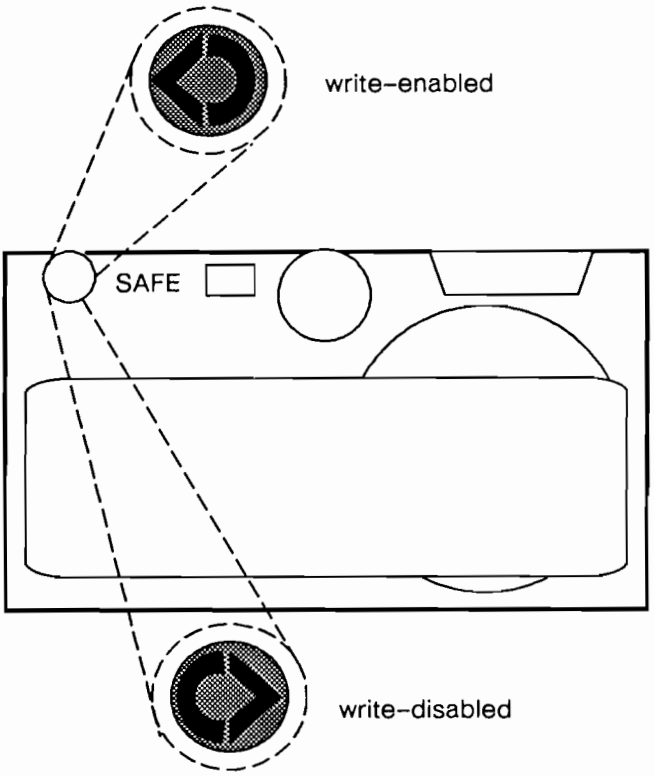


Figure 2–1. Write–Enabled/Write–Disabled Cartridge Tapes

5. Use the appropriate **DI** command to select the cartridge tape drive as the device from which to load boot software.

   On **sau2**, **sau3**, **sau4**, **sau5**, **sau6**, **sau7**, **sau8**, and **sau10** nodes, the correct command is

   >DI C

   On **sau9** nodes, the command is

```
>DI T
```

You can use the Mnemonic Debugger command **LD** (List Sau) to find out the sau number of the node.

6. Start the **calendar** program:

```
>EX CALENDAR
```

Respond to the series of prompts. See Appendix B for a detailed description of the prompts. Running **calendar** at this point ensures that **invol** will create valid Unique Identifiers (UIDs) for the objects it creates on the disk.

7. When you finish running **calendar**, reset the node again:

```
>RE  [ or RE W for Series 10000 workstations ]
><RETURN>
MD8  Rev. 4.2, 1987/04/29.12:48:02
>
```

8. Select the cartridge tape drive again:

```
>DI C                          (sau2-8, sau10)

>DI T                          (sau9)
```

9. Start the **invol** program:

```
>EX INVOL
```

If you are unfamiliar with the **invol** program, refer to Appendix A, which provides a detailed description of the **invol** procedure. When you finish with Appendix A, go to the next step.

On the **invol** menu, select option 1, Initialize a Virgin Physical Volume. Respond to the subsequent prompts. When asked

```
Anything more to do?
```

enter **y**. Then select option 8, Create or Modify an OS Paging File. Respond to the subsequent prompts. Unless you have special paging size requirements, accept the default paging size. When asked,

```
Anything more to do?
```

enter **no**.

10. Reset the node:

```
>RE  [ or RE W for Series 10000 workstations ]
><RETURN>
MD8  Rev. 4.2, 1987/04/29.12:48:02
>
```

11. Select the cartridge tape drive again:

```
>DI C                          (sau2-8, sau10)

>DI T                          (sau9)
```

12. Run the **calendar** program again, this time to set the time for objects subsequently installed on the disk:

```
>EX CALENDAR
```

See Appendix B for a detailed description of the **calendar** prompts.

13. When you finish running **calendar**, load the minimum bootable system software onto the node, with the command

```
>EX DOMAIN_OS
```

14. A confirmation prompt appears:

```
*** This program will replace system software
on your disk. Do you wish to proceed? (Y/N):
```

Answer **yes**.

The boot program then loads a subset of the operating system, enough to run the DM or Server Process Manager (SPM) process. The Domain/OS installation tools are also loaded at this point.

15. When the boot program enters phase II, indicated by the) prompt, enter

) GO

If you are at a node with a display, the DM is started, and the login prompt appears. If you are at a DSP, the SPM is started.

Skip the following section and continue with the section "Loading Domain/OS from the Distribution Media."

Booting from Floppy Disks

Use the following procedure to boot your node from floppy disks.

NOTE: To boot from floppy disk, the node must have an internal Winchester disk.

1. Make sure the target node is in SERVICE mode. This mode is usually controlled by a toggle switch on the back panel of the CPU. See your node's operating guide for more information.

2. If the node has a display, shut down the node by entering the **shut** command at the Display Manager (DM) command line:

   ```
   Command: shut
   ```

   If you are at a DSP, enter the **shut** command in the **emt** window or at the terminal input line:

   ```
   shut
   ```

   Wait for the message

   ```
   SHUTDOWN SUCCESSFUL
   ```

   and for the Mnemonic Debugger (MD) prompt to appear. The prompt depends on the node firmware, but it will end in a >.

3. Enter a reset command, followed by a carriage return at the next prompt:

```
>RE
><RETURN>
MD8  Rev. 4.2, 1987/04/29.12:48:02
>
```

4. Put the floppy disk labeled **FLP*x*_PREP*n*** into the disk drive, where

   x is **8**, if you are using 8-inch floppies, and **5** if you are using 5-1/4-inch floppies.

   n is the number corresponding to the **/sau*n*** directory required to run the node you are using.

   Table 2-1 shows the correspondence between the machine types for which a floppy boot is possible and the **/sau*n*** directories. Note that the **/sau2** and **/sau4** machines use 8-inch floppies to boot with and magnetic tape for the actual software installation.

Table 2-1. The /sau Directories by Machine Type for Floppy Boot

/sau	Machine Type
/sau2	Series 300, 320, 330
/sau4	Series 160, 460, 660
/sau7	Series 3500, 4000, 4500
/sau8	Series 3000

For example, if you are using a DN3000 with a 5-1/4-inch floppy, choose the floppy labeled **FLP5_PREP8**.

The *x* and *n* variables in all subsequent references of the form **FLP*x*, PREP*n*, *and* BOOT*n*** have the same meaning as those described here.

5. Select the floppy disk drive as the device from which to load software with the command

```
>DI F
```

6. Start the **calendar** program:

```
>EX CALENDAR
```

Respond to the series of prompts. See Appendix B for a detailed description of the prompts. Running **calendar** at this point ensures that **invol** will create valid Unique Identifiers (UIDs) for the objects it creates on the disk.

7. When you finish running **calendar**, reset the node and select the floppy drive again:

```
>RE
><RETURN>
MD8  Rev. 4.2, 1987/04/29.12:48:02
>DI F
```

8. Start the **invol** program:

```
>EX INVOL
```

If you are unfamiliar with the **invol** program, refer to Appendix A, which provides a detailed description of the **invol** procedure. When you finish with Appendix A, go to the next step.

On the **invol** menu, select option 1, Initialize a Virgin Physical Volume. Respond to the subsequent prompts. When asked

```
Anything more to do?
```

enter **y**. Then select option 8, Create or Modify an OS Paging File. Respond to the subsequent prompts. Unless you have special paging size requirements, accept the default paging size. When asked

```
Anything more to do?
```

enter **no**.

9. Reset the node, select the floppy drive, and run **calendar** again, this time to set the time for objects subsequently installed on the disk:

```
>RE
><RETURN>
MD8  Rev. 4.2, 1987/04/29.12:48:02
>DI F
>EX CALENDAR
```

See Appendix B for a detailed description of the **calendar** program.

10. Remove the floppy from the disk drive. Then insert the floppy disk labeled **FLP*x*_BOOT*n*** into the drive, after you make sure it is **write-enabled**.

Check the floppy disk as follows:

5 1/4" floppy:
: Write-enabled if the notch on the floppy's edge is visible. If the notch is not visible, remove the adhesive tab covering it. (See Figure 2–2.)

8" floppy:
: Write-enabled if notch on the floppy's edge is not visible. If the notch is visible, cover it with an adhesive tab. (See Figure 2–3.)

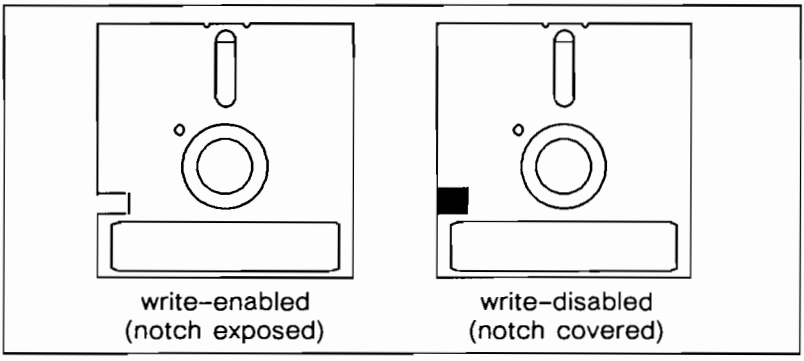


Figure 2–2. Write-Enabled/Write-Disabled 5–1/4–Inch Floppy Disks

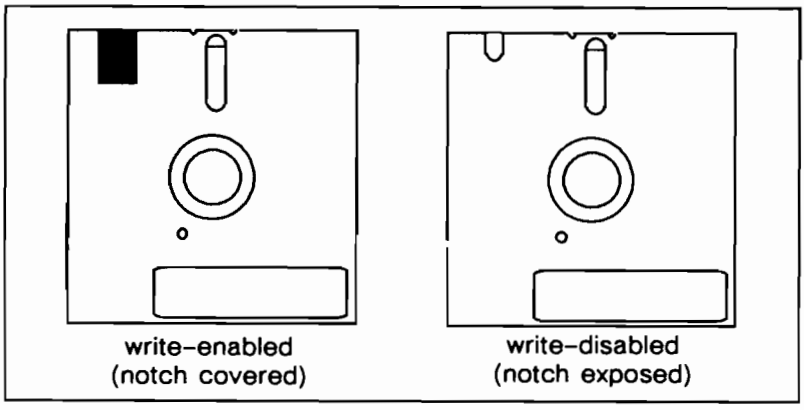


Figure 2-3. Write-Enabled/Write-Disabled 8-Inch Floppy Disks

11. Reset the node, select the floppy drive, and boot from the floppy:

```
>RE
><RETURN>
MD8  Rev. 4.2, 1987/04/29.12:48:02
>DI F
>EX DOMAIN_OS
```

After a series of messages, the phase II boot shell comes up, indicated by the) prompt.

12. Transfer the contents of the floppy to the hard disk with the command

```
)CF /FLP/INSTALL/LOAD_BOOTn
```

13. Shut down the node, reset it, and boot from hard disk:

```
)SHUT
BEGINNING SHUTDOWN SEQUENCE
          .
          .
          .

SHUTDOWN SUCCESSFUL

>RE
><RETURN>

>EX DOMAIN_OS
```

14. Remove the floppy disk from the drive.

15. Insert the remaining Domain/OS floppy disks into the floppy drive and load their contents to disk. Insert the disks in the order shown below, or according to the prompts of the program, if they are different. If additional floppies are required, you are prompted for them by name.

    After you insert each floppy, enter the command

)CF /INSTALL/LOAD_*floppy_name*

    where *floppy_name* is the part of the floppy disk name appended to **FLP***x*_. For example, to load **FLP***x*_**BASIC_1**, enter

)CF /INSTALL/LOAD_BASIC_1

    NOTE: Before you insert each floppy, make sure it is **write-disabled**:

    5 1/4" floppy:
    Write-disabled if the notch on the floppy's edge is not visible. If the notch is visible, cover it with an adhesive tab. (See Figure 2–2.)

    8" floppy:
    Write-disabled if the notch on the floppy's edge is visible. If the notch is covered, remove the adhesive tab. (See Figure 2–3.)

    Floppy Insertion Order:

FLP*x*_BASIC_1

FLP*x*_BASIC_2

FLP*x*_BASIC_*n* (Additional basic floppies as needed)

FLP*x*_TOOLS_1

FLP*x*_TOOLS_2

FLP*x*_TOOLS_*n* (Additional tool floppies as needed)

FLP*x*_LIBRARY_1

FLP*x*_LIBRARY_2

FLP*x*_LIBRARY_*n* (Additional library floppies as needed)

16. If your node is a DN460 or DN660, insert the floppy labeled **FLP*x*_UCODE4**. Then load its contents (**/sau4** microcode) with the command

)CF /INSTALL/LOAD_UCODE4

17. Remove the floppy disk from the drive.

18. Switch the node from SERVICE to NORMAL mode.

19. At this point, enough of the operating system has been loaded to run the DM or the SPM. To complete the booting process, enter

)GO

    If you are at a node with a display, the DM is started, and the login prompt appears. If you are at a DSP, the SPM is started.

Continue with the next section.

Loading Domain/OS from the Distribution Media

After you initialize and boot the first node from the Domain/OS distribution media, you must create an Authorized Area on the node, load Domain/OS from the distribution media to the

Authorized Area, and finally install an operational configuration of Domain/OS from the Authorized Area to the node.

We provide an interactive tool, named **minst**, that enables you to easily accomplish these tasks. The **minst** program automatically starts whenever you log in as **user** after booting from the distribution media. The program is identified by the prompt `MINST>`. **minst** leads you through the entire process step by step, displaying detailed, explicit instructions along the way; for the most part, running **minst** is self-explanatory.

The procedure we present here is an overview, tailored specifically to the task of installing Domain/OS for the first time in a pre-SR10 network. We suggest you read through this procedure before you begin **minst**.

To use **minst** to install Domain/OS on the first node,

1. Log in as **user** on the first node. (Just press <RETURN> at the password prompt.) Since you just booted the node from the Domain/OS distribution media, the **minst** program starts automatically.

2. Enter **c** (continue) in response to the first prompt:

   ```
   Do you wish to continue with MINST, or quit
   now?
   : [ continue  quit  help ]
   MINST> c
   ```

   If for some reason you must quit rather than continue **minst**, rerun **minst** when you are ready by entering the command **/install/tools/minst** at a shell prompt.

3. When **minst** asks whether you want to run the program in novice mode or expert mode, select novice mode unless you are an experienced user of the installation tools and need to customize the installation in a way that novice mode does not allow. The remainder of this procedure assumes **minst** is running in novice mode.

NOTE: In novice mode, **minst** installs Domain/OS using a *closed* (versus *open*) protection model. See Chapter 8 for more information about open and closed environments. Also, in novice mode, **minst** installs Domain/OS on the first node, using hard links to the Authorized Area, rather than local copies. This saves disk space on the node.

4. When asked to enter the pathname of the Authorized Area, accept the default. The default is the node entry directory on the current node (*//first_node*).

5. When asked to enter the pathname of the target — where you want to install the operational configuration of Domain/OS — accept the default. The default is the current node (*//first_node*).

6. When asked if you intend to install Domain/OS on the target, enter **y**.

7. Specify the distribution media type (cartridge tape, magnetic tape, or floppy), insert the first volume of Domain/OS into the drive, and press <RETURN>, as instructed by **minst**.

   minst then loads the first file on the tape to the Authorized Area (*//first_node*). The first file includes the installation tools and their associated help files, release notes for Domain/OS, and several template files. The template files enable you to load and install a configuration of Domain/OS tailored to the specific needs of your site. All these files are loaded to the directory *//first_node*/**install**.

8. **minst** pauses to allow you to read the online release notes for Domain/OS. Read them or the hard-copy version distributed with the operating system. Pay particular attention to Chapter 2 of the release notes, as it provides information you need in the next step.

9. **minst** requests you to choose which Domain/OS configuration you want to load into the Authorized Area and install. You choose from a list of templates. Each template represents a self-consistent configuration of Domain/OS

software tailored to fit on a mass storage device of a given capacity.

Each template name includes one or more of the following environment names:

- **aegis** (Apollo's proprietary Aegis environment)
- **bsd4.3** (Berkeley Software Distribution 4.3)
- **sys5.3** (AT&T System V Release 3)

The environment names identify which environments **minst** will load into the Authorized Area and install on the node.

Three Aegis configurations are available for installation. These are identified by their relative sizes: small, medium, and large. Likewise, two BSD and SysV configurations are available: medium and large. Each template name includes a size designation—**small**, **medium**, or **large**—indicating which size of the associated environment or environments will be loaded and installed. For example, the template "aegis bsd.4.3 large" installs a large aegis and large BSD configuration.

The actual size of each Domain/OS configuration is provided in Chapter 2 of the Domain/OS release notes. Chapter 2 also lists the components included/excluded in each configuration. We recommend that you load the largest configuration of your chosen environment(s) that the first node's disk will accommodate.

10. After you select a template, **minst** begins to load the selected Domain/OS configuration into the Authorized Area. At the end of each tape or floppy, **minst** prompts you to insert the next tape or floppy into the drive.

11. When all files are loaded, **minst** asks

```
Do you wish to select Domain/OS version xx.x
or quit from minst?
:  [ select  quit ]
```

Enter **s** (select) in response. This causes **minst** to install an operational configuration of Domain/OS from the Authorized Area to the node.

If for some reason you wish to quit **minst** (enter **q**) rather than select Domain/OS for installation, rerun **minst** when you are ready by entering the command **/install/tools/minst**. Then repeat the entire **minst** procedure. You do not need to initialize the disk again before you rerun **minst**.

12. When **minst** completes execution, check the transcript for errors and warnings (prefixed by ERROR and WARNING). The "Troubleshooting" section in Chapter 7 describes some of the errors that may occur during the installation phase of **minst**. If you find errors, rerun **minst**.

Shutting Down and Rebooting the Node

Once you successfully install Domain/OS, shut down the node and boot it from its own disk, using the following procedure.

1. Enter the following command:

   Command: shut (at the DM prompt on a workstation)

   shut (in an **emt** or **crp** window on a DSP)

2. If you loaded software on a DN460 or DN660, reload the microcode with these commands:

   ```
   >gb
   %ua
   ```

   NOTE: Though the prompt should return to > after the **ua** command, sometimes the % prompt persists. If this occurs, simply continue with the next step.

3. Reset and boot the node, using the following commands:

   ```
   >RE  [ or RE W on Series 10000 workstations ]
   ><RETURN>
   MD8  Rev. 4.2, 1987/04/29.12:48:02
   >EX DOMAIN_OS
   ```

The first node is now running Domain/OS. Now set up a Domain/OS registry site, using the procedures in the next section.

Setting Up a Domain/OS Registry Site

Because Domain/OS requires a different registry from that used by pre-SR10 nodes, you must establish a Domain/OS registry on the first Domain/OS node in your network.

Creating a Domain/OS Registry Database

You can create a Domain/OS registry in one of the following ways:

- You can create a new registry database if your site has never had a registry before. Do this if you are setting up an Apollo workstation or network for the first time.
- You can convert a pre-SR10 registry database if your network has a registry and the master registry is currently located on an SR9.7 node.
- You can restore a previously converted Domain/OS registry database. If you converted your SR9.7 registry database to the Domain/OS format before installing Domain/OS on the first node and archived the database somewhere, you can restore it and use it for your Domain/OS registry database.

After determining which method is appropriate for your network, use *one* of the following procedures.

Creating a New Registry

Use the procedure below if you are creating a new registry, at a site which has never had a registry before.

1. Log in as **user** to the Authorized Area node; that is, the node you just installed to. If the node is a DSP, issue the **shell** command to the SPM via the terminal connected to the serial I/O port to log in.

NOTE: The SPM does not issue a prompt, but it will accept some commands from the console. Type **help** to the SPM for a list of commands it supports.

2. The **rgy_create** (create new registry) program resides in the directory *AA*/**install/tools**, where *AA* is the pathname of the Authorized Area you created on the node you just installed to. Invoke the **rgy_create** tool by typing

   AA/install/tools/rgy_create

   The **rgy_create** tool runs as **root**. Therefore, we recommend that, after using it to create a master registry, you either remove it from the network or change the permissions on the program so that only trusted users are able to run it.

Now that you have created a registry database on your Domain/OS node, start your Domain/OS registry processes by using the directions in the section "Starting the Registry Processes" later in this chapter.

Converting an SR9.7 Registry Database

Use the procedure below if you are converting an SR9.7 registry to the Domain/OS format.

1. Log in as **root** or **%.locksmith.%** to the SR9.7 node that is your network's master registry site. If your network is running Domain/IX, the master registry site must have Domain/IX installed. (You can **crp** on if you want.) The **cvtrgy** (convert registry) tool only runs on an SR9.7 node.

   NOTE: If your site does not have Domain/IX installed, skip Steps 2 and 3 and go on to Step 4.

2. Copy the SR9.7-compatible version of **crpasswd** from the first Domain/OS node to the SR9.7 node you are working at, using the following command line:

/com/cpf *AA*/install/tools_sr9/crpasswd /etc/crpasswd

where:

AA
is the pathname of the Domain/OS Authorized Area on the node you just installed.

3. Run the new SR9.7-compatible version of **crpasswd** to make sure all accounts are in the **passwd** file:

/etc/crpasswd

4. Back up the registry tree, as well as the **/etc/passwd** and **/etc/group** files, if they exist, by typing

```
cpt /registry   /registry.old
cpf /etc/passwd  /etc/passwd.old
cpf /etc/group  /etc/group.old
```

5. The **cvtrgy** program resides in the **install/tools_sr9** directory of the Authorized Area on the Domain/OS node you just installed. Invoke **cvtrgy** like this as a single command line:

AA/install/tools_sr9/**cvtrgy** **-from9to10**
-from *source* **-to** *dest*
-owner *pgo* **-first**

where:

AA
is the pathname of the Domain/OS Authorized Area on the node you just installed.

source
is the name of the SR9.7 master registry site, in the form *//node_name*/**registry/rgy_site**; for example, **//color/registry/rgy_site**.

dest
is the name of the first Domain/OS node in the form *//first_node*; for example, **//sound**. This should be the name of the Domain/OS node on which you have just installed software.

pgo
is the Domain/OS registry owner, in the SID form *p.g.o*, where the project, group, and organization names and the actual registry owner account already exist in the SR9.7 registry. For example, the owner of a registry might be **root.admin.software**.

The **cvtrgy** tool creates a read-only Domain/OS registry on *//first_node* and adds a few new accounts to the SR9.7 registry to make it compatible with the Domain/OS registry. You may ignore messages warning you about creating new accounts.

6. If you are running Domain/IX, run **crpasswd** again on this node. Use the same version that you copied from the Domain/OS media. To do this, enter

   /etc/crpasswd

Now that you have created a registry database on your Domain/OS node, start your Domain/OS registry processes by using the directions in "Starting the Registry Processes." For more information on Domain/OS registries and on the **cvtrgy** program, see *Making the Transition to SR10 Operating System Releases* and the managing system software manuals.

Restoring a Previously Converted Domain/OS Registry

If you already created and archived a Domain/OS registry, now restore it from the media on which it was archived. Insert the media in the first Domain/OS node's drive (or mount the magnetic tape). Then, assuming you backed up the registry according to the instructions given earlier in this chapter, enter the following command line on the first node:

```
rbak -dev device -fid registry /sys/registry
                 -as /sys/registry -sacl -du -pdt
```

where *device* is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

Now that you have created a registry database on your Domain/OS node, start your Domain/OS registry processes by using the directions in the next section.

Starting the Registry Processes

After you have a Domain/OS registry database on your Domain/OS node, start up the registry processes by using the following procedure:

1. Log in to the Domain/OS node as **user**. At this point, there should be a read-only copy of the registry database on this node in the **/sys/registry** directory.

2. Start a local location broker process on the node with the following command:

   ```
   /etc/server -p /etc/llbd &
   ```

3. Create the following files in **/etc/daemons** on the Domain/OS node. The existence of these files causes the registry server and the local location broker to start up when the node is rebooted. Use one of the following commands:

   (UNIX environments)
   ```
   touch /etc/daemons/llbd /etc/daemons/rgyd
   ```

   (Aegis environment)
   ```
   crf /etc/daemons/llbd /etc/daemons/rgyd
   ```

4. If your network was previously running the Network Computing System (NCS), and there is already a copy of the global location broker daemon (**glbd**) running somewhere on your network, skip the next two steps and continue with Step 6. Otherwise, start a global location broker process on the node with the following command:

   ```
   /etc/server -p /etc/glbd -first -create &
   ```

5. Create a file named **glbd** in the **/etc/daemons** directory on the Domain/OS node to insure that **glbd** is started whenever the node boots. Use one of the following commands:

   (UNIX environments)
   touch /etc/daemons/glbd

   or

   (Aegis environment)
   crf /etc/daemons/glbd

6. Reboot the node so the registry processes are started by **init** and run with **root** privileges.

If you created a new registry, we recommend that you change the initial passwords for the Apollo-supplied account entries. All Apollo-supplied accounts, including **root**, ship with the password **-apollo-** by default. Then change the ownership of the registry database, originally **%.%.%**. If you used **rgy_create** to create your registry database, you may also want to add site-specific person, project, and organization information. At this point, log in as **root** and use the **edrgy** program to modify the registry database. For more information on **edrgy**, see the online manual pages and the command reference manual for your environment.

You have now completed the basic setup procedure of your Domain/OS registry. To complete the first node installation procedure, perform the node administration tasks described next.

Node Administration Tasks

Once you have set up the Domain/OS registry, perform the following node administration tasks. You may have to be logged in as **root** to perform some of the tasks.

Restoring ns_helper Files

If the first Domain/OS node was not an **ns_helper** site before you updated it to Domain/OS, skip this section. If this node was an **ns_helper** site and you want it to remain an **ns_helper** site now

that it's been updated to Domain/OS, perform the following procedures.

If you backed up the SR9.7 **ns_helper** database files, restore the files to their new Domain/OS locations by executing the following commands:

```
rbak -dev device -fid ns_helper -l -sacl
          /sys/node_data/ns_helper.db
          -as /sys/ns/helper_data/ns_helper.db

rbak -dev device -fid ns_helper -l -sacl
          /sys/node_data/ns_helper.prop
          -as /sys/ns/helper_data/ns_helper.prop

rbak -dev device -fid ns_helper -l -sacl
          /sys/node_data/ns_helper.err_log
          -as /sys/node_data/system_logs/ns_helper.err_log
```

Start **ns_helper** again with the following command line:

```
/etc/server /sys/ns/ns_helper &
```

If you did not back up the **ns_helper** databases from the SR9.7 node, reinitialize them by invoking **edns** with the node ID of the first Domain/OS node and issuing the **init** command to **edns**.

> **NOTE:** The **edns** program requires an internet address for the node ID. If you do not know the numeric form of the node ID for the first node, run **bldt** on the node to find it out.

For example,

```
$ /etc/edns 1907A
The default ns_helper is 1907A
<edns> init
5 nodes responded to lcnode request
5 entries added to directory
0 names already existed 0 errors
<edns> q
$
```

Find the following lines in **/etc/rc.user** and uncomment them so that **ns_helper** starts up whenever the node is booted:

```
#
#  Naming server:
#
# if [ -f /sys/ns/ns_helper ]; then
#         (echo " ns_helper\c" >/dev/console)
#         /sys/ns/ns_helper &
# fi
#
```

After uncommenting the appropriate lines, the same portion of the file should look like this:

```
#
#  Naming server:
#
if [ -f /sys/ns/ns_helper ]; then
         (echo " ns_helper\c" >/dev/console)
         /sys/ns/ns_helper &
fi
#
```

Restoring User Trees

If you used **wbak** to back up user trees before running **invol** on the disk, use the Domain/OS version of **rbak** to restore them to their original positions in the file system. To do this, enter the command

/etc/rbak -dev *device* **-sacl -pdt -l -fid user_trees -all**

where *device* is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

Restoring Other Files

We do not recommend that you restore any other files at this time. The functions of most operating system and DM startup files have been taken over by the following files:

- **/etc/rc**
- **/etc/rc.local**
- **/etc/rc.user**

If you wish to have your old files on the node for reference, do not restore them to their original locations; restore them instead to some other directory. Do not restore any files to the following directories:

- **/install**
- /sys
- **/com**
- **/lib**
- **/sau2, /sau3, /sau4, /sau*n*, ... /sau10**

You have now completed installing Domain/OS on the first node. To install Domain/OS on other pre-SR10 nodes, use the procedure in Chapter 3. To install optional products, first load the optional products from distribution media into the Authorized Area, using the procedures in Chapter 5. Then install an operational configuration of the products on other nodes as desired, using the information in Chapter 6 and Chapter 7.

Recommendations for Small Networks

If your network has exceptionally limited disk space (for example, if the largest disk you have is only 50 megabytes), you can recover

disk space on the node containing your Authorized Area using the following methods.

- After you install Domain/OS on all nodes that you want to update, delete from your Authorized Area node any **/sau** directories not needed for ordinary use by the Authorized Area node or by nodes that regularly boot diskless from it.

- Delete unneeded directories from the **install** directory. For example, you may not need the **tools_sr9** or **sr9.7_compat** directories. See Chapter 4 for more information about the contents of an Authorized Area.

- When you complete all software installation, including optional products, for the time being, remove the **install** subdirectory from the Authorized Area altogether. If installed products on the Authorized Area node are hard-linked to the Authorized Area, rather than local copies, you can still remove the **install** directory. But the space saved is minimal.

  Once you remove the **install** directory, you cannot install a product again unless you first create a new Authorized Area and load the product, and any other products on which it is dependent, from media. You can reload a product into the Authorized Area using the **distaa** tool, as described in Chapter 5.

- Run **/etc/salacl** to merge duplicate Access Control Lists (ACLs) into a single copy and delete unused ACLs.

Chapter 3

Installing Domain/OS on Subsequent pre–SR10 Nodes

This chapter explains how to install Domain/OS (SR10.*x*) from an Authorized Area to a pre–SR10 node. We assume you have already installed Domain/OS from distribution media and created an Authorized Area on at least one node in the network. (This procedure is described in Chapter 2.) You can now use that node as a source for installing Domain/OS across the network on other pre–SR10 nodes.

We refer to the disk or storage module on which you plan to install Domain/OS as the **target**. This chapter describes how to install Domain/OS on two types of targets:

- A disk or storage module connected to a workstation (also called a **disked node**)
- A disk or storage module connected to a Domain Server Processor (DSP)

In both cases, you must initialize the pre–SR10 target (using a Domain/OS version of the **invol** utility) before you install Domain/OS on it. For this reason, the installation procedure is significantly different from simply updating an SR10.*x* node to a later version of Domain/OS.

Installing on a Target Connected to a Workstation

This section explains how to install Domain/OS from an Authorized Area to a pre–SR10 target (disk or storage module) connected to a

workstation. We refer to the target disk and workstation collectively as the **target node**.

During these procedures, you sit at the target node and enter commands at its keyboard. You boot the target node from another node, the **partner node**, that is already running Domain/OS (see Figure 3–1). You use the partner's utility programs (**invol**, **mtvol**, and **dmtvol**) to initialize the target, mount the target on the partner, and unmount the target after the installation is complete.

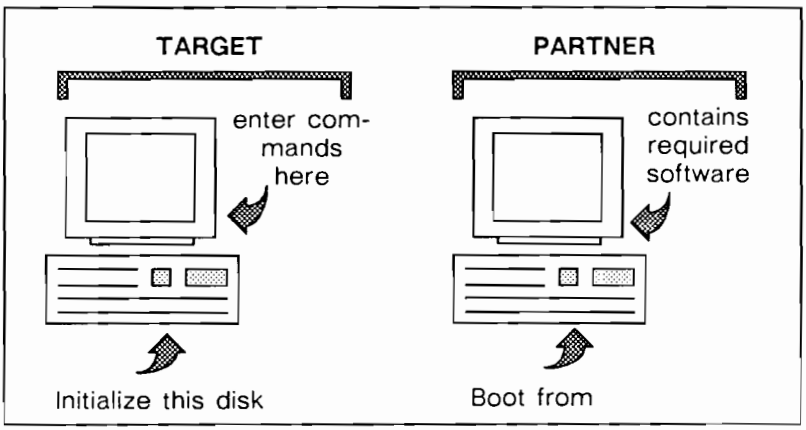


Figure 3–1. Installing Domain/OS on a Workstation

Partner Node Requirements

Select a partner node that satisfies the following criteria:

- The partner must be running Domain/OS (SR10.0 or later). Use the **bldt** command to check this:

  bldt *//partner_node_name*

- The partner must be running **netman**. Use one of the following command lines to check this:

  pst -n *//partner_node_name* (Aegis environment)

ps -e -n *//partner_node_name* (SysV environment)

ps -ax -n *//partner_node_name* (BSD environment)

- The partner must contain the /sau directory for the target node's machine type. Table 3-1 lists the /sau directory required for each machine type.

Table 3-1. The /sau Directories by Machine Type

/sau #	Machine Type
/sau2	DN300, DN320, DN330
/sau3	DSP80, DSP80A, DSP90
/sau4	DSP160, DN460, DN660
/sau5	DN550, DN560, DN570, DN580, DN590
/sau6	DSP500-T, DN570-T, DN580-T, DN590-T
/sau7	DN35*xx*, DN4000, DN45*xx*, DSP35*xx*, DSP4000, DSP45*xx*
/sau8	DN3000, DSP3000
/sau9	DN2500
/sau10	DN10000

Information Checklist

Before you begin the installation procedure, have the following information in hand:

- The target node's name and node ID. The **bldt** command returns this information in the form

  **** Node *xxxxx.node_id* **** "*//node_name*"

- The node ID of the partner node. (Use **bldt** *//partner_node_name* to find this out.)

- The pathname of the Authorized Area containing the version of Domain/OS that you want to install. The Authorized Area need not reside on the partner node.

- The number of logical volumes on the disk, if the target is a Winchester disk. (Use option 5 of the **invol** program at the Mnemonic Debugger level to find this information.) If

the disk contains more than one logical volume, identify which volume(s) you want to initialize.

Most users initialize the entire disk as one logical volume. However, you can use just one logical volume as the Domain/OS boot volume and preserve other existing volumes, using option 2 or 3 of the **invol** program.

- The number of units the storage module contains, if the target is a storage module, and which unit you want to initialize as the boot volume. Although you must initialize only one unit as the boot volume, we recommend you run the Domain/OS version of **invol** a soon as possible on all units.

Summary of Installation Procedure

This section provides a summarized account of the procedure for initializing the target and installing Domain/OS on it. A detailed procedure is provided in the next section. Experienced users may find this summary sufficient for performing the installation.

1. Back up user directories and files on the target to another node or tape.

2. Shut down the target node and boot it from the partner node. Log in.

3. Invoke **invol** from a shell to initialize the target volume (**invol** option 1) and reset the size of the OS paging file (**invol** option 8).

4. Mount the target volume on the partner.

5. Create a configuration file for the target:

   /AA/install/tools/config -s *AA* -c *configuration_file*

6. Install Domain/OS on the target:

   /AA/install/tools/install -vxp -c *configuration_file* -s *AA* *target*

7. Final steps:

   — Unmount the target volume.

- Shut down the target node.
- Reset the target's calendar.
- Reboot the target node.
- Recatalog the target node.
- Restore user directories and files.

Installation Procedure

The following subsections comprise a single, detailed procedure for initializing the target and installing Domain/OS on it.

Back Up User Files

Before you install Domain/OS on a pre-SR10 node, you must initialize the target volume, destroying all data on it. Therefore, if the target volume contains user data files that you want to save, you should back up (and later restore) these files.

You can back up the user files by copying them to another node in the network, and then move them back to the original node after the installation is complete. Or you can back up and restore from tape, using the **wbak** and **rbak** commands.

> NOTE: We recommend that you do *not* back up system directories such as /sys and /com, since you will install SR10.x versions of these. If you choose to back up system files, do so separately from other data, and do *not* restore these files to a node running Domain/OS. They will not run and may corrupt the system.

For a discussion of which directories and files you may want to back up, see the section "Preparing the First Node" in Chapter 2.

Boot Diskless From the Partner Node

1. Shut down the target node by entering the **shut** command at the Display Manager (DM) command line:

Command: shut

Wait for the message

SHUTDOWN SUCCESSFUL

and for the Mnemonic Debugger (MD) prompt to appear. The prompt depends on the node firmware, but it will end in a >.

2. Enter a reset command, followed by a carriage return at the next prompt. The reset command for Series 10000 workstations is **RE W**. For all other workstation types, the command is simply **RE**. For example,

```
>RE
><RETURN>
MD8  Rev. 4.2, 1987/04/29.12:48:02
>
```

3. Select the partner node as the node from which to boot, using the command

   >DI N 0*xxxx*

   where *xxxx* is the hexadecimal node ID of the partner node.

4. Boot the target node from the partner node, with the command

   >EX DOMAIN_OS

5. After a series of messages, the DM login prompt appears. Log in as **user** or yourself.

Initialize the Target Volume

Run the **invol** program to initialize the disk (or boot volume) on the the target node, and reset the size of the OS paging file:

1. Invoke **invol** from a shell by entering

   /etc/invol (UNIX environment)

/com/invol (Aegis environment)

At this point, if you are unfamiliar with the **invol** program, you may wish to refer to Appendix A, which provides a detailed description of the individual **invol** prompts. When you finish with Appendix A, go to the next section, "Mount the Target Volume." If you are an experienced user, continue with the next step.

2. On the **invol** menu, select option 1, initialize a virgin physical volume, to initialize the entire disk. Or, if the target disk contains more than one logical volume, and you want to initialize and install Domain/OS on only one of the volumes, select option 3, re-initialize an existing logical volume.

3. Respond to the subsequent **invol** prompts until asked

   ```
   Anything more to do?
   ```

   Enter **y**.

4. Select option 8, Create or Modify an OS Paging File. Respond to the series of prompts. When asked,

   ```
   Anything more to do?
   ```

   enter **n**.

Mount the Target Volume

Mount the volume on which you plan to install Domain/OS. The command you use depends on which environments are installed on the partner node. The SysV and BSD mount command (**/etc/mount**) requires that you are logged in as **root**. The Aegis mount command (**/com/mtvol**) requires no special permissions.

In an Aegis environment, use the command line

/com/mtvol {w|w*x*:*y*|s*n*} [*logical_volume_number*] */pathname*

where:

{w|w*x*:*y*|s*n*}
You enter w if the target is a Winchester disk; w*x*:*y* if the

target is a Winchester disk on a Series 2500 workstation, where *x* is the controller number and *y* is the unit number; or *sn* if the target is a storage module, where *n* is a unit number from 0 to 3 inclusive. The default unit number is 0.

logical_volume_number
is the number of the logical volume you want to mount for installation. The default is 1; you can omit this option if you initialized the target as a single logical volume.

pathname
is a unique pathname, usually the node name of the target node.

In a SysV environment, use the commands

mkdir */pathname*

(Winchester disk)
/etc/mount /dev/dsk/W*N*d0s1 */pathname*

(Storage module)
/etc/mount /dev/dsk/S*N*d0s1 */pathname*

where:

pathname
is a unique pathname, usually the node name of the target node.

N is the disk or storage module unit number.

In a BSD environment, use the commands

mkdir */pathname*

(Winchester disk)
/etc/mount /dev/wn*N*a */pathname*

(Storage module)
/etc/mount /dev/sm*N*a */pathname*

where:

pathname
is a unique pathname, usually the node name of the target node.

N is the disk or storage module unit number.

Create a Configuration File for the Target

Use the **config** installation tool to create a configuration file for the target node. The configuration file primarily specifies which products in the Authorized Area are to be installed on the target (in this case, which version of Domain/OS), which components of each product are to be installed, and how each component is to be installed (copied or linked). You supply the configuration file as input to the **install** tool (as described in the next section), which actually installs the selected product(s) on the target. For a more comprehensive description of **config**, refer to Chapter 6 and Appendix C.

> **NOTE:** You can combine the steps in this subsection and the following one by using the **install++** tool. See Chapter 6 and Appendix C for more information.

1. Invoke **config** with this command line:

   AA/install/tools/config -s *AA* -c *configuration_file*

   where:

   AA
   is the pathname of the Authorized Area.

   configuration_file
   is the pathname of the configuration file you want to create or modify. You can place the file anywhere on the network (including a pre-SR10 node). If you don't plan to use the configuration file again, we suggest you place the file on the target node and delete it later.

   Upon entry, **config** displays a list of the products available for installation. (You can redisplay this list with the **config** command **s a**.)

2. Select the version of Domain/OS that you want to install, by entering

   ```
   CONFIG> se os 10.revision_level
   ```

   where:

   revision_level
   is the desired revision level (0, 1, or 2) of SR10. For Series 10000 workstations, a **p** is appended to the revision level; for example, **10.2.p**. The selected version must be in the list of available products displayed upon entry to **config**.

3. Begin the configuration process for the selected version by entering

   ```
   CONFIG> config os 10.x
   ```

   config asks you to respond to a series of questions. Generally, for each component of Domain/OS, **config** asks whether you want to install the component (if it is optional), install the component by copying it to the target node, or install the component by creating a link from the target to another node on which the component is already installed.

4. When you finish responding to the questions, enter

   ```
   CONFIG> exit
   ```

Install Domain/OS on the Target

1. Install Domain/OS on the target node, using the command line

   ```
   AA/install/tools/install -pvx -c configuration_file
                -s AA target
   ```

   where:

   AA
   is the pathname of the Authorized Area.

   configuration_file
   is the pathname of the configuration file you created or modified with **config**.

target
is the pathname of the directory that you mounted the target volume as. For example, if you mounted the target volume as /my_node, enter **/my_node** as the target.

For a more comprehensive discussion of **install**, see Chapter 7 and Appendix C.

2. **install** displays a series of messages as it installs the software. When **install** completes execution, check the transcript pad for warnings and errors (prefaced with the labels `WARNING` and `ERROR`). Chapter 7 provides descriptions of some common error messages.

   Upon completion **install** instructs you to shut down and reboot the target node. Ignore this message; follow the instructions in the next subsection.

Final Steps

After you successfully install the Domain/OS software,

1. Unmount the target volume from the partner node, using the appropriate command. Use the same command line variables that you used when you mounted the volume (as described in the section "Mount the Volume.")

   In an Aegis environment:

   /com/**dmtvol** {w|w*x*:*y*|s*n*} [*log_vol_number*] */pathname*

   In a SysV environment:

   (Winchester disk)
   /etc/**umount** /dev/dsk/W*N*d0s1

   (Storage module)
   /etc/**umount** /dev/dsk/S*N*d0s1

   In a BSD environment:

   (Winchester disk)
   /etc/**umount** /dev/wn*N*a

   (Storage module)
   /etc/**umount** /dev/sm*N*a

2. Enter the **shut** command in the DM command line:

   ```
   Command: shut
   ```

3. When the MD prompt appears, enter a reset command, followed by a carriage return at the next prompt. The reset command for Series 10000 workstations is **RE W**. For all other workstation types, the command is simply **RE**. For example,

   ```
   >RE
   ><RETURN>
   MD8  Rev. 4.2, 1987/04/29.12:48:02
   >
   ```

4. If the target is a storage module connected to a DN460 or DN660, reload the microcode with these commands:

   ```
   >gb
   %ua
   ```

   NOTE: Though the prompt should return to > after the **ua** command executes, sometimes the % prompt persists. If it does, simply continue with the next step.

5. Execute the **calendar** program:

   ```
   >EX CALENDAR
   ```

   Answer the series of prompts displayed by **calendar**. For a description of the individual prompts, see Appendix B.

6. Reset the node again (as described in Step 3):

   ```
   >RE [W]
   ><RETURN>
   MD8  Rev. 4.2, 1987/04/29.12:48:02
   >
   ```

7. Boot the target node, using the command

   ```
   >EX DOMAIN_OS
   ```

   When the login prompt appears, log in as **user** or yourself.

8. Recatalog the target node, using the command

(Aegis environment)
/com/ctnode *node_name node_id* **-root -r**

(UNIX environment)
/etc/ctnode *node_name node_id* **-root -r**

where:

node_name
is the name under which you want to catalog the target node, that is, the name you want for the node entry directory. Usually you use the target node's original node name. Do *not* proceed the name with any slashes.

node_id
is the system-supplied ID of the target node (obtainable with the **bldt** command).

9. Restore any user directories and files that you backed up prior to initializing the disk. If you **crp** onto an SR9.7 node to restore files using its tape drive, use this version of **rbak**:

/sr9.7_compatibility/sr9.7_executables/com/rbak

The installation of Domain/OS is now complete.

Installing on a Storage Module Connected to a DSP

This section describes how to install Domain/OS (SR10.*x*) from an Authorized Area to a pre-SR10 storage module connected to a Domain Server Processor (DSP).

During these procedures, you log in to the DSP and boot diskless from a partner node that contains the appropriate **/sau** directory. Your work node is the partner node; enter all commands from its keyboard. The partner provides the **invol**, **calendar**, **mtvol** or **mount**, and **dmtvol** or **umount** programs to initialize the target.

Backing Up User Files

If the target volume contains data that you want to retain, do a complete backup of all user directories, as well as user-modified

startup files in /sys/**node_data**, /sys/**node_data**.*node_ID*, and /sys/**dev**, before you begin these procedures. You will destroy all data on the storage module during the **invol**.

> **NOTE:** If you are installing Domain/OS to the target for the first time, we do not recommend backing up your system directories. If you do wish to back up standard system files, do so separately from other data. Do not restore pre-SR10 standard system files to a node running Domain/OS, as they will not run and may corrupt the system.

Checklist

Before you begin to initialize, make sure of the following:

- You know how many units the storage module contains, and which ones you want to **invol**.
- You know the DSP's node ID and the name under which you plan to catalog it in the network.
- You know the location of an Authorized Area on this network.
- The work node contains the appropriate /**sau** directory for the DSP. See Table 3–1 earlier in this chapter to determine which /**sau** directory you will need.
- The work node is set up as a partner to the DSP (see Figure 3–2), and is running **netman**. Refer to your managing system software manuals for more information about setting up a partner.
- You know your time zone.
- You know whether **ns_helper** (the naming server) is running on your network.
- The DSP is running.

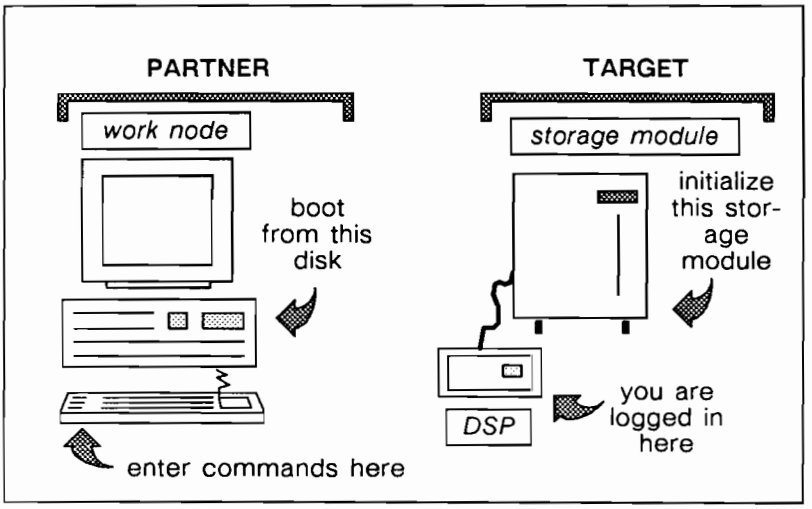


Figure 3–2. Installing Domain/OS on a Storage Module Connected to a DSP

Installation Procedure

Follow these steps to initialize the storage module.

1. Log in to the work node.
2. Power down the storage module.
3. Reboot the DSP as a diskless device, using your DSP operating guide as a reference. You should have already set up the work node as a permanent partner of the DSP, as directed in the "Checklist" section.
4. From the work node, create a process on the DSP with the following command line:

   crp -on *node_id*

   where *node_id* is the node ID of the DSP.
5. When the **crp** banner appears, log in.
6. Power on the storage module.

7. Invoke the **calendar** program in the **crp** window so that the **invol** program will apply the correct Unique Identifiers (UIDs) to the objects it creates:

   /com/calendar

   When asked for the disk type, enter **s**. Appendix B explains how to respond to the **calendar** prompts.

8. Invoke the **invol** program in the **crp** window to initialize the storage module:

   /com/invol (Aegis environment)

   /etc/invol (UNIX environment)

   After the storage module has been initialized, the program asks you

   ```
   Anything more to do?
   ```

   Answer **yes** and select Option **8** to set the paging size. See Appendix A for step-by-step instructions on the **invol** prompts, including the option that sets the paging size.

   NOTE: The recommended size of the paging file for SR10 and SR10.1 is 590 blocks. For SR10.2, the recommended size is 2048 blocks for Series 10000 workstations and 640 blocks for all other workstation types.

9. Run **calendar** again to set the correct time on the disk:

   /com/calendar

   When asked for the disk type, answer **s**. See Appendix B for an explanation of the **calendar** prompts.

10. Uncatalog the current target node's name (that is, the name of the DSP) with the following command. Do not preface the target node's name with slashes (//). Include the **-root** option if **ns_helper** is running on your network, to remove the old name from the naming server database. Use the appropriate command line:

    /etc/uctnode *target* [-root] (UNIX environment)

/com/uctnode *target* [-root] (Aegis environment)

11. Mount the initialized storage module on the file system of the partner node. The method you use to do this depends on the default environment running on the partner node. To mount a storage module in a UNIX environment, you may have to become superuser. In an Aegis environment, the command for mounting a storage module requires no special permissions. Enter the **mtvol** or **mount** command in the **crp** window of the work node.

    A. In a SysV environment:

    Enter the following command to create a directory on the partner's file system on which to mount the target module. We recommend that you give it the name **/target**.

    mkdir */target*

    Then enter the **mount** command on the target node. In the following command, *N* is a unit number from 0 to 3 inclusive.

    /etc/mount /dev/dsk/S*N*0d0s1 */target*

    B. In a BSD environment:

    Enter the following command to create a directory on the partner's file system on which to mount the target module. We recommend that you give it the name **/target**.

    mkdir */target*

    Then enter the **mount** command on the target node. In the following command, *N* is a unit number from 0 to 3 inclusive.

    /etc/mount /dev/sm*N*0a */target*

    C. In an Aegis environment:

    Run the **mtvol** command on the target node. When you mount the storage module, specify the name under which you plan to catalog it. This may or may not be the same name (*target*) that you used in Step 10. Mount the storage module on the DSP as follows, where *n* is a unit number from 0 to 3 inclusive.

mtvol *sn //partner_node/target*

12. Install the standard operating system software, as well as your choices from the available optional software products, on the target node by entering the following command line:

    AA/install/tools/install++ -s *AA* *//partner_node/target*

    where:

    AA is the pathname of an Authorized Area containing Domain/OS.

    //partner_node/target
    is the pathname at which the target volume was mounted on the partner node.

    Answer the prompts of the **install++** program as described in Chapter 6. After you have finished the installation, do not shut down the target. Return to these procedures and continue with Step 13.

13. Before you shut down the target node, unmount the initialized storage module from the file system of the partner node. Again, the method you use to do this depends on the default environment running on the partner node. Unmounting a storage module in a UNIX environment may also require that you become superuser. In an Aegis environment, the command for unmounting a storage module requires no special permissions. The following commands are entered in the **crp** window on the work node.

    A. In a SysV environment:

    Use the **umount** command. In the following command, the *N* following the S is a unit number from 0 to 3 inclusive.

    /etc/umount /dev/dsk/S*N*d0s1

    B. In a BSD environment:

    Use the **umount** command. In the following command, the *N* following the sm is a unit number from 0 to 3 inclusive.

/etc/umount /dev/sm*N*a

C. In an Aegis environment:

Use the **dmtvol** command to unmount the target node that you mounted in Step 11. In the following command, *n* is a module unit number from 0 to 3 inclusive.

dmtvol sn *//partner_node/target*

14. Log out from the DSP by typing

shutspm

15. Reboot the DSP system. You can perform this operation using an **emt** window over a serial line or a dumb terminal connected to a serial line.

```
>RE W
><RETURN>
MD3X    Rev. 6.0, 1986/03/05, 16:52:12
>EX DOMAIN_OS
```

Alternately, you can press the RESET button on the DSP.

16. Log back on to the work node as yourself, and create a process on the DSP by entering

crp -on *node_id*

For example, for a DSP with node ID 1123, enter

crp -on 1123

17. Uncatalog the current name of the target by entering

/etc/uctnode node_*node_id*

For example, for a DSP with node ID 1123, enter

/etc/uctnode node_1123

18. Catalog the target under the name you want it to have. (This name may be the same as the name it had before it was initialized.) Then update the naming server database, if there is one on your network, and the naming trees of the other nodes in the network.

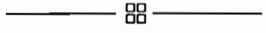
If **ns_helper** is running in your network, enter

/etc/ctnode *target target_node_id* -root -r
/etc/ctnode -update

If **ns_helper** is *not* running in your network, enter

/etc/ctnode *target target_node_id* -r
/etc/ctnode *target target_node_id* -on //?*

The initialization and software installation procedure is now complete.



Chapter 4

Installation Concepts

This chapter introduces the full set of Apollo installation tools and provides some conceptual background to help you use them more effectively. The concepts presented in this chapter are helpful in solving any installation problems unique to your site. Before reading this chapter, please read Chapter 1 if you have not already done so.

Authorized Area Structure

An Authorized Area is a directory that acts as a repository, or distribution center, for software products to be installed across the network on other nodes. The directory contains the tools and data structures necessary to install products, as well as the products themselves. The directory can be at the node entry level or a subdirectory. An Authorized Area is similar to the "source areas" used in pre-SR10 Apollo installations.

Products stored in an Authorized Area are not stored in the same way as they are on nodes using installed versions of the products; products in an Authorized Area are not operational configurations. For example, an Authorized Area located at **//node1/source** will ordinarily not contain the subdirectories **com**, **sys**, **etc**, or **bsd4.3**. The only directory guaranteed to exist in an Authorized Area is its **install** subdirectory; **//node1/source/install**, in our example. However, if the Authorized Area is the node entry directory of a node

that has Domain/OS installed, the Authorized Area will include the usual root directories (**lib**, **usr**, **saun**, and so on) in addition to the **install** directory.

The minimum contents of an Authorized Area are shown in Figure 4–1.

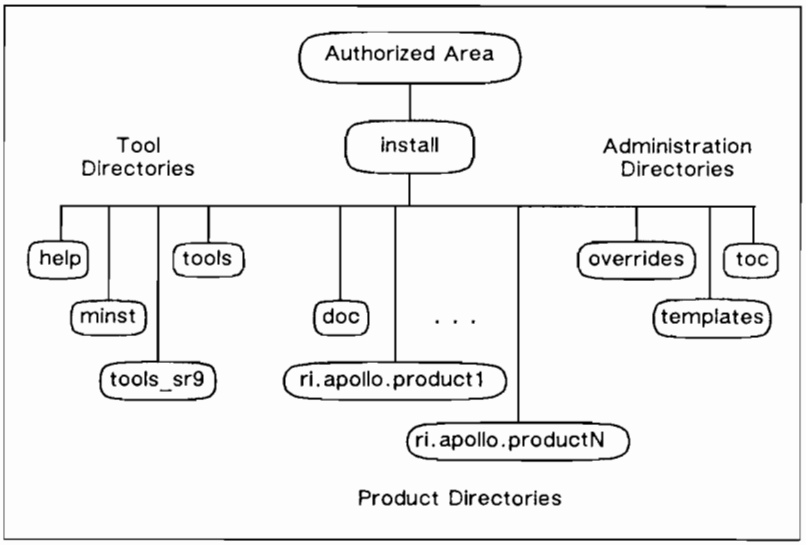


Figure 4–1. The install Directory in an Authorized Area

Most of the objects in the **install** directory of an Authorized Area can be organized into three major groups: the tool directories, the product directories, and the administration directories.

Tool Directories

The tool directories contain the executable versions of the installation tools and the data files to support them. They do not contain any product-specific objects.

The install/help Directory

A **help** directory contains online documentation for the installation tools.

The install/minst Directory

The **minst** directory contains data files needed by **minst**. We strongly recommend that you do not alter the **install/minst** directory.

The install/tools Directory

The **tools** directory contains the executable installation tools. The **install/tools** directory contains the following six basic tools:

distaa Loads software from distribution media into an Authorized Area. In general, **distaa** modifies the contents of an Authorized Area according to a selection file created by **cfgsa**. You can also use it to distribute an Authorized Area over more than one node in a network.

mrgri Merges two product release directories from an Authorized Area into a single product release directory containing a release index for the new merged product. With **mrgi**, you can merge a patch with the product it patches. You can also merge a version of a product that runs on Series 10000 workstations with the version that runs on other Apollo workstations.

cfgsa Is an interactive program for creating selection and override files. It allows a system administrator to restrict in advance the range of choices presented during the configuration phase of subsequent installations.

config Uses the release indexes for products in an Authorized Area to prompt the user about customizations allowed by the release index and any active override file for the product. It then creates a configuration file based on the answers supplied by the user. The configuration file created by **config** is used by the **install** tool to control the configuration of the installed software.

install Installs an operational configuration of software from an Authorized Area to a target node. The configuration of the installed software is determined by the release index supplied with each

product, as constrained by an override file for the product created by **cfgsa** (if one exists in the **install/overrides** directory), and as modified by a configuration file created by **config**.

inprot Sets up a protection model for the directories and files installed on a target node, according to a protection template file.

Two other tools—**install++** and **minst**—layered on top of the six basic tools.

install++ Is an interactive program that calls **config** and **install** to install software from an Authorized Area to a target node.

minst Is an interactive program we provide to make it easy for you to install Domain/OS for the first time in a pre-SR10 network. **minst** invokes **rbak** to load the installation tools and other administrative files from the distribution media into an Authorized Area, invokes **distaa** to load the Domain/OS system software into the Authorized Area, and invokes **install++** to install Domain/OS from the Authorized Area onto the first node.

The install/tools_sr9 Directory

The **tools_sr9** directory contains the same set of tools contained in the **tools** directory compiled to run on SR9.7 systems. If an optional product is compatible with SR9.7 and has a release index (as indicated by the character string "RAI" on the label of the distribution media), you can run the **tools_sr9** tools on an SR9.7 node to install the product on an SR9.7 node. Do not, however, use the **tools_sr9** tools to install an SR10-compatible product from an SR9.7 node to an SR10 node. The resulting Access Control Lists (ACLs) on the installed product will not be correct.

If you want to install an SR9.7-compatible RAI product from an Authorized Area on an SR10 node to an SR9.7 node, you can use the regular tools (the tools in the **install/tools** directory).

Product Directories

The product directories contain the source areas for the installation of any products loaded in the Authorized Area. The contents of these directories depend on the products available.

The install/doc Directory

The **doc** directory contains release documentation. In the **doc** directory, release documentation is grouped into subdirectories by company. Apollo release documents are shipped in the directory **install/doc/apollo**. There may be documentation from other companies stored in other subdirectories of the **install/doc** directory.

The Release Directories

There can be one or more release directories in the **install** directory, one for each software product available for installation. If more than one version of a product is available, there is a release directory for each version. The name of a release directory is in the form

ri.*company.product_name***.v.***version*

where:

company
: is a unique name identifying the company that developed the product. Apollo products use **apollo** for *company*.

product_name
: is the name of the software product recognized by the installation tools. A Pascal compiler might use **pas** for *product*; a FORTRAN compiler might use **ftn**.

version
: is the version number of the product, for example **10.2**, **10.2.p**, or **2.1**.

For example, the release directory for Version 2.1 of Apollo's Pascal compiler would be **ri.apollo.pas.v.2.1**.

Each release index directory contains a **release index** for that product, as well as source files for all the objects that make up the product. The name of a product's release index file is identical to the name of that product's release index directory.

For example, the release index for Version 3.2 of an Apollo Pascal compiler might be named **ri.apollo.pas.v.3.2**, and its release directory might look somewhat like the tree in Figure 4–2.

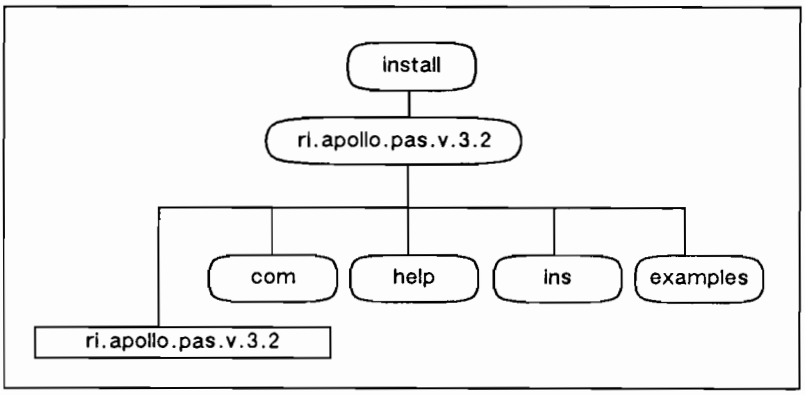


Figure 4–2. Structure of a Release Directory

The release index for a product describes all the allowed configurations for the product. It, in a sense, defines the product for the installation tools. The release indexes are the source of the questions displayed by **cfgsa** and **config**, as well as the blueprints used by **install** to build an operational file system on the target node.

Administration Directories

The administration directories contain the data files needed to administer an Authorized Area and control installations from it.

The install/templates Directory

The **templates** directory contains templates for data files used in various installation tasks. The template files provided by Apollo are stored in the **install/templates/apollo** directory. The **install/templates** directory may contain other subdirectories containing template files supplied by other companies. Figure 4–3 shows the structure of the **install/templates** directory.

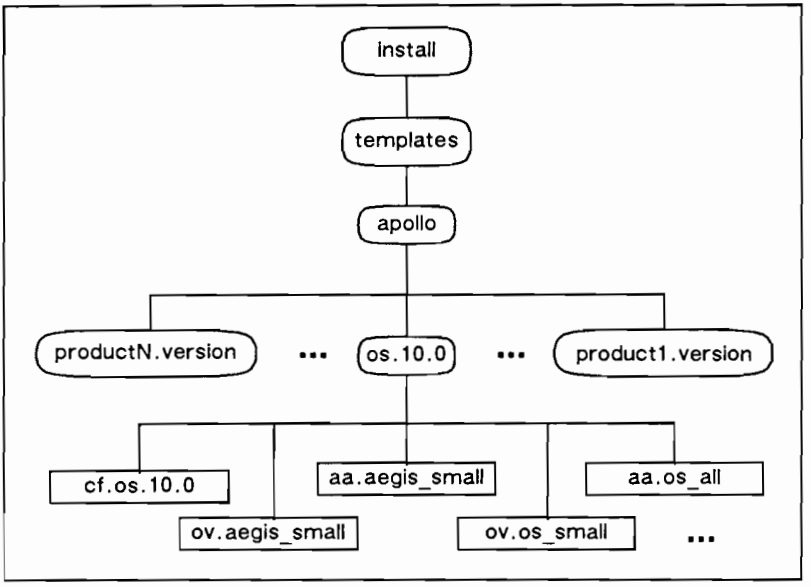


Figure 4–3. Structure of install/templates Directory

There are currently three types of installation data files stored in the **install/templates** directory: **selection files, override files,** and **configuration files**.

Selection files have the prefix **aa.** . You can use them with **distaa** to selectively load components of a product, rather than the entire product, from distribution media to an Authorized Area. In addition to the default selection files that ship with a product, additional selection files can be created with the **cfgsa** tool.

Override files in the **install/templates** directory have the prefix **ov.** and constrain the configuration choices allowed by the release index for a product. In addition to the default override files shipped with a product in the **install/templates** directory, customized override files can be created with the **cfgsa** tool. An override file is said to be active if it is in the **install/overrides** directory and has the same name as the release index for the product for which it was created. The installation tools ignore any override files not in the **install/overrides** directory.

Configuration files have the prefix **cf.** and are used by **install** to determine which of the possible configurations defined by the

release index and allowed by the active override file (if any) is the actual configuration to be installed on the target node. A configuration file defines just one configuration of operational software, whereas a release index defines all of the possible configurations.

The install/overrides Directory

The **overrides** directory contains override files. Override files allow system administrators to restrict the configuration of software installed from the Authorized Area. Only one override file per product can be in the **overrides** directory and thereby in effect at any given time. The name of a product's override file in the **overrides** directory must be identical to the name of that product's release index file and product release directory (**ri.***company.product_name*.**v.***version*).

The **overrides** directory can also contain the **excludes.list** and **protections.list** files. The **overrides/excludes.list** file contains pathnames (relative to the node entry directory, /) of files that are not to be installed from the Authorized Area. The **overrides/protections.list** file is a file you can create to set object permissions on nodes when software is installed. If a **protections.list** file exists, the **install** program calls the **inprot** (install protections) tool to set permissions on the installed objects whenever an installation completes. The "Template File Format" section in Chapter 8 describes the required format for a **protections.list** file.

The install/toc Directory

The **toc** directory contains files that index the contents of media loaded into the Authorized Area. It is a record of the source of all products loaded into the Authorized Area. File names in the **install/toc** directory are of the following form:

toc.*company.volume_id.media_type*

where:

company
: is a unique name indicating the company that developed the product. Apollo products use **apollo** for *company*.

volume_id
: is a string containing the volume identifier for a piece of

media. For media distributed by Apollo, it consists of two uppercase alphabetic characters followed by two hexadecimal and two decimal numeric characters.

media_type
is c if the distribution media was cartridge tape, **m** if magnetic tape, and **f** if floppy disk.

For example, a file on a cartridge tape supplied by Apollo might be named **toc.apollo.ST0104.c**.

Node Installation Records

The installation record files are not strictly part of an Authorized Area. They are put in the **/install** directory of a node when Domain/OS is installed in order to guide future installations to the node. If an Authorized Area is the node entry directory of a node that has Domain/OS installed, the **/install** directory of the node doubles as the **install** directory of the Authorized Area.

The /install/baseline Directory

The **baseline** directory contains files named **baseline.***n* where *n* is an integer. The **install** tool uses the most recent **baseline.***n* file, if any, to determine the current software configuration on the node. It uses the information it finds in the most recent **baseline.***n* file to avoid installing products that have already been installed on the node, and to avoid updating files that are already current.

When it completes an installation, the **install** tool creates a new **baseline.***n* file to record the new configurations of products installed on the node.

Note that if two dependent products are listed in the baseline file, and you attempt to install an update of either of the products after removing one of them from the Authorized Area, **install** reports an error and does not install the product.

The /install/preserve.list File

You can create a **preserve.list** file on a target node, listing the absolute pathnames of all files on that node that should be preserved during an installation. The **install** tool checks the contents

of the **/install/preserve.list** file and will not overwrite any of the files named in it. The pathnames in a **preserve.list** file must be relative to the node entry directory (/).

The /install/not_installed File

If, for some reason, the **install** tool cannot install an object, it logs the pathname of the object in the **/install/not_installed** file. A subsequent run of **install** will attempt to install any objects it finds in the **/install/not_installed** file, and will update the file as necessary upon completion.

The /install/rai_acl_temp.?* Files

Any object that existed on the target of the installation and whose initial permissions could not be preserved during the installation is preserved in a file named **/install/rai_acl_temp.***number*. The *number* is an arbitrary numeric value assigned by the installation program. The installation transcript shows the original pathnames for each of the **/install/rai_acl_temp.***number* files created during an installation.

Authorized Areas and Distribution Media

file 1 on the first volume of every product's distribution media contains all components of an Authorized Area except the actual products (i.e., the product release directories). Specifically, **file 1** contains the installation tools and their associated help files; the release documentation for the product; the predefined override, configuration, and selection files; and the toc files. The version of the tools shipped with a product in **file 1** is the latest version available when the product was released, not when the product was shipped to you. The **distaa** tool loads only the product release directories, not the contents of **file 1**.

> NOTE: At some point after the release of SR10.2, optional products will no longer be released with the installation tools on their distribution media.

When you load Domain/OS for the first time using **minst** (as we describe in Chapter 2), **minst** calls **rbak** to load the contents of

file 1 into an Authorized Area, and then calls **distaa** to load the product release directories. When you load Domain/OS using **distaa** (as we describe in Chapter 5), we instruct you to **rbak** all of **file 1** before you load the actual Domain/OS product with **distaa**. This ensures you have the latest version of the installation tools, as well as the other Authorized Area components.

When you load an optional product from distribution media into an Authorized Area using **distaa**, we instruct you to **rbak** the product release documentation and optionally the templates from **file 1**. Users experienced with the installation tools may wish to **rbak** additional Authorized Area components from **file 1** of an optional product tape.

Be sure to always read the release documentation before you run **distaa** to load a product from distribution media. Pay particular attention to the product-specific installation information provided in Chapter 2 of a product's release notes.

The Apollo Installation Model in Detail

As stated in Chapter 1, software installation is a two-step process. First, you load a product from the software distribution media into an Authorized Area on the Authorized Area node. Second, you install an operational configuration of the product from the Authorized Area to a target node, which may or may not be the Authorized Area node. Now that you know the basic tools and data files used during installation, it is possible to explain the Apollo installation model in finer detail.

Release Indexes, Selection, Override, and Configuration Files

As stated earlier in this chapter, a product is released together with a release index. The release index is a plan for transforming the contents of a product release directory into an operational software configuration on a node. It describes all the possible configurations allowed for the product, including lists of every object in the release and each object's type, attributes, permissions (ACLs), and dependencies on other objects in the release and on software previously installed on the target node. The release index indicates which parts of the product are required and which are optional, and contains a series of questions whose answers will determine which

of the possible product configurations is actually installed on a node.

Installation, in its broadest sense, consists of selectively narrowing the options afforded by a product's release index down to a single product configuration, described by a configuration file, desired on a node. The Apollo model is extremely flexible about where in the installation process that narrowing takes place, which options remain available at each phase of the process, and who controls the process.

When you load a product from distribution media to an Authorized Area, you can use a selection file with the **distaa** tool to restrict which components of a product are loaded. You then use the **config** tool, either directly or via **install++**, to further restrict the product configuration that is actually installed on a node. Each selection file has a corresponding override file; the override file restricts the set of configuration choices presented by **config** so the choices are consistent with the product subset specified by the selection file.

For very simple products, or at a site consisting of only a few workstations and administered by a single person, it may make sense to preserve all the choices allowed by the release index until the creation of the configuration file. At larger sites with more decentralized responsibility for system administration, selection and override files can be useful administration tools.

A Conceptual View of the Installation Process

We are ready to look at the entire installation process at a conceptual level. This is not a recipe for installing a product at your site, and it ignores most of the details involved in actually running the tools. It is only meant to show you how the tools were designed to work together. This conceptual description begins with a release index and ends with an installed product. Some of the steps covered are optional.

Phase 1. Selecting Products from the Distribution Media

The first phase is the creation of selection and override files for a product set. They are created by **cfgsa**, an interactive program that prompts the user with questions from a product release index and uses the answers, together with the dependency information coded into the release indexes, to produce a selection file and associated override file.

The selection file is used in Phase 2 to restrict what **distaa** loads into the Authorized Area. A large product, such as Domain/OS, can be shipped with a set of selection files in the **install/templates** directory, in which case this step takes place before the **install** directory is created on the distribution media; that is, before you ever receive the product. Each of the selection files shipped in the **install/templates** directory with the product defines a self-consistent subset of the objects that make up a product.

Every selection file is paired with an override file. The override file is used in Phase 4 to restrict the questions asked by **config** to those consistent with the subset of the product defined by the associated selection file. In other words, **cfgsa** helps you select a subset of objects from a released product that will still result in a functional product when installed, and produces two files based on your selections: a selection file that directs **distaa** to load only the chosen product components into the Authorized Area, and an override file that directs **config** to ask only those questions consistent with the chosen components.

Phase 2. Loading Software into the Authorized Area

The next phase is the loading of software into the Authorized Area with **distaa**. The **distaa** tool is not interactive. It either takes its orders from a selection file or, if no selection file is specified, it loads all the products from the distribution media into the Authorized Area. You can also use **distaa** to modify the contents of an Authorized Area based on the selection file created with **cfgsa**.

If you load only a subset of a product into an Authorized Area by using a selection file when invoking **distaa**, you must also make sure the override file associated with that selection file is in the **install/overrides** directory of the Authorized Area. Unless an override file is in the **install/overrides** directory, **config** will ignore the override file's restrictions in Phase 4 and assume that the entire product, as defined by the release index, is available. As a result, **config** may produce a configuration file that cannot be satisfied by the subset of the product present in the Authorized Area.

Phase 3. Constraining Subsequent Product Configurations

It is possible at this point to further restrict future product configurations to a subset of those supported by the software loaded into the Authorized Area in Phase 2. You can do this by creating new

override files in the **install/overrides** directory for the products in the Authorized Area by using **cfgsa**. The **cfgsa** tool uses only the release indexes, and any currently active override file, to determine what product configurations are possible; it doesn't check that the necessary objects actually exist in the release directories of the Authorized Area.

Phase 4. Establishing a Product Configuration

The next phase is the creation of a configuration file describing a single configuration of a product. You create a configuration file by running the interactive tool **config** and answering the questions it asks. The **config** tool scans all the release indexes and override files in an Authorized Area to determine the product configurations still allowed, and asks any questions provided in the release index that haven't already been answered in an associated override file.

The result of running **config** is a configuration file describing a single configuration of a product. The configuration file is used by the **install** tool in Phase 5 to determine the operational product configuration on the target node.

Phase 5. Installing the Product Configuration

You next run **install** to assemble an operational product configuration on the target node. The **install** tool uses the product release indexes in an Authorized Area, any active override files and **excludes.list** file there, and the configuration file specified on its command line to determine the configuration. It does not prompt the user.

The **install** tool uses the latest **baseline.***n* file, if one exists, to avoid installing products that are already installed, and to avoid updating files that are already current. If the **/baseline** directory is empty or doesn't exist, the **install** tool assumes the product is not installed. Therefore, the absence of a valid **baseline.***n* file can slow the operation of **install** considerably.

If the target node has a **/install/not_installed** file, **install** also attempts to install any of the files listed there from the Authorized Area. If the target node has a **/install/preserve.list** file, **install** does not install the objects listed in the file; it issues warnings to that effect and continues with the installation.

The **install** tool lists the pathnames of any objects it was unable to install in the **/install/not_installed** file on the target node. It also records the target node's current software configuration in a file named **baseline.***n* in the target node's **/install/baseline** directory.

Phase 6. Protecting the Installed Software

In the final phase, if an **install/protections.list** file exists, **install** uses the **inprot** tool to set permissions on objects according to the permissions specified in the **protections.list** file. You can also use the **inprot** tool after software installation to make global changes to object permissions on a node.

Why install++ and minst

As stated earlier, there are two installation tools layered on top of the basic tools: **install++** and **minst**. They provide a more interactive and easier to use, but less flexible, interface to the basic tools. They do not provide any additional functionality over the basic tools.

What is install++?

The **install++** tool is an interactive interface to the **config** and **install** tools. It combines Phase 4 and Phase 5 of the installation process by first invoking **config** to create a temporary configuration file and then invoking **install** with that configuration file to install the product configuration. Whereas the **config** and **install** tools take several required arguments on the command line, **install++** has only two required arguments. In using **install++**, you trade some flexibility for ease of use.

The **install++** tool performs a quick installation of a single product configuration to one or more nodes in a network. It allows you to install a product from an Authorized Area without understanding what a configuration file is, and takes care that all the information necessary for successful cooperation between the **config** and **install** tools remains consistent between their separate invocations. However, unless you specify the name of a configuration file when you invoke **install++**, **install++** does not save the configuration file it generates. In this case, you must reconfigure the same products the next time you install them.

What is minst?

As mentioned earlier, the **minst** tool is an interactive interface to **distaa** and **install++** that makes installing Domain/OS for the first time in a pre-SR10 network easier. **minst** invokes **rbak** to load the installation tools and other administrative files from the Domain/OS distribution media into an Authorized Area, invokes **distaa** to load Domain/OS into the Authorized Area, and calls **install++** to install Domain/OS from the Authorized Area onto the first node. In the context of our conceptual model, **minst** performs Phase 1, Phase 2, Phase 4, and Phase 5 in one swoop.

You could perform the same tasks without **minst** by invoking the **rbak**, **distaa**, and **install++** tools explicitly. However, **minst** does not require you to understand the structure of the Authorized Area or know the contents of the distribution media. In novice mode, **minst** simply loads the selected configuration of Domain/OS from the distribution media into the Authorized Area and installs this configuration on the node. In expert mode, **minst** allows for more customization, but not as much as is possible with the basic tools.

Chapter 5

Managing Authorized Areas

This chapter describes tools and procedures for managing an Authorized Area. The discussions in this chapter expand on the concepts discussed in Chapter 4.

The cfgsa, distaa, and mrgri Tools

Three installation tools are provided for managing Authorized Areas:

cfgsa	Creates selection and override files
distaa	Loads products into an Authorized Area
mrgri	Merges two products into one product

The **cfgsa** tool is used to create pairs of selection and override files. Selection files can be used with the **distaa** tool to load only parts of products, rather than entire products, from distribution media into an Authorized Area. For instance, if your site has restricted product configurations, selection files enable you to avoid wasting disk space on product components that are not needed to support the restricted configurations.

Override files pre-answer some of the questions usually presented to a user during the configuration phase of the installation process.

You can use an override file to ensure that the possible configurations for a product are consistent with the subset of a product loaded from media with the corresponding selection file. Or you can use an override file to simply further restrict the installable configurations of a product in an Authorized Area.

The **mrgri** tool merges two products from an Authorized Area into a single product, producing a new release index for the merged product in the process. For example, you can use it to patch products already loaded into the Authorized Area, and then install the product and patch as a single product. You can also use it to merge two releases of a single product version meant for different machine types into a single product that can be installed and run on machines of either type.

Restricting Available Product Configurations

The **cfgsa** tool has two functions. One is to create an active override file for a product already loaded into an Authorized Area. The override file will restrict the choices a user can make when configuring a product to a subset of the possible choices allowed by the product's release index. The other is to create a selection and override file pair for use in consistently modifying an Authorized Area with **distaa**. This section describes how to create an active override file for a product already loaded into an Authorized Area.

When you execute **cfgsa**, it first scans the Authorized Area for the products available in it. When you select a product, **cfgsa** displays all the questions that would be shown to users if they were configuring that product with the **install++** or **config** tool. For each question displayed, you may choose to:

- Allow the user to answer the question
- Limit the possible answers to the question
- Answer the question so the user will not see question at all

You can then direct **cfgsa** to save the constraints you selected in an active override file in the directory *AA*/**install/overrides**. The constraints apply to all subsequent configurations of the product; any previous active override file for the product is replaced.

Creating an Active Override File

The **cfgsa** tool is interactive and its commands can be abbreviated to the point of uniqueness. For instance, at the CFGSA> prompt, the **help** command can be abbreviated as **h**, but the **select** and **save** commands can be abbreviated only to **se** and **sa**, respectively.

To create an active override file for a product,

1. Invoke **cfgsa** by entering

   AA/**install/tools/cfgsa** *AA*

   where *AA* is the pathname of the Authorized Area containing the products you want to constrain.

   The program displays a list of the products available in the Authorized Area and the prompt changes to CFGSA>.

   NOTE: If at any time you want to redisplay the list of available products, enter the **available** command at the prompt. At any point in the program, the **help** command displays a brief summary of the commands available.

2. Select a product to be preconfigured with the **select** command:

   select *product_name*

   where *product_name* is the name of the product as it appears on the list of available products.

   or

   select *product_number*

   where *product_number* is the number to the left of the product in the list of available products.

3. Preconfigure the product by typing **constrain**.

   The **cfgsa** program loads the release index for the product last selected. It then displays the questions that are shown to a user during a configuration session for that product,

and displays the possible answers that a user can supply. For each question, you are given the following choice of options:

answer
Answer the question for the user; the user will not be shown this question at all during later configuration sessions.

limit
Limit the user's choices for this question; when the question is presented during later configuration sessions, the user will be presented with a reduced answer set that you specify.

user
Let the user see and answer the question as it is shown here.

help
List the **constrain** options as shown here.

refresh
Redisplay the question and answer.

abort
Exit constrain mode for the currently selected product. The constraints already applied are saved.

<RETURN>
A carriage return chooses the default option, which is **user**.

Note that choosing **limit** may result in users being presented with slightly confusing questions during later configuration sessions because they will only be allowed to choose from a subset of the options expressed in the question. For example, while in **cfgsa** you might give the following responses to the following questions:

```
QUESTION: /DOMAIN_EXAMPLES
The /domain_examples directory contains online programming
examples. If this directory exists and is NOT a link on the
target, you may want to install /domain_examples/cc.
If /domain_examples IS a link from the target to another
node, then you must install cc on that node if you
want the latest version of these examples to be available
```

```
on the target.
Do you want a local copy of the online examples
for cc, a link to another node or neither?

ANSWERS:  Up to 1 of [ copy(D) link none ]

YOUR CHOICE [ Answer Limit User(D) Help Refresh Abort ]: limit

Pick 3 of [ copy(D) link none ]: link none
```

With the above responses, subsequent users installing the software will see the question below:

```
/DOMAIN_EXAMPLES
The /domain_examples directory contains online programming
examples.  If this directory exists and is NOT a link on the
target, you may want to install /domain_examples/cc.
If /domain_examples IS a link from the target to another
node, then you must install cc on that node if you
want the latest version of these examples to be available
on the target.

     The examples files for cc require on the order of
     .5 MB of disk space.

Do you want a local copy of the online examples
for cc, a link to another node or neither?
: [ link none ]
==>
```

When you finish constraining a product or if you **abort**, the program returns to the CFGSA> prompt.

4. If you change your mind about the constraints you have made for the selected product, you can remove the constraints on it, using the **revert** command. **revert** removes any constraints you've applied to the selected product during the current **cfgsa** session.

5. Enter the **save** command to create an active override file for the selected product in the *AA*/**install/overrides** directory. The file is named **ri.***company.product_name*.**v.***version*. Any override file for the selected product already present in the *AA*/**install/overrides** directory is overwritten with the new one. Later configurations of the product use the product's active override file to determine what questions are asked of the user at configuration time.

6. When you have finished constraining as many products as you want, enter **exit** to leave **cfgsa**.

You can later restore the range of possible configurations previously supported by the product by removing the override file from the *AA*/**install/overrides** directory. The **cfgsa** tool does not modify the contents of product release directories. You need to run the **distaa** tool to do that.

> NOTE: We recommend that you never delete the active override file for a product unless all components of that product are loaded in the Authorized Area. If your Authorized Area doesn't contain all of the components of a product, then removing the currently active override file for that product restores the full range of possible configurations supported by the release index, and that may be more configurations than are supported by the subset of the product present in the Authorized Area. The result is that **config** or **install++** can create product configurations that are inconsistent with the product as it was loaded into the Authorized Area, which will cause numerous error messages at installation time.

Loading New Products into an Authorized Area

This section describes how to load new products into an Authorized Area from media with the **distaa** tool. You use these procedures to load optional products and to load an update of the Domain/OS product (that is, to load SR10.x on a node running an earlier version of SR10.x). Once you load a product into an Authorized Area, you can install an operational configuration of the product on a node using the procedures in Chapter 6 and Chapter 7.

You can either load all components of a product present on the media or use a selection file to load only some of the components. Also, if the distribution media contains more than one product, you can use selection files to load only some of the products. You can use selection files that are supplied on the distribution media for each product or use ones you create with the **cfgsa** tool.

NOTE: To load an SR9.7-compatible optional product from distribution media onto an SR9.7 node, read the subsection "Loading Optional Products onto an SR9.7 Node" (at the end of this section) before you attempt to perform the following procedures.

Using distaa to Load All Available Product Components

Use the following procedure to load all components of *all* products on the distribution media. (A single set of distribution media can contain more than product.) This is the simplest way to load optional products; you can always remove components of a product from the Authorized Area, as described later in this chapter. To load only some components of a product or to load only some products, if the media contains more than one product, use the procedure in the next subsection.

If you are loading Domain/OS, make sure you have enough disk space before you use this procedure; it loads all components of all three operating-system environments (Aegis, SysV, and BSD). Chapter 2 of the SR10.x release notes provides the size of this configuration (the Large Aegis/SysV/BSD configuration). If you do not have enough disk space, use the procedure described in the section "Using distaa to Selectively Load Product Components" or the procedure in "Distributing Products when Loading from Distribution Media" later in this chapter.

To load all components of all products into an Authorized Area,

1. Put the first volume of the distribution media in the drive. (If you are loading Domain/OS, insert the first product volume, not the boot volume.) Then enter this command (on one line):

   ```
   AA/install/tools/rbak_sr10 -dev dev
         -f 1 -ms -sacl -pdt -force -du
         {install/doc -as AA/install/doc | -all}
   ```

   where:

   AA
   is the pathname of the Authorized Area into which the software is to be loaded.

dev
is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

install/doc -as *AA*/**install/doc**
is entered if you are loading an optional product.

-all
is entered if you are loading Domain/OS.

If you are loading an optional product, this command loads the release documentation from **file 1** on the media into the Authorized Area. If you are loading Domain/OS, this command loads all of **file 1**—the release documentation, the installation tools, the templates—from the media. For more information about **file 1**, see the "Authorized Areas and Distribution Media" section in Chapter 4.

2. Read the release documentation for each product on the distribution media. The documentation is located in the directory *AA*/**install/doc/***company_name*. Pay particular attention to Chapter 2 of the release notes. This chapter discusses product-specific installation issues and dependencies.

3. Load the product(s) into the Authorized Area, using this command line:

   AA/install/tools/distaa -f -m *dev* -a *AA*

   where:

   AA
   is the pathname of the Authorized Area into which the product is to be loaded.

   dev
   is **c** for cartridge tape, **f** for floppy disk, or **m** for magnetic tape.

   The **distaa** program loads the product(s) into the Authorized Area. If the products require more than one piece of media, you are prompted during the loading process to remove media from the drive and replace it with the next piece of media.

4. Perform this step only if you had loaded the product you just loaded on some earlier occasion. Copy the override file

AA/**install/templates/***company/product_name*.**v.***version*/**ov.***template*

where:

template
is the override file that specifies the configuration that includes all product components

to the file

AA/**install/overrides/ri.***company.product_name*.**v.***version*

This step makes the new override file active and removes any previously active override file for the product.

Using distaa to Selectively Load Product Components

Use the following procedure to load just some components of a product into an Authorized Area, rather than the entire product. Also use this procedure if the distribution media contains more than one product and you do not want to load all of the products.

In both cases, you supply the **distaa** tool with the name of a selection file for each product you want to load. A set of selection files and corresponding override files are shipped with each product for this purpose. Each product ships with at least one selection file that loads all components of the product. You use this type of selection file when the distribution media contains more than one product but you only want to load some of the products; for each product you want to load, you supply the name of the selection file that loads all components of the desired product. Some products, such as Domain/OS, also ship with selection files that cause **distaa** to load only some components of the product.

The selection and override files reside in the directory **install/templates/apollo/***product_name*.**v.***version* in **file 1** of the distribution media. The release notes for each product describe which components of the product are loaded with each selection file. You can also use a selection file that you create with the **cfgsa** tool.

To selectively load components of one or more products into an Authorized Area using a predefined selection file,

1. Put the first volume of the distribution media in the drive. (If you are loading Domain/OS, insert the first product volume, not the boot volume.)

If you are loading an optional product, enter this command (on one line):

AA/**install/tools/rbak_sr10 -dev** *dev*
-f 1 -ms -sacl -pdt -force -du
install/templates -as *AA*/**install/templates**
install/doc -as *AA*/**install/doc**

where:

AA
is the pathname of the Authorized Area into which the software is to be loaded.

dev
is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

This command loads the release documentation and Apollo-supplied selection and override files from **file 1** on the media into the Authorized Area. For more information about **file 1**, see the "Authorized Areas and Distribution Media" section in Chapter 4.

If you are loading an updated version of Domain/OS, enter this command line:

AA/**install/tools/rbak_sr10 -dev** *dev*
-f 1 -ms -sacl -pdt -force -du -all

where:

AA
is the pathname of the Authorized Area into which the software is to be loaded.

dev
is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

This command loads all of **file 1**—the release documentation, the installation tools, the selection and override files—from the media into the Authorized Area.

2. Read the release documentation for each product on the distribution media. The documentation is located in the directory *AA*/**install/doc/***company_name*. Use the

information provided in Chapter 2 of each product's release notes to determine which Apollo-supplied selection file you want to use. Also pay particular attention to product-specific installation issues and dependencies discussed in Chapter 2.

Selection files have names of the form

AA/**install/templates/***company*/*product_name*.**v.***version*/
aa.*template_name*

where:

company
is the name of the company that developed the product. Apollo products use **apollo** for *company*.

product_name
is the name of the software product recognized by the installation tools. A Pascal compiler might use **pas** for *product*; a FORTRAN compiler might use **ftn**.

version
is the version number of the product, for example **10.0** or **2.1**.

template_name
is the name of the template, for example **aegis_small**.

An example of a selection file pathname is *AA*/**install/templates/apollo/os.v.10.2/aa.aegis_small**.

3. Load the product components into the Authorized Area, using this command line:

AA/**install/tools/distaa** **-f** **-m** *dev AA selection_file* [*selection_file*]

where:

AA
is the pathname of the Authorized Area into which the product components are to loaded.

dev
is **c** for cartridge tape, **f** for floppy disk, or **m** for magnetic tape.

selection_file [*selection_file*]
is a list of selection file pathnames, one for each product you want to load.

The **distaa** program loads the product components into the Authorized Area. If a product requires more than one piece of media, you are prompted during the loading process to remove media from the drive and replace it with the next piece of media.

4. Copy the override file associated with each selection file you used to the file

   AA/**overrides/ri.***company.product_name*.**v.***version*

   The override file associated with each selection file resides in the same directory as the selection file and has the same name, except the prefix **ov** is used instead of **aa**. This step makes the override file active, ensuring that questions presented during product configuration are consistent with the subset of the product that you loaded.

Loading Optional Products onto an SR9.7 Node

To load an SR9.7–compatible optional product from distribution media to an SR9.7 node, you can use the procedures in the previous two sections with these differences:

- If an Authorized Area already exists on the SR9.7 node, use the command *AA*/**install/tools_sr9/rbak_sr9** instead of *AA*/**install/tools/rbak_sr10**.
- If an Authorized Area does not exist on the node, use the following **rbak** command line, instead of the one shown in step 1 of the previous procedures:

  /**com/rbak** -dev *dev* -f 1 -ms -sacl -pdt -force -du -all

  where:

  dev
  is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

- Use the command *AA*/**install/tools_sr9/distaa** instead of *AA*/**install/tools/distaa**.

If you have at least one Authorized Area on a node running SR10.x, you can install an SR9.7–compatible product on an SR9.7 node, using the standard tools and procedures. Load the product into the Authorized Area on the Domain/OS node, then run the tools on the Domain/OS node to install the product from the Authorized Area to the SR9.7 node.

Removing Software from an Authorized Area

Occasionally you may want to remove unneeded product components from an Authorized Area. In most cases this will never be necessary. If a product requires a large commitment of disk space for its installation and supports more than one possible operating configuration, that product ships with a group of selection files and associated override files created to support installation of those configurations.

We cannot, however, anticipate every situation, and you may have a need to customize a product still further than what is defined in the selection files that ship with the product. You can use the **cfgsa** tool together with **distaa** to constrain a product's configurations to those you wish to support at your site, and then remove any product components not needed to support your restricted set of product configurations.

To do so, run **cfgsa** to restrict the product configurations to those you want in the Authorized Area. Use the **generate** command to create a selection file and an override file implementing your restrictions. Then make the override file created by **cfgsa** the active override file for the product, delete the product release directory for the product from the Authorized Area, and finally use **distaa** with the selection file created by **cfgsa** to reload only the components you need. In other words, you first decide which components of a product you want to retain, then delete the entire product from the Authorized Area and reload only the components you want.

Whenever **cfgsa** creates a selection file, it also creates an override file that constrains the choices users will be able to make during subsequent installations. The tool works by reading the release indexes from an Authorized Area; when you select a product, the program displays all the questions that would be shown to a user configuring that product with the **install++** or **config** tool. For each question displayed, you may choose to

- Allow the user to answer the question
- Limit the possible answers to the question
- Answer the question (User will not see question at all.)

You can then direct **cfgsa** to create a selection and an override file based on your choices. The override file will, when placed in the *AA*/**install/overrides** directory, restrict the choices the user can make to a subset of the possible choices allowed by the release index consistent with the selection file. The selection file is used with **distaa** to load only the product components needed to support the restricted choices allowed by the override file.

The cfgsa Session

The **cfgsa** tool is interactive and its commands can be abbreviated to the point of uniqueness. For instance, at the CFGSA> prompt, the **help** command can be abbreviated as **h**, but the **select** and **show** commands can be abbreviated only to **se** and **sh**, respectively.

To use **cfgsa** to constrain installable product configurations, follow the steps below. This procedure is identical to that for creating an active configuration file up to Step 5.

1. Invoke **cfgsa** like this:

   AA/**install/tools/cfgsa** *AA*

   where *AA* is the pathname of the Authorized Area containing the products you want to constrain.

   The program displays a list of the products available in the Authorized Area, and the prompt changes to CFGSA>.

   NOTE: If at any time you want to redisplay the list of available products, enter the **available** command at the prompt. At any point in the program, the **help** command displays a brief summary of the commands available.

2. Select a product to be preconfigured with the **select** command like this:

select *product_name*

where *product_name* is the name of the product as it appears on the list of available products.

or

select *product_number*

where *product_number* is the product number on the list of available products.

3. Preconfigure the product by typing **constrain**. The **cfgsa** program loads the release index for the product last selected. It then displays the questions that are shown to a user during a configuration session for that product, and displays the possible answers that a user can supply. For each question you are given the following choice of options:

   answer
   Answer the question for the user; the user will not be shown this question at all during later configuration sessions.

   limit
   Limit the user's choices for this question; when the question is presented during later configuration sessions, the user will be presented with a reduced answer set that you specify.

   user
   Let the user see and answer the question as it is shown here.

   help
   List the **constrain** options as shown here.

   refresh
   Redisplay the question and answer.

   abort
   Exit constrain mode for the currently selected product. The constraints already applied are saved.

   <RETURN>
   A carriage return chooses the default option, which is **user**.

Note that choosing **limit** may result in users being presented with slightly confusing questions during a later configuration session because they will only be allowed to choose from a subset of the options expressed in the question. For example, while in **cfgsa** you might give the following responses to the following questions:

```
QUESTION: /DOMAIN_EXAMPLES
The /domain_examples directory contains online programming
examples. If this directory exists and is NOT a link on the
target, you may want to install /domain_examples/cc.
If /domain_examples IS a link from the target to another
node, then you must install cc on that node if you
want the latest version of these examples to be available
on the target.
Do you want a local copy of the online examples
for cc, a link to another node or neither?

ANSWERS:  Up to 1 of [ copy(D) link none ]

YOUR CHOICE [ Answer Limit User(D) Help Refresh Abort ]: limit

Pick 3 of [ copy(D) link none ]: link none
```

With the above responses, subsequent users installing the software will see the question below:

```
/DOMAIN_EXAMPLES
The /domain_examples directory contains online programming
examples.  If this directory exists and is NOT a link on the
target, you may want to install /domain_examples/cc.
If /domain_examples IS a link from the target to another
node, then you must install cc on that node if you
want the latest version of these examples to be available
on the target.

     The examples files for cc require on the order of
     .5 MB of disk space.

Do you want a local copy of the online examples
for cc, a link to another node or neither?
: [ link none ]
==>
```

When you finish preconfiguring a product or if you abort, the program returns to the CFGSA> prompt.

4. If you change your mind about the constraints you have made for the selected product, you can remove the constraints on it by typing **revert**. The **revert** command removes any constraints you've applied to the selected product.

5. You can now use the **generate** *template* command to produce both a selection file and an override file in the current directory. The selection file that **generate** *template* creates has the name **aa.***template*, and the override file has the name **ov.***template*.

6. When you have finished creating as many selection/override file pairs for as many products as you want, enter **exit** to leave **cfgsa**.

More on the generate Command

The **cfgsa** tool has two functions. One is to create an active override file for a product to constrain the options presented to users of **config** or **install++** during the product configuration phase of the installation process. The other is to create a selection and override file pair for use in safely modifying an Authorized Area with **distaa**.

The **generate** *template* command creates both a selection and an override file in the current directory. These files are analogous to the **aa.***template* and **ov.***template* files that can ship with a product in the *AA*/**install/templates/***company.product_name*.**v.***version* directory. You can use the **aa.***template* file with **distaa** to remove unwanted pieces of a product already loaded into the Authorized Area, as described in the next section.

Preparing to Reload the Product Components

Before you can use **distaa** to load the new selection of components for a product defined by the selection file you created, you must first delete the old product release directory and activate the associated override file. This keeps future installations of the product consistent with the configurations supported by the subset of the product described by the selection file.

1. Make the override file created by **cfgsa** the active override file for the product by moving it from **ov.***template* in the current directory to the file

AA/**install/overrides/***release_index_name*

where *release_index_name* is the name of the release index file for the product. The release index name has the form

ri.*company.product_name***.v.***version_number*

where **ri.** is the prefix denoting a release index file name, *company* indicates the company that produced the product, *product_name* is the name of the product, and *version_number* is the product version number (for example, **ri.apollo.os.v.10.2**).

The release index file name is also the name applied to the product release directory containing the release index and the product components.

2. Delete the entire existing release directory for the product with one of the following command lines:

   (UNIX environments)
   rm -rf *AA*/install/*release_index_name*

   or

   (Aegis environment)
   dlt -f *AA*/install/*release_index_name*

You can now reload the selected product components, as shown in the next section.

Loading the New Selection of Product Components

You are now ready to load only the product components necessary to support the constraints you elected in the **cfgsa** session. Run **distaa** to load the new selection of product components as defined in the new selection file:

AA/**install/tools/distaa** -m *dev AA* **aa.***template*

where:

dev is **c** for cartridge tape, **f** for floppy disk, or **m** for magnetic tape.

AA is the pathname of the Authorized Area into which software is to be loaded.

aa.*template*
is the name of the selection file created with **cfgsa**.

The **distaa** program loads software into the Authorized Area. If the product you are loading requires more than one piece of media, you are prompted during the loading process to remove media from the drive and replace it with the next piece of media.

Merging Products in an Authorized Area

The **mrgri** (merge release indexes) tool has two main uses. It can merge a patch with the product it modifies to create a patched product that can be installed in a single pass. In this case, installation of the merged product has the same result as would installing the product first and then installing the patch on top of it. The advantage of merging the patch is that there is only one product in the Authorized Area to manage and install instead of two.

The other main use is the creation of compound products. With the advent of the Series 10000 workstations and servers, Apollo releases two versions of most products: a version that runs on nodes based on Motorola's 68000 series of microprocessors, and a version that runs on the Series 10000 *PRISM* (*P*arallel *R*educed *I*nstruction *S*et *M*ultiprocessor) nodes. We encapsulate these two CPU architectures with the concept of an ISP type, where ISP stands for Instruction Set Processor. The Series 10000 nodes belong to the **a88k** ISP type, and all other nodes belong to the **m68k** ISP type.

In product releases, we distinguish between the two ISP types by adding the extension **.p** on the version field of the release index name for products with the **a88k** ISP type. Thus, **ri.apollo.os.v.10.2.p** is the name of the release index for the version 10.2 of Domain/OS that runs on the Series 10000 nodes, whereas **ri.apollo.os.v.10.2** is the name of the release index for the version that runs on nodes with the **m68k** ISP type. You can sometimes use **mrgri** to merge two products that differ only in ISP type into a single product that can be installed and run on nodes of both ISP types. A product that is the result of merging versions with different ISP types is called a **compound product**.

Limitations of mrgri

The **mrgri** tool uses the release indexes of the products it merges to control how it combines the objects that constitute the products. In order for two versions of a product to merge successfully, their release indexes must contain a consistent set of configuration dependencies for the products' constituent objects. In other words, the products must be designed to merge successfully, and their release indexes must reflect that design. Therefore, you should not indiscriminately merge any two products. The release notes for a product should state whether that product can be successfully merged with **mrgri**, which versions should be merged together, and what restrictions you must follow when installing and using the merged product.

You should also be aware that a product created by **mrgri** cannot be unmerged to recover its constituent products. If you delete the individual product release directories after merging them, or if you use **mrgri** to overwrite one of the individual products with the new merged product, that product cannot be recovered from the merged product. You must reload it from the distribution media or copy it from another Authorized Area.

Merging Patches

Before you can create a patched product, the patch and the product to be patched must be loaded into the same Authorized Area. You can patch a product in the Authorized Area with the following command line:

AA/install/tools/mrgri -s *AA*
product_release_index patch_release_index

where:

AA is the pathname of the Authorized Area containing both the product and the patch.

product_release_index
is the name of the release index file for the product.

patch_release_index
is the name of the release index file for the patch.

The resulting merged product is placed in the product release directory of the unpatched product and has the same name as the unpatched product. **mrgri**, in effect, applies the patch directly to the product in the product's current release directory.

Merging ISP Types

As stated before, the **mrgri** tool takes the two product release directories specified on the command line and merges them into a single product release directory with a single release index. When **mrgri** encounters two executable objects with the same name but different ISP types, it combines the two objects to create a compound executable type (**cmpexe**). The **cmpexe** object is an executable object that contains code for both ISP types. The result of this kind of merge is a compound product that can be installed and run on nodes of either ISP type.

For example, you can use the **mrgri** tool to merge the 10.2.p version of Domain/OS with the 10.2 version, and produce an operating system that can be installed and run on nodes of either ISP type. You can also boot nodes of either ISP type diskless off the node running the merged Domain/OS, and let nodes of either ISP type link to the node running the merged Domain/OS for OS components you don't want to duplicate on every node.

The following **mrgri** command line merges an **a88k** version of a product with an **m68k** version of the same product to create a new product that can be installed and run on Series 10000 nodes and other Apollo nodes.

AA/install/tools/mrgri -s *AA* -v *version_number*.cmpexe
prism_product_release_index product_release_index

where:

AA is the pathname of the Authorized Area containing both the product and the patch.

version_number
is the version number of the *PRISM* product, including the **.p** suffix. (The version number used after the -v option to **mrgri** should be the same as the one in the *version_number* field in *prism_product_release_index*.)

prism_product_release_index
is the name of the release index file for the *PRISM* version of the product. It has a **.p** suffix on the version number.

product_release_index
is the name of the release index file for the **m68k** version of the product. It doesn't have a **.p** suffix on the version number.

A compound product is larger than either one of its constituent products because each compound executable (**cmpexe**) in the compound product contains executable code for each ISP type. However, since products are not composed entirely of executable objects, the size of a compound product is less than a simple sum of the sizes of the two products before merging.

A good rule of thumb is that a compound product will require about 70 percent of the disk space occupied by the two unmerged products together. If you lack the requisite additional 70 percent, you may want to use the **-t** option to **mrgri**, which causes **mrgri** to create the merged compound product in another Authorized Area. An alternate solution is to let **mrgri** overwrite one of the constituent products by omitting the **-v** and **-t** options from the **mrgri** command line.

The merging of two large and complex products, such as two different ISP versions of Domain/OS, can take a very long time. We have measured execution times of about 12 hours for **mrgri** when operating on complete Domain/OS products.

Creating, Copying, and Moving Authorized Areas

Authorized Areas are directories that can be treated just like any other directory. But because they are usually large and take some effort to build, you should take precautions when manipulating them. This section describes how to manipulate Authorized Areas with common Aegis and UNIX utilities.

The only real distinguishing feature of an Authorized Area is that it must contain an **install** directory, as explained in Chapter 4. Although any directory on any Domain/OS node can serve as an Authorized Area, we recommend that you either use a node entry directory or create a new directory for the Authorized Area. If you

create a new directory for an Authorized Area, that Authorized Area holds only the **install** directory containing the installation tool directories, override directory, product release directories, and other Authorized Area subdirectories. If a node entry directory is used for an Authorized Area, the Authorized Area will contain the usual root-level system directories found on Domain/OS nodes, in addition to the **install** directory required in every Authorized Area.

The node entry directory, /, of all Domain/OS nodes contains an **install** directory of its own. This directory contains the installation record files used by the **install** tool, as explained in Chapter 4. When a node entry directory doubles as an Authorized Area, the **/install** directory of the node doubles as the **install** directory of the Authorized Area. Fortunately, the contents of the **/install** directory present on every Domain/OS node and the **install** directory of an Authorized Area were designed to coexist.

You can copy or move an Authorized Area to another node on a network with the UNIX **cp** command or the Aegis **cpt** command. You do not need the installation tools to do it. To copy an Authorized Area from one directory (*source_AA*) to another directory (*destination_AA*), follow this procedure:

1. If you are using a UNIX environment, you do not have the **cpt** command with its **-md** option. Consequently, if the Authorized Area you wish to copy from is a node entry directory and the destination directory is also a node entry directory, you will have to protect the contents of the existing **/install** directory in the destination Authorized Area by temporarily changing its name. Use the following command to preserve the **/install** directory in the destination Authorized Area:

   mv *destination_AA*/install *destination_AA*/install.tmp

   where *destination_AA* is the pathname of the destination Authorized Area. It may be a node entry directory (*//node_name*) or the pathname of some other directory created for a new Authorized Area (**//admin/AA** for example).

2. If an **install** directory does not exist in the destination Authorized Area, create one with the UNIX **mkdir** command or the Aegis **crd** command. To do this, enter

   (UNIX environments)
   mkdir *destination_AA*/install

(Aegis environment)
crd *destination_AA*/install

where *destination_AA* is the pathname of the destination Authorized Area. It may be a node entry directory (*//node_name*) or the pathname of some other directory created for a new Authorized Area (**//admin/AA** for example).

3. Merge the contents of the **install** directory of the source Authorized Area into that of the destination Authorized Area. Use one of the following command lines:

   (UNIX environments)
   cp -rpsoP *source_AA*/install/* *destination_AA*/install

   (Aegis environment)
   cpt -pdt -sacl -md
   source_AA/**install** *destination_AA*/install

   where:

   source_AA
   is the pathname of the Authorized Area you want to copy.

   destination_AA
   is the pathname of the destination Authorized Area.

4. If you are using a UNIX environment and you protected the contents of the destination Authorized Area's **/install** directory by executing the **mv** command in Step 1, you must now restore those contents to their proper place with the following command line:

   cp -rpsoP *destination_AA*/install.tmp/*
   destination_AA/install

   and remove the temporary **/install.tmp** directory you created in Step 1 like this:

   rm -rf *destination_AA*/install.tmp

You now have a copy of the Authorized Area at *destination_AA*. If your intent was only to move the Authorized Area from *source_AA* to *destination_AA*, delete the Authorized Area at *source_AA*. Do it in whatever way is most familiar to you, but if *source_AA* is a node

entry directory, be sure to delete only the **/install** directory—not all directories in the Authorized Area. You should also preserve **/install/baseline**, **/install/not_installed**, **/install/preserve.list**, and **/install/doc**, if they exist.

Distributing an Authorized Area

An Authorized Area node need not hold all of the objects belonging to its Authorized Area on its own disk. Portions of an Authorized Area may be distributed via soft (symbolic) links to other Domain/OS disks on a network. Distribution of the Authorized Area is possible because an Authorized Area is not a special type of object; it is just a directory with a subdirectory named **install** that contains executables, data files, and product release directories used for software installation and management. Any of the constituents of the **install** directory may be replaced by a symbolic link to another Domain/OS volume.

Using links to distribute an Authorized Area among more than one disk volume allows you to build a single Authorized Area that is larger than the disk space you might prefer to commit from a single volume. It also lets you build multiple Authorized Areas without duplicating products that you want to make available from more than one Authorized Area. Remember, though, that Authorized Areas can be created on Domain/OS nodes only, and that, if you put links to other volumes in an Authorized Area in order to distribute its contents, those links must also resolve to Domain/OS nodes.

Distributing Products Already Loaded in an Authorized Area

It is easy to distribute pieces of an Authorized Area to another Authorized Area on a network. You do not need the installation tools to do it. For instance, to replace a product release directory in one Authorized Area (*source_AA*) with a link to another Authorized Area (*destination_AA*), follow this procedure:

1. Copy the product release directory from the source Authorized Area to the destination Authorized Area. Use one of the following command lines:

   (UNIX environments)
   cp -rpsoP

source_AA/install/*product_release_directory*
destination_AA/install

(Aegis environment)
cpt -pdt -sacl
source_AA/install/*product_release_directory*
destination_AA/install/*product_release_directory*

where *product_release_directory* is the product release directory you want to replace with a link.

2. Delete the product release directory you've just copied from the source Authorized Area. Use one of the following command lines:

   (UNIX environments)
   rm -rf *source_AA*/install/*product_release_directory*

   (Aegis environment)
   dlt -f -du *source_AA*/install/*product_release_directory*

3. Create a link from the source Authorized Area to the destination Authorized Area. Use one of the following command lines:

   (UNIX environments)
   ln -s *destination_AA*/install/*product_release_directory*
   source_AA/install/*product_release_directory*

   (Aegis environment)
   crl *source_AA*/install/*product_release_directory*
   destination_AA/install/*product_release_directory*

If you wish to permit installation from the destination Authorized Area as well, be sure to copy or link any active override files for the product into the **install/overrides** directory of the destination Authorized Area.

> **NOTE:** Do not use this procedure to distribute *subcomponents* of a product (subdirectories of a product release directory) among more than one Authorized Area, since different subcomponents may use the same file.

Distributing Products when Loading from Distribution Media

If a product is too large to load on a single disk, you can distribute it among more than one Authorized Area as you load it from media

with **distaa**. To do so, you first load (using **rbak**) and edit one of the selection files shipped with the product you want to distribute. Selection files are ASCII files and can be edited with any text editor. You then load the product using **distaa**, supplying the name of the selection file as a command line option to **distaa**.

To distribute a product as you load it from distribution media,

1. Put the first volume of the distribution media in the drive. (If you are loading Domain/OS, insert the first product volume, not the boot volume.)

   If you are loading an optional product, enter this command (as one line):

   AA/install/tools/rbak_sr10 -dev *dev*
   -f 1 -ms -sacl -pdt -force -du
   install/templates -as *AA*/install/templates
   install/doc -as *AA*/install/doc

   where:

   AA
   is the pathname of an Authorized Area.

   dev
   is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

   This command loads the release documentation and Apollo-supplied selection and override files from **file** 1 on the media into the Authorized Area. For more information about **file** 1, see the "Authorized Areas and Distribution Media" section in Chapter 4.

   If you are loading Domain/OS, enter this command line:

   AA/install/tools/rbak_sr10 -dev *dev*
   -f 1 -ms -sacl -pdt -force -du -all

   where:

   AA
   is the pathname of an Authorized Area.

   dev
   is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape.

This command loads all of **file** 1—the release documentation, the installation tools, the selection and override files—from the media into the Authorized Area.

2. Choose the selection file you want from those shipped with the product. Chapter 2 of the release notes for the product (just loaded into the directory *AA*/**install**/**doc**/*company_name*) explains which selection files are shipped with the product, and which components of the product will be loaded with each selection file.

   Selection files reside in the directory

   AA/**install**/**templates**/*company*/*product_name*.v.*version*

   The names of the selection files begin with the characters **aa**.

3. Edit the selection file to be used as input to the **distaa** tool. Predefined selection files are composed of lines of the form:

   move *release_index_name product_component destination*

   where:

   release_index_name
   is the name of the release index for a product. For example, the release name for the SR10.2 version of Domain/OS is **ri.apollo.os.v.10.2**. Do not edit *release_index_name*.

   product_component
   is the name of a component of the product as described in the release index. Do not edit *product_component*.

   destination
   is either **-rootaa** or **-nil**. If *destination* is **-rootaa**, the objects that comprise *product_component* are loaded to the appropriate area in the Authorized Area specified on the **distaa** command line (the root Authorized Area). If *destination* is **-nil**, the objects defined to *product_component* are not loaded from the distribution media. You must edit the *destination* field to distribute a product to more than one Authorized Area.

Replace **-rootaa** in the *destination* field for components you want to distribute with the pathname of the Authorized Area (for example, the directory **//AA_2**) to which you want to distribute them. The *destination* pathname should not be the same as the root Authorized Area. Also, the directory you specify must exist, although no Authorized Area components, including the **install** subdirectory, need reside in the directory already.

The objects for which you change **-rootaa** are loaded to the pathname you specify, and a link is created to those objects in the appropriate place in the root Authorized Area.

In the sample selection file shown in Figure 5-1, the first three lines indicate that the Domain/Dialogue objects in the **/com**, **/sys**, and **/usr** subcomponents are to be loaded into the Authorized Area specified on the **distaa** command line. The objects in the **/doc** subcomponent are to be loaded into an Authorized Area on node **//color** (into the directory **//color/install/ri.apollo.dialog.v.3.4**). The objects in the **/examples** subcomponent are not restored from the distribution media.

```
/install/templates/apollo/dialog.v.3.4/aa.small
move   ri.apollo.dialog.v.3.4 /com      -rootaa
move   ri.apollo.dialog.v.3.4 /sys      -rootaa
move   ri.apollo.dialog.v.3.4 /usr      -rootaa
move   ri.apollo.dialog.v.3.4 /doc      //color
move   ri.apollo.dialog.v.3.4 /examples -nil
```

Figure 5-1. Sample Selection File

4. Run **distaa** with the name of the edited selection file as the selection file argument:

   AA/install/tools/distaa -f -m *dev AA selection_file*

   where:

   dev is **c** for cartridge tape, **f** for floppy disk, or **m** for magnetic tape.

AA is the pathname of the Authorized Area you used in the **rbak** command line of Step 1.

selection_file
is the pathname of the selection file that you edited.

The **distaa** program loads the product components with a destination of **-rootaa** into the Authorized Area specified on the command line and loads the products components for which you supplied a pathname to the specified pathname. For the latter components, links are created in the root Authorized Area. If the product you are loading requires more than one piece of media, you are prompted during the loading process to remove media from the drive and replace it with the next piece of media.

Chapter 6

Configuring Products

This chapter describes how to configure products you plan to install. In this chapter, we describe how to modify an existing configuration file or create a new one, using the **config** tool.

Choosing a Tool

In the Apollo model, software product installation consists of copying software from an Authorized Area into an operational configuration on a node. The operational configuration is determined by five files:

- A release index that describes all the possible configurations supported by the product. The release index file ships with the product and cannot be modified at your site.
- An optional active override file that constrains the set of possible product configurations installable from an Authorized Area to a subset of those described in the release index. It is usually created when a product is loaded into an Authorized Area.
- A configuration file that further narrows the possible product configurations by establishing a single configuration that can be installed on a target node. The configuration file is usually created at your site.

- An optional **preserve.list** file you can create on the target node that lists the names of files not to be overwritten when a product is installed on a node.

- An optional **excludes.list** file you can create that lists the names of files not to be installed from an Authorized Area.

The basic tool for establishing a product configuration is **config**. Product configurations are defined by configuration files that are used by **install** or **install++** to direct the installation of software on the target node. Only **config** can create a configuration file. It is the tool of choice when you want to create a set of standard configurations to guide installations at your site. The **config** tool does not install software; it is only an interactive tool for creating configuration files.

We also provide the **install++** tool, which invokes **config** in its configuration phase and then passes the resulting configuration file to the **install** tool in its installation phase. By default, the configuration file that **install++** creates is deleted when **install++** completes its installation phase. You may find **install++** more expedient when you are installing a unique configuration to a node, and you are sure that you won't be needing the configuration you establish for that node again.

Although you can run **install++** noninteractively, it is primarily intended for one-shot interactive installations of custom software configurations. The **install** tool, on the other hand, is intended for noninteractive, batch mode, installations of a single product configuration to many nodes on a network. However, neither of the installation tools has any functional advantage over the other; **install++** works by invoking **config** and passing the resulting configuration file to **install**.

Starting the Configuration Process

Start the configuration process by running either the **config** or the **install++** program. Once you've entered the interactive phase of either program, the user interface is the same.

Running the config Tool

The **config** tool always runs interactively. To run the tool, use the following command line:

AA/install/tools/**config** -s *AA* -c *configuration_file*

where:

AA is the pathname of the Authorized Area containing the products you want to constrain.

configuration_file
is the pathname of the configuration file that you want to modify or create.

Running the install++ Tool Interactively

You can run the configuration phase of the **install++** tool interactively with or without specifying the name of a preexisting configuration file on the command line. Specify the name of a configuration file if you want to configure the target node similarly, but not identically, to the way it would be configured using that file, or if you want to create a new configuration file with that name. If you run **install++** without specifying a configuration file, the program creates a temporary one which exists only for the duration of the installation.

Invoke the **install++** program with the following command line:

AA/install/tools/install++ -pvx
-s *AA* -c *configuration_file target* [*target* ...]

where:

AA is the pathname of the Authorized Area containing the products you want to configure.

configuration_file
is the pathname of the configuration file that you want to modify or create.

target [*target* ...]
is a list of targets to install to during the installation phase.

A target can be the pathname of any directory. Typically, however, the pathname of a node entry directory, **//my_node** for example, or the pathname of a mount point for a mounted volume is specified.

The **install++** tool begins its configuration phase by invoking **config**. Unlike **config**, the **install++** tool actually installs the software configuration by invoking **install** and passing it the configuration file created during the configuration phase. The configuration phase is interactive; the installation phase is not.

Interacting with the Configuration Program

Once you have invoked **config** or **install++** according to the above directions, the program lists all products available from the Authorized Area. There may be more than one version available for certain products. Then the prompt changes to `CONFIG>`. Figure 6–1 illustrates the start of an interactive **install++** session.

```
$ install++ -s //sr10 -c my_config  //my_node

Scanning Authorized Area in //sr10

       RAI Config Tool  --- Version 1.0  06/15/88

CONFIG> s a
The following is a list of products/versions
available for
selection
1. asm     1.0
2. cc      3.0
3. d3m     6.0
4. ftn    10.0
5. gpio   10.0
6. os     10.0
7. pas     1.0
8. spe     2.0

Type 'help' for command information

CONFIG>
```

Figure 6-1. Beginning an Interactive Configuration Session

General Guidelines

Below are some guidelines to bear in mind while configuring software:

- The **config** tool is interactive and its commands can be abbreviated to the point of uniqueness.
- You can exit from the configuration process at any time. If you enter **exit** at the CONFIG> prompt, the tool saves all configuration changes made in this session. At installation time, any configuration questions you did not answer are set to default values shipped with the software. If you enter **abort**, the tool exits without saving any of the changes you have made.

Using the Configuration Commands

Follow the steps below to configure products for the target node:

1. At the CONFIG> prompt, enter **show selection** or **s s**. The program lists all products currently selected for installation on the target node.

   To redisplay the list of available products shown when the program is first invoked, enter **show available** or **s a** at the prompt.

2. If the list produced by the **show selections** command is correct (that is, if it displays the products you want to install on the target node), begin configuring the products selected, as described in Steps 4 through 8. Otherwise, use the **select, deselect, update,** or **update all** commands to modify the list of selected products. At the CONFIG> prompt, enter the commands as shown in the following descriptions.

   select *available_product* [*available_version*]
   adds an available product you wish to install to the list of selected products. If you choose a product that is incompatible with a currently selected product, you will be warned when you try to install the configuration. The **config** tool does not, however, warn you at this time.

   deselect *selected_product* [*selected_version*]
   deletes a selected product you do not wish to install from the list of selected products. If you deselect a product that is necessary for the operation of a currently selected product, you will be warned when you attempt to install the configuration. The **config** tool does not, however, warn you at this time.

   NOTE: Deselecting a product currently installed on a node does not cause the installation program to "de-install" that product by removing it from the node.

   update *selected_product*
   replaces any and all of the current versions of *selected_product* with the most recent version available, if different.

update all
replaces any and all of the currently selected versions of every product with the most recent version available, if different.

NOTE: If you update one or more products, the answers to previously answered questions are retained if the latest version retains those questions.

select all
adds to the list of selected products the latest version of every product in the Authorized Area, and deletes from the list any earlier versions of those products.

where:

available_product
is the name of a product as it appears on the list of available products.

available_version
is the version of a product as it appears on the list of available products.

selected_product
is the name of a product as it appears on the list of selected products.

selected_version
is the version of a product as it appears on the list of selected products.

3. After several modifications to the list of selected products, you may want to enter the **show selections** command again to verify the result. You may also want to use the **show queries** *product* command to see what configuration options are available for a given product. Note that for the operating system product (designated **os**) there are a very large number of questions, reflecting an equally large number of configuration options.

4. When you are satisfied with the list of selected products, configure the products for which you want nondefault configurations. If you are installing Domain/OS operating system software, configure the operating system product first.

For each product you want to configure, enter either the **configure** or the **reconfigure** command at the CONFIG> prompt:

configure *selected_product* [*selected_version*]

or

reconfigure *selected_product* [*selected_version*]

The **configure** command presents you with questions not answered for the product in previous configuration sessions. This command is especially useful when you exit in the middle of a configuration session and then return to finish configuring a product.

The **reconfigure** command presents you with all questions for the product being configured. Use this command to create a new configuration file by modifying an old one, or to change the answers you supplied in a previous configuration session.

5. For some products, the program asks if you want to see detailed explanations of the configuration questions for the product. We recommend that you answer **yes** to this question, at least the first time you configure a given product.

6. The program displays questions specific to the product you are configuring (see Figure 6–2). Common types of questions ask whether you want to install an optional part of the product (for example, help files, manual pages, or files in **/domain_examples**), and whether you want to install the product by copying the files or by linking to them on another node. All questions have a default answer, indicated by a **(D)**, supplied by Apollo. You can select the default by pressing <RETURN> at the ==> prompt.

```
/SYS/HELP
 The /sys/help directory contains help files for the
 Aegis environment.If this directory exists and is NOT a
 link on the target, you may want to install the help
 files for ftn.  If /sys/help IS a link from the target
 to another node, then you must install ftn on
 that node if you want the latest version of these files
 to be available on the target.

Do you want a local copy of ftn help files (/sys/help),
a link to another node or neither?
: [ copy(D) link none ]
==>

/DOMAIN_EXAMPLES
 The /domain_examples directory contains online
 examples.If this directory exists and is NOT a link on
 the target, you may want to install /domain_examples/ftn.
 If /domain_examples IS a link from the target to another
 node, then you must install ftn on that node if you
 want the latest version of these examples to be available
 on the target.

Do you want a local copy of the online examples
(/domain_examples) for ftn, a link to another node
or neither?
: [ copy(D) link none ]

==>
```

Figure 6-2. Sample Product Questions

NOTE: If you are configuring the Domain/OS product, the program asks you to select the environments to be installed; the number available to you depends on the contents of the Authorized Area.

Answer the questions for the product you are configuring. When you have finished answering all the questions for a given product, the CONFIG> prompt returns. If you wish to return to the CONFIG> prompt before answering all the questions for a product, enter STOP at the ==> prompt. You may resume configuring this product by entering **configure** at the CONFIG> prompt. If you do, answers you've made to questions are retained. After you exit **config**, the program assigns configuration values according to your answers plus the default answers to any questions you have not answered.

7. Once you've configured the product, you can indicate whether you want the installation program to make certain checks before it actually installs the product on the target nodes. Enter the following command at the CONFIG> prompt:

   install checking *selected_product* [*selected_version*] *value*

   where:

   selected_product
   is the name of a product as it appears on the list of selected products.

   selected_version
   is the version of a product as it appears on the list of selected products.

   value
   is one of the following:

   version — Install only if the object already on the target differs from the version of the object in the Authorized Area. This is the default install checking and the type we recommend.

   exist — Install only if named object already exists on target. This option can be used to avoid replacing objects that were deleted deliberately to save disk space. Choosing **install checking exist** does not imply **install checking version**.

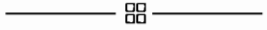
   none — No install checking. This option forces installation of all objects in the configuration, ignoring the software already on the target node. It is not normally necessary or recommended.

8. Repeat Steps 4 through 7 as appropriate, for each product whose configuration you wish to modify. Remember that you can end the configuration session and save your changes after finishing any product by entering **exit**, or end the session without saving changes by entering **quit** at the CONFIG> prompt.

9. When you've completed configuring all the products on the list of selected products, enter **exit**:

```
CONFIG> exit
```

The **config** or **install++** program terminates the configuration session. If you began the session by running **install++**, the program now begins installing software on the target nodes. If you began the session by running **config**, you are returned to the shell command line. In either case, refer to Chapter 7 for information on continuing the installation process.



Chapter 7

Installing Products

This chapter describes how to use the **install** and the **install++** tools to install products. To install products according to the installation procedures described in this chapter, an Authorized Area containing the products you want to install must exist on your network.

The install and install++ Tools

In the Apollo model, software product installation consists of creating an operational configuration of software on a node from one or more products loaded into an Authorized Area. The operational configuration is determined by five files:

- A release index that describes all the possible configurations supported by the product. The release index file ships with the product and cannot be modified at your site.
- An optional active override file that constrains the set of possible product configurations installable from an Authorized Area to a subset of those described in the release index. It is usually created when a product is loaded into an Authorized Area.
- A configuration file that further narrows the possible product configurations by establishing a single configuration that can be installed on a target node. The configuration file is usually created at your site.

- An optional **preserve.list** file you can create on a target node that lists the names of files not to be overwritten when a product is installed on a node.

- An optional **excludes.list** file you can create that lists the names of files not to be installed from an Authorized Area.

You can establish a product configuration before attempting to install software products, or you can postpone the creation of a configuration file until installation time. If you want to impose a consistent product configuration on every node at your site, you probably want to establish a single product configuration by using the **config** tool directly, as described in Chapter 6, and then using the **install** tool to install products. If you do not care to save a product configuration for future use, you may find the **install++** tool more expedient.

In general, the **install++** tool is intended for one-shot interactive installations of custom software configurations, whereas the **install** tool is intended for noninteractive, batch mode, installations of a single product configuration to many nodes on a network. However, neither of the installation tools has any functional advantage over the other; **install++** works by invoking **config** and passing the resulting configuration file to **install**.

The installation tools accept many options on the command line to support different installation strategies. This chapter describes only some of the possible ways to use **install** and **install++**. For more information on their capabilities, consult Appendix C or the online help in the **install/help** directory of an Authorized Area.

Preparation

This chapter is intended primarily for installing optional products from an Authorized Area to another node, and for installing an updated version of Domain/OS (SR10.x) on a node already running Domain/OS.

If no nodes on your network are running Domain/OS (SR10.x), first use the procedures described in Chapter 2 to create an Authorized Area and install Domain/OS on one node in the network. If you want to install Domain/OS from this Authorized Area to another pre-SR10 node, use the procedures in Chapter 3. If you want to install an optional product or an update of Domain/OS, and haven't yet loaded the product from the distribution media into an Authorized Area, first load the product using the procedures in Chapter 5.

Things to Check before Installation

Before you begin installation, check the following conditions:

- When you are installing Domain/OS or large optional products, the node on which you invoke the install tool should contain at least 10 MB of free disk space. This is an approximation; the exact amount of free space required depends on the product configuration you specify and the size of the baseline files. The **/com/lvolfs** command and the UNIX **df** command show the amount of free disk space.

- Make sure you are not running the **lprotect** program on the target. The installation program will not install to a target running **lprotect.**

- If you are installing an optional product, make sure each target is running the version of Domain/OS required by the product. Consult the product's release documentation to find out the product's operating system requirements. Use the **/com/bldt** or **/usr/apollo/bin/bldt** program to learn which version of Domain/OS the target is currently running. Domain/OS, and any optional software that requires Domain/OS to operate, cannot be installed on a target running a pre-SR10 version of the operating system.

Things to Know Before Installation

Before you can begin installation, you must know the following information:

- You must know the node names of the targets of the installation (see Figure 7-1). Software can only be installed on

a node with a disk. If the node on which you are installing software is booted from its own disk, the "target" is that node's name (for example, **//color**). If the node is booted diskless from and mounted on a partner node, the target is the pathname of the original node's disk as mounted on the partner's file system (for example, **//ergo/color**).

- You must know the pathname of your site's Authorized Area. The Authorized Area must be located on a Domain/OS node or file server.

- You must know the pathnames of the configuration files you want to use, if any. Apollo distributes standard configuration files with each product release. You must have an existing configuration file to use the **install** tool. If you do not wish to use an existing configuration file, you can use the **install++** tool without specifying a configuration file on the command line.

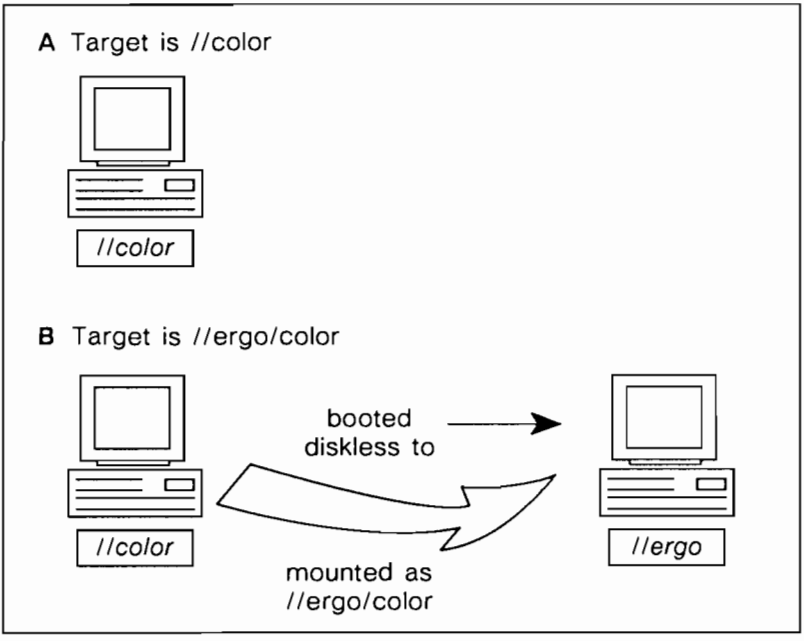


Figure 7–1. Installation Targets

You should also read the software release documentation, located online in the *AA*/**install/doc/***company* directory (for example, in **//pasta/install/doc/apollo**, if **//pasta** is an Authorized Area), for each product you intend to install. Pay particular attention to any product-specific installation issues and dependencies discussed in Chapter 2 of each product's release notes.

> **NOTE:** We generally recommend the use of the options **-p** (purge baseline files), **-v** (verbose output), and **-x** (continue on error) with every installation. Therefore, those options are combined into the argument **-pvx** shown in all the command lines to follow.

When an installation program completes, go on to the section "Completing the Installation" later in this chapter for information on error checking and rebooting your node.

Installing Interactively

You can run the **install++** tool interactively with or without specifying the name of a pre-existing configuration file on the command line. Specify the name of a configuration file if you want to configure the target node similarly, but not identically, to the way it would be configured by using that file, or if you want to create a new configuration file with that name. If you run **install++** without specifying a configuration file, the program creates a temporary one which exists only for the duration of the installation.

Invoke the **install++** program by entering

AA/**install/tools/install++** **-s** *AA* [**-c** *configuration_file*]
target [*target* ...]

where:

AA
: is the pathname of the Authorized Area containing the products you want to configure.

configuration_file
: is the pathname of the configuration file that you want to modify or create.

target [*target* ...]
is a list of targets to install to during the installation phase. For a target, you usually supply the name of a node entry directory, **//this_node** for example, or the pathname of a mount point for a mounted volume, **//this_node/that_node** for example. However, you can supply the pathname of any directory, and the product will be installed there.

The **install++** tool begins its configuration phase by invoking **config**. Unlike **config**, **install++** actually installs the software configuration by invoking **install** and passing it the configuration file created during the configuration phase. The configuration phase is interactive; the installation phase is not. See Chapter 6 for more information on configuring products for installation.

Other than invoking **config** to create a temporary configuration file, the **install++** tool functions identically to the **install** tool, and both tools share the same set of command-line options. The rest of this chapter uses **install** in its examples. You may, in general, substitute **install++** for **install** where it appears in the following procedures. The only resulting difference is that, if you are installing via default configuration files that contain unanswered questions, those unanswered questions will be presented to you during the configuration phase of **install++** unless you specify the **-d** option on the command line. The **install** tools will supply default answers to unanswered configuration questions, as will **install++ -d**.

Simple Batch Installation

You can use the **install** or the **install++** tool to install system software and optional products according to an existing configuration file. It may be a standard configuration file supplied by Apollo or a system administrator, or it may be one you created yourself by using the procedure described in Chapter 6.

Using a Single Configuration File

Use the following **install** command line to install a single, previously created configuration on one or more target nodes:

AA/install/tools/install **-pvx** -s *AA*
-c *configuration_file target* [*target* ...]

where:

AA
is the pathname of an Authorized Area containing the products you want to install.

configuration_file
is the pathname of the configuration file describing the product configuration you want to install.

target [*target* ...]
is a list of installation targets. For a target, you usually supply the name of a node entry directory, **//this_node** for example, or the pathname of a mount point for a mounted volume, **//this_node/that_node** for example. However, you can supply the pathname of any directory.

You can install the default configuration for a single product by using the **-c** option. Locate the default configuration file for the product in the *AA*/**install/templates**/*company*/*product* directory. Each directory contains a default configuration file for the associated product.

For example, if your Authorized Area is at **//admin**/AA, and you want to install version 10.2 of Domain/OS, the default configuration file is

//admin/AA/install/templates/apollo/os.v.10.2/cf.os

To install this default configuration to the disked node **//that_node** and the disk volume mounted at **//this_node/mount_point**, you could use the following command line:

```
//admin/AA/install/tools/install -pvx -s //admin/AA
        -c  //admin/AA/install/templates/apollo/
        os.v.10.2/cf.os
        //that_node //this_node/mount_point
```

Using Multiple Configuration Files

The **install** and **install++** tools enable you to use more than one configuration file during a single install. You can list each configuration file in the same command line, preceding each file name with the **-c** option:

AA/install/tools/install -pvx -s *AA* -c *configuration_file_1*
-c *configuration_file_2* -c *configuration_file_n*
target [*target* ...]

where:

AA
is the pathname of an Authorized Area containing the products you want to install.

configuration_file_{1, 2, n}
are the pathnames of different configuration files.

target [*target* ...]
is a list of installation targets. For a target, you usually supply the name of a node entry directory, **//this_node** for example, or the pathname of a mount point for a mounted volume, **//this_node/that_node** for example. However, you can supply the pathname of any directory. All of the product configurations specified in the different configuration files are installed on all of the targets.

install++ (but not **install**) also enables you to create a file that contains a list of the configuration file pathnames (one per line). You then supply the name of the list file as input to **install++**, preceded by the **-C** option:

AA/install/tools/install++ -pvx -d -s *AA*
-C *configuration_file_list_file* *target* [*target* ...]

For example, suppose you want to install Domain/OS, the Domain Pascal compiler, and the Domain FORTRAN compiler, using the Apollo-supplied default configuration for each product. (Each directory of the form *AA*/**install/templates**/*company/product* contains a default configuration file for the associated product.) First, create a file listing each default configuration file. If your Authorized Area is named //AA, the list file might look something like this:

```
//AA/install/templates/apollo/os.v.10.0/cf.os
//AA/install/templates/apollo/pas.v.8.04/cf.pas
//AA/install/templates/apollo/ftn.v.10.0/cf.ftn
```

Then install the products by supplying the name of the list file to **install++**, as shown in the previous command line.

Once you invoke the installation program, go to the section "Completing the Installation" later in this chapter for information on error checking and rebooting your node.

Installing to Multiple Nodes

As with configuration files, **install** and **install++** enable you to specify more than one installation target for a single invocation of the install tool. As shown in previous examples, you can list multiple targets on the command line. Or you can use the **-n** option to specify a file containing a list of targets. The target list file should specify one target per line.

Push Installations

By default, the **install** tool runs exclusively on the node it was invoked from and installs the specified configuration to each target in succession, one after another. This is known as a "push" installation; the product is pushed from the Authorized Area onto the target nodes from the node on which **install** is running. See Figure 7–2.

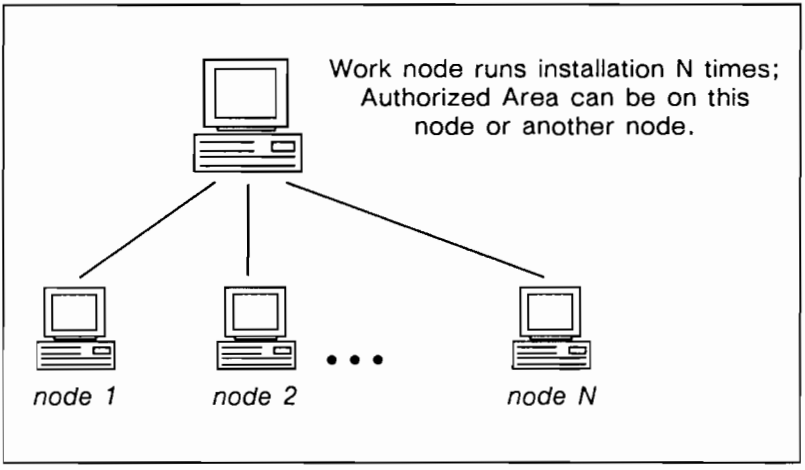


Figure 7–2. Push Installation

Pull Installations

The **install** or **install++** tool can also install products on target nodes by executing remotely on those nodes under certain

conditions (see Figure 7–3). You invoke the **install** or **install++** program on a work node, and the program executes remotely on each of the individual target nodes. This is known as a "pull" installation because the products are pulled from the Authorized Area by the target node. A pull installation distributes the workload and results in faster multiple installations. For pull installations to be effective, the following conditions must be met.

- The remote target nodes must be running Domain/OS.
- The remote target nodes must be running the Server Process Manager (**spm**) program.
- The product configuration must be fully specified before **install** or **install++** is invoked. You cannot configure products interactively when launching pull installations with **install++**.

When these conditions are not met for a target, the installation program executes locally on the work node (the node on which you invoke **install** or **install++**) for that target, just as if the target node were the target of a push installation. A pull installation is still performed on those targets that meet the conditions.

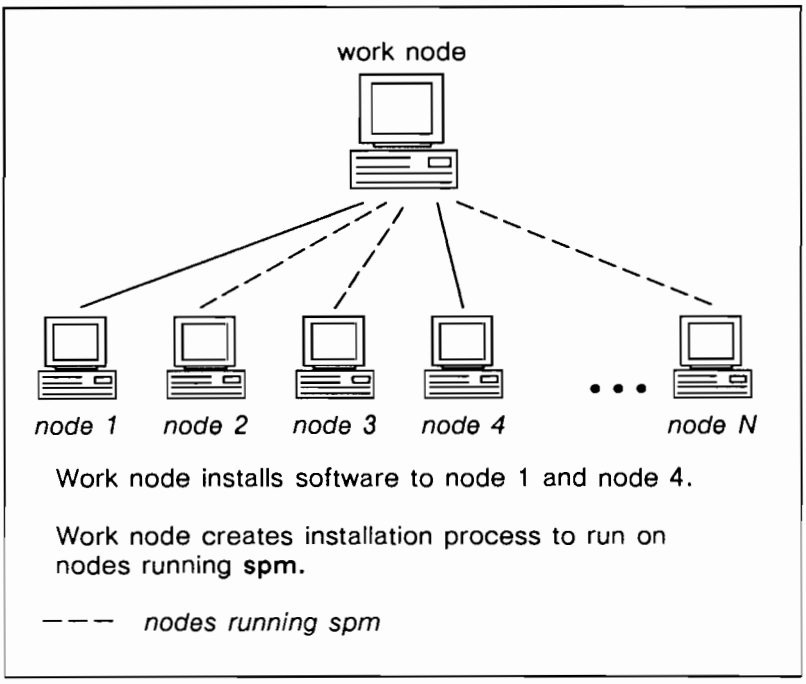


Figure 7–3. Pull Installation

To perform a pull installation on multiple targets, you must specify the **-r** option on the **install** command line. You may specify the product configuration with the **-c** option, and the target nodes with or without the **-n** option. For example, to install the product configuration defined by *configuration_file* to multiple targets listed in the file *target_list_file*, you can invoke **install** like this:

AA/install/tools/install -pvx **-r** -s *AA* -c *configuration_file*
-n *target_list_file*

where:

AA
is the pathname of an Authorized Area containing the products to be installed.

configuration_file
is the pathname of the configuration file describing the

product configuration you want to install on each of the targets listed in *target_list_file*.

target_list_file
is the pathname of a file containing a list of installation targets.

During pull installations, the **install** tool creates an installation record file for each remote target. The record file records any error or warning messages from the target and other pertinent installation information. Each installation record file has the name *AA/node_name.x*, where *x* is an integer value.

Other Installation Features

In addition to allowing multiple configuration files and parallel installation to multiple target nodes, the **install** tool has other features you may find useful. Consult Appendix C or the online help files in the **install/help** directory of an Authorized Area for an exhaustive list of **install** and **install++** options. The following sections discuss only a few of the most interesting ones.

Linking to the Authorized Area

If the Authorized Area you are installing from is on the target node, you can use the **-l** option to install the product configuration on the node via hard links to the Authorized Area. Use of the **-l** option to **install** (or **install++**) does not restrict future modifications to the Authorized Area or installations to the node. The links created are hard links. Future installations will not affect the contents of the Authorized Area, nor will modifications of the Authorized Area affect the operational hardware installed on the node.

We recommend you use the **-l** option when installing to the node containing the Authorized Area, since **install** runs faster and much disk space is saved because most product components do not have to be copied during the installation.

Ignoring Object Customization

By default, the **install** (and **install++**) tool assumes that, if the software it finds on a target node differs from what was previously

installed, the difference is due to intentional customization by a user of the node since the previous installation completed. Therefore, if a user of the node has deleted some of the objects installed on the node, **install** will not recreate those objects unless told to do so. The default behavior of **install** is generally what you want; however, you can force **install** to install the product in the exact configuration specified on the command line by using the **-m** option.

The **-m** option is the easiest way to get a "clean" configuration on a node whose software you suspect may have been modified or not quite properly installed.

Without the **-m** option, the **install** tool does not install an object as a local copy if a previously installed version of the object is a link. Likewise, **install** does not install an object as a link if the existing version is a local copy. With the **-m** option, you can override this action for *files*—a file will be installed as a local copy or link regardless of whether the existing file is a local copy or link. However, the **-m** option does not permit the change of a *directory* from a local copy to a link, or from a link to a local copy.

Help for Unattended Installations

Because **install** is not interactive, you can run it unattended. You can also run **install++** noninteractively, and hence unattended, depending on the command line options you supply. If you run **install++** interactively, you can let the installation phase run unattended once you complete the interactive configuration phase. Installation of large product configurations, or even small configurations to multiple nodes, can take hours, so you may want to run an installation overnight or over a weekend. Both **install** and **install++** accept two options, **-i** and **-o**, that are primarily useful during long, unattended installations.

By default, **install** checks to make sure there is enough free disk space on a target node for the product configurations it is about to install, and warns the user if there isn't sufficient disk space. The **-i** option turns off this checking. It is most useful when you are replacing an old version of a product with a newer version, since most of the space required by the new objects is available after deletion of the old ones. We recommend that you use this option for unattended batch installations—during an overnight installation of a new version of Domain/OS, for example.

Without the **-o** option, **install** installs only once to each target specified on the command line, even if target names are repeated.

This is generally what you want, but occasionally you might want to force **install** to attempt each target multiple times. If you run **install** with the **-o** option, **install** attempts to install the product configuration to each named target as many times as that target is specified.

In an unattended installation on a large and heavily loaded network, you can improve your chances of success by naming each target twice and specifying **-o** on the **install** command line.

Completing the Installation

When the **install** (or **install++**) tool begins to install the product configuration you have specified, it displays various informational messages about the configuration and installation process (see Figure 7–4). Follow the directions in this section to complete the installation on each target node.

```
RAI install++ 0.21 1988/04/01
            .
            .
Authorized area is on //sim4
The selected switch settings are:
    Existence of files in the AA will not be checked
    sys_sr10 is the configuration file for all target nodes
    Continue on error has been selected
            .
            .
Installation is for:
    //sim4
            .
            .
Installing //sim4
Using baseline file baseline.00000001 for node //sim4
Checking status of baseline file entry ri.apollo.os.v.10.0
    Still checking at Wed Apr 27 11:29:32 1988
Checking status of configure file entry ri.apollo.os.v.10.0
Computing installable set for //sim4
The installation requires 29665 blocks of free disk space
You have 34557 blocks available. Do you wish to continue
(y,n)?
```

```
yes
```

Figure 7–4. Messages During Installation

Checking the Installation Transcript

When the installation is complete, you see one of the following messages:

```
RAI install has successfully completed.
                or
RAI install has completed with errors.
```

In either case, we recommend that you check the transcript produced by **install** (or **install++**). The **install** tool prefixes all warnings with the label

WARNING:

and all errors with the label

ERROR:

If the transcript of the installation session contains errors, you should consider running **install** again to replace any objects that couldn't be installed due to transient network problems. Another run of the **install** tool should be much faster than the first because, by default, **install** copies only those objects that were not successfully installed during the first run.

> NOTE: The **install++** tool may prompt you to shut down, reset, and restart the target node. Do so only after you have checked the transcript for errors.

Troubleshooting

When you locate an error or warning, correct any problems that may exist. Then, if necessary, rerun **install** (or **install++**). Following are some of the more common installation error and warning messages and suggestions for dealing with them.

`ERROR:Could not copy file ... to ...`
For some reason, **install** could not copy the named file. The **install** tool must run with the user ID **root**. If the permissions on **install** are intact (that is, it is set-UID **root**), then the problem was probably temporary; try rerunning install. If objects could not be installed due to missing directories on the target or other such problems, you may want to rerun **install** with the **-m** option. The **-m** option should cause the files not installed the first time to be copied to the target.

`ERROR:... must be a local copy not a soft link`
An object must be installed as a local copy, but the target directory for the object, or the current object at the target pathname, is a link. The **install** tool changes the link to a copy only if the **-m** option is used and the object being installed is not a directory, or if the product containing the object is configured with **install checking none**. Delete the offending link, and reinstall.

`ERROR:Cannot install soft link ...; already is a local copy`
The product configuration called for creating a symbolic link on the target, but there is already a local copy at the

target pathname for the object. If you want a link there, you must delete the local object and rerun **install**. If the object on the target is a file, running **install** with the **-m** option will replace that file with a link as directed.

`WARNING:... would install through a link - item is ignored`
The product configuration called for creating an object at a pathname that resolves through a symbolic link on the target node, so the object was not installed. If the object must be installed, delete the link and reinstall.

`WARNING:Could not delete ...`
The product configuration called for deleting an object on the target, but **install** was unable to delete it. Either **install** is running with the wrong permissions (that is, it is not set-UID root), or a temporary problem prevented the action. Check that **install** is set-UID root (using the **lsacl** command), and reinstall.

`ERROR:... not found in authorized area`
A product in the configuration to be installed does not exist in the Authorized Area specified on the command line with the **-s** option. Make sure you have specified the Authorized Area correctly, and that the specified configuration files were created from the specified Authorized Area.

`ERROR:Could not delete existing type ... for ...`
The **install** tool could not delete an installed type from a target node as called for by the release index for the named product. Try deleting the type yourself with the **dlty** command or reinstall the product.

`ERROR:Could not find UID for ...`
The named object is cataloged, but it could not be located. The Authorized Area node or the target node may need salvaging. Run **salvol** on the offending disk volume and reinstall.

`ERROR:Could not get node UID for ...`
An installation target could not be located. Make sure the named node is cataloged. Try cataloging it with the **ctnode** command and reinstall.

`ERROR:Could not hard link from ... to ...`
The specified product configuration requires a hard link

that cannot be created, so the installation was aborted. Try rerunning **install** without the **-l** option.

`ERROR:Original ACLs could not be placed on ...`

or

`ERROR:Could not preserve original ACLs for ...`
While trying to set the permissions on a new object to the original permissions of the old object on the target, some error occurred. The original permissions on the object were lost.

`ERROR:Could not recover original copy of ...`
The **install** tool encountered an error while attempting to preserve the original permissions on a file. To preserve permissions, **install** must rename the file, copy in the new file, copy the old permissions to the new object, and delete the old copy. Somewhere in this process, an unrecoverable error occurred. The original file is in **rai_acl_temp.*** in the **/install** directory on the target. Recopy this file to its original name and location and try the installation again. Alternatively, use the **edacl** or **chmod** command to change the permissions on the new object to match those on the old object.

`WARNING:Could not change type of object ...`
The product configuration called for changing the type of an object on the target node, but **install** could not change the type. Check the permissions on the **install** tool and re-install. If that doesn't work, you may have to run **salvol** on the target or remove the offending object with **rm** or **dlt**.

`WARNING:Could not create baseline file ...`
The **install** tool could not write a baseline file on the target node for some reason. Future installations will have to make do without one.

`ERROR:Cannot access authorized area on ...`
The **install** tool could not access an Authorized Area specified on the command line. Check to make sure the pathname after the **-s** option on the **install** command line is correct.

`ERROR:Could not find configure file: ...`
The **install** tool could not find a configuration file specified on the command line. Check that the configuration file pathnames are correct.

`WARNING:Target path file ... could not be found`
The **install** tool could not find a target list file specified on the command line with the **-n** option. Check that the target list file pathnames are complete and correct.

`WARNING:Target path file ... is not readable ...`
The **install** tool could not read a target list file specified on the command line with the **-n** option. Check that the target list file is readable.

`WARNING:Invalid configuration for remote install`
You invoked **install++** with the **-r** option and options that called for interactive configuration, so the **-r** option was ignored.

`WARNING:Remote process call failed ...`
The **install** tool could not start a remote process on a target node, so it ran the program locally.

`ERROR:Installation of...to...has been aborted.`
`ISP of target node and product not the same.`
The configuration included a product whose ISP type differed from that of the target node. By default, the **install** tool will not allow the installation of a product whose ISP type is incompatible with the target, because such a configuration may prevent the installed software from running and, in some cases, may even prevent the target from booting. You can use the **-h** option to **install** to override the default behavior.

Once you have resolved any errors in the installation and rerun it if necessary, check the transcript to see if it contains any messages telling you to shut down your node, reset, and reboot. These will normally appear if you have installed system software or software that makes changes to system libraries.

If you installed optional software products only and you were not prompted to shut down and reboot your node, then you are ready to run the software you have installed. Go on to the section "After the Installation" at the end of this chapter.

Rebooting the Target Node

If you installed new system software on a node already running Domain/OS, you must shut down and reboot the target node, using this procedure:

1. Shut down the target node by entering one of the following commands.

   If the target node is a workstation running the Display Manager (DM), use the the DM **shut** command:

   ```
   Command: shut
   ```

   If the target is a workstation that is not running the DM, log on to the target as **root** and issue the UNIX **shutdown** command:

   ```
   /etc/shutdown -y -g0 -i5          (SysV environment)

   /etc/shutdown -h now              (BSD environment)
   ```

   If the target is a storage module attached to a DSP, issue the **shut** command to the Server Process Manager (**spm**), or issue the **/etc/shutspm** command via **crp** to the target.

   Wait for the message

   ```
   SHUTDOWN SUCCESSFUL
   ```

   and for the Mnemonic Debugger (MD) prompt to appear. The prompt depends on the node firmware, but it will end in a >.

2. If you installed software on a DN460 or DN660, reload the microcode with the following commands:

   ```
   >gb
   %ua
   ```

   NOTE: Though the prompt should return to > after the **ua** command, sometimes the **%** prompt persists. If this occurs, simply continue with the next step.

3. Enter a reset command, followed by a carriage return at the next prompt. The reset command for Series 10000 workstations is **RE W**. For all other workstations, the command is simply **RE**. For example,

   ```
   >RE
   ><RETURN>
   MD3X    Rev. 6.0, 1986/03/05, 16:52:12
   >
   ```

4. Boot the node with the command

   >EX DOMAIN_OS

5. Log on to your node by using your own account name.

You are now ready to run the software you installed.

After the Installation

When you have completed a software installation, the target node contains a top-level **/install** directory containing the following directories:

- **/install/baseline**
  Contains the baseline file for this target.

- **/install/doc**
  Holds directories for each company whose products are installed on this target. Under this, **/install/doc/apollo** holds release notes for all Apollo products installed on this target.

It may also contain one or more of the following files. The existence of either file implies that there were problems with the installation.

- **/install/not_installed**
  Lists objects that could not be installed from the last installation. Repeating the same **install** (or **install++**) command causes these objects to be installed, provided the problem that prevented their installation has been fixed.

- **/install/rai_acl_temp.***number*
  An object whose initial permissions could not be preserved. The *number* is an arbitrary numeric value assigned by the installation program. The installation transcript pad shows the original pathnames for each of the **/install/rai_acl_temp.***number* files created during that installation.

You may want to create the file **/install/preserve.list**, containing the pathnames of all files on the node that you do not want changed by a subsequent installation. The pathnames in **/install/preserve.list** must be relative to /, the node entry directory.

NOTE: You may notice that you have a copy of the directory **/sys5.3/bin** even if you didn't install the SysV environment. This directory contains a minimal set of Bourne shell commands that support installation of applications created and distributed by our solution suppliers. We recommend that you not delete this directory. Delete it only if you do not use any applications designed by third-party vendors on your Apollo systems. See Appendix D for more information on this command set.

Chapter 8

Protecting Software

This chapter describes how to use the **inprot** (install protection) tool and discusses other protection issues related to software installation. **inprot** enables you to create a set of permissions for objects installed on a node running Domain/OS. **inprot** requires the existence of a Domain/OS version of the registry database somewhere on the network.

The protection model used by Domain/OS is significantly different from that used in previous versions of the operating system. For information on the changes, see *Making the Transition to SR10 Operating System Releases* and the *Domain System Software Release Notes*. For a complete description of the protection model, see the managing system software manuals.

Setting Permissions on Installed Objects

When you configure Domain/OS, you have the opportunity to choose an **open** or **closed** environment. In an open environment, the node owner has control over the objects on his or her node, including the system directories. In a closed environment, system administrators control the system software on all nodes; by default, users who are not system administrators cannot modify or delete critical system files and directories. A network in which all nodes are running closed environments is said to be a **secure network**.

NOTE: When you use **minst** in novice mode to install Domain/OS, the resulting environment is closed.

If you choose to install operating system software in an open environment, we recommend that you protect critical system directories and files after completing the installation. You do this by setting permissions so that the only nonprivileged (that is, neither **root** nor **locksmith**) account with access to and control over those objects is you, the node owner.

The **inprot** program allows you to set the permissions for a set of objects on a node running Domain/OS. The program accepts as input a file pathname and a target pathname. The file pathname points to a file containing templates for the types of protection desired for different types of files and directories. The target pathname can be a node name or the pathname of an Authorized Area (AA). The program modifies the permissions associated with the desired objects as they exist on the target.

NOTE: To use **inprot** to set permissions on system files, you must be **root** or **locksmith**, or set **inprot** to setuid root. If you set **inprot** to setuid root, be careful about who can run **inprot**, as it can be used to break security on your system.

Invoke **inprot** with the following command line:

AA/install/tools/inprot [-a *template_file*] [-t *target*]

where:

template_file
is the pathname of the template file to be used by the tool. For a description of the required format for this file, see the next section.

target
is the pathname (usually the node name) of the target, for example **//color**. The target must be a Domain/OS node.

If you omit the template file pathname, the tool looks on the target for a file with the pathname **/install/acl_list**. If you omit the target pathname, the tool assumes the current working directory. Run

inprot only from a Domain/OS node. If you run the tool on a node that is running a pre-SR10 version of the operating system, you will get a warning.

Template File Format

The **inprot** template file contains sets of entries, one entry on a line, that are used to establish the permissions for objects on a Domain/OS file system. There are two types of entry set: one, describing a file object, begins with a file object description entry; the other, describing a directory object, begins with a directory object description entry.

Each set of entries begins with an object description entry like the ones below, either for a file (begins with F) or for a directory (begins with D). Include required permission entries (P, G, O, W) only if you wish to change the current values for those entries. An entry may contain more than one extended permission (E) entry. P, G, O, W, and E entries may be listed in any order.

> NOTE: The entry description characters (F, D, P, G, O, W, E), the *required_rights* fields, and the *extended_rights* fields are *not* case sensitive. The *name* and *pers.grp.org* fields, however, must be lowercase.

File Description Entries:

F *file_path* **(object description entry)**
P *person_name required_rights* **[-setuid | -setuidoff]**
G *group_name required_rights* **[-setuid | -setuidoff]**
O *org_name required_rights* **[-setuid | -setuidoff]**
W *% required_rights*
E *pers.grp.org extended_rights*
E ...
.
.
.

Directory Description Entries:

D *directory_path* **[-files|-dirs] (object description entry)**
P *person_name required_rights*
G *group_name required_rights*
O *org_name required_rights*
W *% required_rights*
E *pers.grp.org extended_rights*
E ...
.
.
.

where:

F *file_path*
shows that this is a file object description entry pertaining to the file whose pathname relative to the target pathname is *file_path*. Domain/OS wildcards are supported.

D *directory_path*
shows that this is a directory object description entry pertaining to the directory whose pathname relative to the target pathname is *directory_path*. Domain/OS wildcards are supported.

-files
shows that this entry applies to all files created within the directory *directory_path*, not to the directory itself. If this flag is applied to a file description entry containing a Domain/OS wildcard expression, the protection on all the files in a tree can be set at one time.

-dirs
shows that this entry applies to all directories created within the directory *directory_path*, not to the directory itself. If this flag is applied to a directory description entry containing a Domain/OS wildcard expression, the protection on all the directories in a tree can be set at one time.

required_rights
is any valid combination of the letters p, w, r, x, k, j, i, and u. For information on the meanings of p, w, r, and x, see the managing system software manuals. The letter j corresponds to the [ignored] attribute. The letters i and u, used together, produce UNIX protection inheritance.

-setuid | -setuidoff
sets the "setuid" bit to "on" or to "off" for this file so that, when the file is executed, the process does or does not take on this person, group, or organization ID. Not valid for directory object entries.

P *person_name required_rights*
gives to all accounts with a "person" field of *person_name* the set of permissions described by *required_rights*, and sets ownership of the object to the "person" entry *person_name*. These permissions apply to the file or directory described by the last object description entry preceding this entry.

G *group_name required_rights*
gives to all accounts with a "group" (or "project") field of *group_name* the set of permissions described by *required_rights*, and sets the "group-ownership" of the object to the "group" entry *group_name*. These permissions apply to the file or directory described by the last object description entry preceding this entry.

O *org_name required_rights*
gives to all accounts with an "organization" field of *org_name* the set of permissions described by *required_rights*, and sets the "organization-ownership" of the object to the organization entry *org_name*. These permissions apply to the file or directory object description entry preceding this entry.

W *% required_rights*
gives to all other accounts the set of permissions described by *required_rights*. These permissions apply to the file or directory described by the last object description entry preceding this entry.

E *pers.grp.org*
gives to the account with a Subject Identifier (SID) of *pers.grp.org* the set of permissions described by *extended_rights*. These permissions apply to the file/directory object described by the last file/directory object description entry preceding this entry.

extended_rights
is any valid combination of the letters p, w, r, and x. For information on the meanings of these permissions, see the managing system software manuals.

The values *person_name*, *group_name*, and *org_name* must exist in the registry database in the lists of persons, groups, and organizations, respectively.

Template File Format Examples

Following are a number of examples of object entries for a template file, with descriptions of the level of protection the example implies.

Protection for open root (/) directory in a UNIX system:

```
D /
P  root  pwrx
G  staff wrx
O  none  -j
W  %  wrx
D / -dirs
P  root  -iu
G  staff -iu
O  none  -j
W  %  -u
D / -files
P  root  -iu
G  staff -iu
O  none  -j
W  %  -u
```

Protection for closed **/bin** directory in a UNIX system:

```
D /bin
P  root pwrx
G  staff  wrx
O  none  -j
W  % rx
D /bin -dirs
P  root pwrx
G  staff  wrx
O  none  -j
W  % rx
D /bin -files
P  root pwrx
G  staff  wrx
O  none  -j
W  % rx
```

Protection for closed system file:

```
F /bin/kill
P  root pwrx
G  staff  wrx
O  none  -j
W  %  rx
```

Protection for open system directory:

```
D /com
P  root pwrx
G  staff  pwrx
O  none  -j
W  %  pwrx
D /com -dirs
P  root pwrx
G  staff  pwrx
O  none  -j
W  %  pwrx
D /com -files
P  root pwrx
G  staff  pwrx
O  none  -j
W  %  pwrx
```

Protection for open system file:

```
F /com/invol
P  root pwrx
G  staff  pwrx
O  none  -j
W  %  pwrx
```

Protection for setuid -root programs:

```
F /etc/mount
P  root  pwrx -setuid
G  staff  -j
O  none  -j
W  %  rx
```

Protection for setuid -mail programs:

```
F /bin/rn
P  bin pwrx
G  mail rx -setuid
O  none  -j
W  %  rx
```

Results of Running inprot

For each required entry, if the existing permissions for the SID allow a change, the **inprot** program changes the rights field to the permissions given in the *required_rights* field. The program also sets the person, group, or organization component of the SID.

For an extended entry, if the SID name matches an existing extended entry, and if the existing rights allow for a change, then **inprot** sets the rights to the *extended_rights*. If the SID is not found, then this extended entry is added to the end of the extended SID list.

Protection Inheritance and Installation

There are a number of installation issues related to changes to our protection model at SR10. This section addresses those issues. It also notes restrictions on how the installs can be used vis-a-vis protection.

Inheritance Following invol

On Domain/OS nodes, you can choose the protection inheritance model for the / directory when you initialize the disk with **invol**. You may choose Aegis, BSD, or SysV protection inheritance; the default is Aegis. In addition, **invol** creates the directories /sys, /sys/node_data and /sys/node_data/system_logs. The inheritance on these directories is always according to the Aegis model, and the directories themselves are unprotected; that is, % (world) has all rights relative to the directories.

Installing an Open or a Closed Protection Model

When you install Domain/OS, you are asked to choose between an open and a closed protection model. The installation tools

implement open and closed protections for a node with UNIX inheritance in a way that is slightly different from the way the tools implement protections on a node with Aegis inheritance.

Another factor that affects the protection result is whether you install system software in **new** mode (following **invol**) or **update** mode (subsequent installation). If you specify **closed** and **new**, pw rights are removed from the world entry of the / and /sys directories and their initial default permissions.

The /sys directory always uses Aegis-style inheritance. Common base software relies on inheritance and does not specify the protection for each object (except setuid and protected subsystems). This ensures that the base software for a new installation will be open or closed by inheriting from / and from /sys.

For Aegis environment installations, protections are inherited from the containing directory for all newly created objects (except setuid and protected subsystems). Closed rights are inherited correctly because the initial permissions on / have had pw rights removed. Open rights are achieved by *not* removing pw rights and relying on inheritance. (An update installation with open protections does not add p or w rights on an Aegis installation, even though a new installation with closed protections removes them.)

For UNIX environment installations, the protection for each object is exactly specified (with no reliance on inheritance). Object protection is generally closed (world has no pw rights). If an open protection model is specified, the protection changes are made *after* the installation is completed; the installation program calls the **edacl** program to change permissions on all system trees by adding p rights for world. This allows a user to customize the protections on his or her node(s) after the software is installed. This changing of the protections on a UNIX installation to open is done for both new and update installations to ensure that newly created objects are accessible.

In general, update installations do not change the protections on preexisting objects, except for the addition of p rights in the case of an installation on an open UNIX environment.

When multiple environments (Aegis and one or more UNIX environments) are installed, specifying **closed** and **new** works as it does for other installations, removing pw rights from / and /sys and relying on inheritance or specified protections as appropriate for the

various objects on the node. Specifying **open** always causes the addition of p rights to world for all the system trees, including **/com**.

Restrictions

The **install++** program (or **config** and **install** programs) should *not* be used to change a node's protection model from open to closed or vice versa. When you install Domain/OS in update mode, we strongly recommend that you select the same type of protection (open or closed) that you selected when you installed the software in new mode (that is, after you **invol**'ed). The results are indeterminate if you mix open and closed protections on a node. You should use other tools, such as **inprot, edacl**, and **chacl**, to alter the protection of a node once the initial installation is done. Or you can **invol** and start again.

If you install Aegis and one or more UNIX environments, and select Aegis as the primary node environment, the installation tools still set up UNIX inheritance on all UNIX environment trees (for example, **/bin**). If you select a UNIX environment as the primary node environment, you get UNIX inheritance on all system trees except those in /sys.

List of setuid -root Files

The Aegis protection model provides node owners with control over node operations such as mounting and unmounting media devices, node shutdown and reboot, and process deletion. On the other hand, in a UNIX environment the above operations are restricted to privileged users and groups. At a site where the Aegis operating environment is not installed, system administrators may want to extend the ability of node owners to perform node administration tasks. We recommend that system administrators at such sites use the **chmod** program to modify the following programs to run as **root**:

- **/etc/mount**
- **/etc/umount**
- **/etc/shutdown**
- **/etc/reboot**

- **/bin/kill**

Protecting 'node_data

SR9.7 nodes cannot **crp** onto Domain/OS nodes if **'node_data** cannot be written to by world (%.%.%). This is one reason we do not more tightly protect **'node_data** in the installations. On a node that does not require SR9.7 **crp** access, an administrator may choose to tighten the protections on **'node_data**. Alternatively, an administrator may want to protect the **'node_data** paths by using the "keep" bit on the /sys, /sys/**node_data**, and especially the /sys/**node_data/etc** trees. We do not provide this capability within the installation procedure.

Appendix A

Running invol

This appendix explains how to use the **invol** program to initialize a disk and reset the size of the OS paging file when you are following one of the procedures in Chapter 2 or Chapter 3. If the target disk is partitioned into more than one logical volume, note that this description assumes you want to initialize the entire disk, not just one of the logical volumes.

For a comprehensive description of the **invol** command, see the online manual pages for **invol** and the command reference manual for your environment (Aegis, BSD, or SysV).

When you run **invol**, the menu shown in Figure A–1 appears. To initialize a disk and reset the size of the OS paging file when you are following one of the procedures in Chapter 2 or Chapter 3, perform the following steps:

```
invol (init_volume) - Offline(8), revision 10.2, date and time
Options are:
0              -  EXIT.
1  [-fnb5uom]  -  initialize virgin physical volume.
2  [-fnb5u]    -  add a logical volume.
3  [-fnb5]     -  re-initialize an existing logical volume.
   The following flags apply to options 1 thru 3, as indicated:
      f:  don't re-format disk    u: don't prompt user - use defaults
      o:  make sr9 format disk    n: make non-bootable volume
      b:  apply bsd unix acls     5: apply sys5 unix acls
      m:  build a multi-disk (e.g., striped) group
4           -  delete a logical volume.
5           -  list logical volumes.
6  [-e]     -  list badspots on disk or volume...-e: list in decimal.
7  [-f]     -  initialize physical badspot list.
8           -  create or modify an os paging file
9           -  add to existing badspot list.
10          -  display/change sector interleave factor.
11          -  remove from existing badspot list

Option:
```

Figure A–1. The invol Menu

1. Select option **1**, initialize virgin physical volume. Do *not* enter any of the available flags (–fnb5uom).

2. You are prompted for the type of disk:

   ```
   Select disk:
   [w=Winch|s=Storage mod|f=Floppy|q=Quit][ctrl#:][unit#]
   ```

   Enter **w** if the target is a Winchester disk.

   Enter **w***ctrl#*:*unit#* if the target is a Winchester disk on a Series 2500 workstation, where *ctrl#* is the disk controller number and *unit#* is the disk unit number. For example, **w5:0** denotes disk drive 0 on controller 5.

   Enter **s** if the target is a storage module. To specify a unit number on a storage module, append 0 or 1 to the letter. For example, **s1** denotes storage module unit 1.

3. If you are initializing a storage module, you are prompted for the drive type:

```
Possible Drive Types are:
1 - PN003863 (CDC)
2 - PN005100 (NEC)
3 - other (unknown)
Enter drive type:
```

Enter the number (1, 2, or 3) associated with the appropriate drive type.

4. You are asked to specify a name for the disk you are initializing:

```
Physical volume name:
```

Enter a character string of your choice, such as **apollo**.

> **NOTE:** If you are initializing a SCSI (Small Computer Systems Interface) disk on a Series 2500 workstation, **invol** formats the entire disk immediately after you enter the physical volume name. The message `Formatting` appears to indicate this. (On other workstation types, **invol** does not format the disk until after you specify logical volumes.) Do not stop the **invol** program while the format is in progress.

5. You are asked to select the method **invol** will use to verify the integrity of the disk:

```
Verification options are:
   1 - no verification
   2 - write all blocks on the volume
   3 - write and re-read all blocks on the volume

Enter verification option:
```

Note that option 1 is the fastest, but the least thorough. **invol** does not read or write to the disk, except to create the volume structure. The disk is not verified until it is mounted and read or written to by the operating system. Option 3 is the safest, but also the slowest; initializing a large disk can take a considerable period of time.

6. You are prompted for the expected average size of files that will reside on the disk:

```
Expected average file size, in kB (CR for default, 5
kB):
```

If you don't know the average file size, accept the default. Selecting a relatively accurate value for the average file size can save space on the disk, because the volume table of contents (a system table) is then allocated more efficiently. Note that the **salvol** program returns the average size of all files on a disk.

7. You are prompted for the size and name of each logical volume that you want formatted:

```
For each logical volume to be formatted, enter the
logical volume size (in kB), followed by the name, in
the form "size, name". Up to 10 volumes may be speci-
fied. Terminate input with a blank line. Specifying a
size of "all" will use all remaining blocks.

There are xxxxxx kB available.
volume 1:
```

To format the disk as a single logical volume, enter **all** (after "volume 1:"). This is the typical response.

To partition the volume into more than one logical volume, enter the desired size and name for the first logical volume. **invol** then prompts you for the name and size of the second logical volume and indicates how much space remains. After you enter the size and name of all logical volumes, enter a blank line to terminate input.

8. You are asked if you want to reuse the prerecorded badspot information shipped with the disk:

```
Use pre-recorded badspot info?
```

Enter **y**.

9. **invol** now initializes the target disk. Depending on the size of the disk and the verification mode you selected, this can take a significant amount of time. During the initializing, **invol** informs you of its progress. For example, if you selected verification option 3, the following messages are displayed:

```
Writing logical volume 1.
   Formatting ... % complete:
             20
              .
              .
              .
            100
   Writing all blocks ... % complete:
             20
              .
              .
              .
            100
   Reading all blocks ... % complete:
             20
              .
              .
              .
            100
   Initialization complete.
```

10. When the initialization completes, you are asked,

    ```
    Anything more to do?
    ```

    Since you must reset the size of the Domain/OS paging file, enter **y**. The **invol** menu (Figure A-1) reappears.

11. Select option 8, create or modify an OS paging file.

12. You are prompted for the type of disk:

    ```
    Select disk:
    [w=Winch|s=Storage mod|f=Floppy|q=Quit][ctrl#:][unit#]
    ```

    Enter **w** if the target is a Winchester disk.

    Enter **w***ctrl#*:*unit#* if the target is a Winchester disk on a Series 2500 workstation, where *ctrl#* is the disk controller number and *unit#* is the disk unit number. For example, **w5:0** denotes disk drive 0 on controller 5.

    Enter **s** if the target is a storage module. To specify a unit number on a storage module, append 0 or 1 to the letter. For example, **s1** denotes storage module unit 1.

13. If the target is a storage module, you are prompted for the drive type:

```
Possible Drive Types are:
1 - PN003863 (CDC)
2 - PN005100 (NEC)
3 - other (unknown)
Enter drive type:
```

Enter the number (1, 2, or 3) associated with the appropriate drive type.

14. **invol** displays the size and name of each logical volume on the disk, and prompts you for a logical volume number:

```
Physical volume "volume-name". Logical volumes:
 #     size(kB)    name
 1     xxx(d)

Enter logical volume number:
```

Enter the number of the target volume (the volume on which you plan to install Domain/OS). If the disk contains only one logical volume, enter **1**.

15. You are prompted for the paging file size:

```
Size in kB for the OS paging file (CR for default
value = xxx):
```

NOTE: The recommended size of the paging file for SR10 and SR10.1 is 590 blocks. For SR10.2, the recommended size is 2048 blocks for Series 10000 workstations and 640 blocks for all other workstation types.

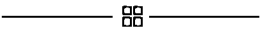
Enter the desired size, or press <RETURN> to accept the default value.

16. **invol** sets the paging size and asks,

```
Anything more to do?
```

Enter **no. invol** completes execution and returns to the calling program (the Mnemonic Debugger or a shell).

Return to where you left off in Chapter 2 or Chapter 3.



Appendix B

Running calendar

This appendix explains how to use the **calendar** program when you are following one of the procedures described in Chapter 2 or Chapter 3. For a general description of the **calendar** program, refer to the online manual pages for **calendar** and the command reference manual for your environment (Aegis, BSD, or SysV).

1. When you invoke **calendar**, you are first prompted for the type of disk:

   ```
   Please select the disk
   [w=Winch|s=Storage mod|f=Floppy|q=Quit]
   [ctrl#:] [unit#] [,lvno].

   If you do not have a disk, enter none (n):
   ```

   Enter **w** if the target is a Winchester disk.

   Enter **w***ctrl#*:*unit#* if the target is a Winchester disk on a Series 2500 workstation, where *ctrl#* is the disk controller number and *unit#* is the disk unit number. For example, **w5:0** denotes disk drive 0 on controller 5.

   Enter **s** if the target is a storage module. To specify a unit number on a storage module, append 0 or 1 to the letter. For example, **s1** denotes storage module unit 1.

2. You are asked if you want to reset the target node's current time zone setting. For example,

```
The time-zone is set to -4:00 (EDT). Would
you like to reset it?
```

Enter **n** if the time zone is correct, then proceed to step 4. Enter **y** if the time zone is incorrect.

3. If you choose to change the time zone, this prompt appears:

```
Please input the time-zone by entering either:

    - a time-zone identifier (EST, EDT, CST,
      CDT, MST, MDT, PST, PDT, GMT, or UTC),
      or

    - the difference between your time-zone
      and Universal Coordinated Time in the
      form "hour:minutes" or "-hour:minutes"
      (e.g. 9:00, -3:00). Time-zone differ-
      ences west of Greenwich are negative and
      those east of Greenwich are positive.

      Time-zone:
```

Enter your time zone (for example, **est** for Eastern Standard Time, or **edt** for Eastern Daylight Time) *or* enter the time difference, as described in the prompt. Note that GMT (Greenwich Mean Time) is the same as UTC (Universal Coordinated Time).

4. The program displays the node's current setting of the date and time, and asks if you want to reset it. For example,

```
The calendar date/time is 1989/07/20 09:13:37
EDT.
Would you like to reset it?
```

Enter **n** if the date and time are correct. **calendar** completes execution; return to where you left off in Chapter 2 or Chapter 3. Enter **y** if the date or time is incorrect.

5. If you choose to change the date and time, you are prompted for a new date and, after you enter a date, a new time. For example,

```
Please enter today's date (year/month/day):
1989/07/20
```

```
Please enter the local time in 24 hour format
(hour:minutes): 09:25
```

Note that if you set the time backward, you receive a warning and are asked to confirm your time selection:

```
Warning: setting the time backward may cause
duplicate unique ID's to be generated. Is the
above information correct?
```

If you set a node's time backward, it is *theoretically* possible that a unique identifier (UID) generated after the time change may be generated at the same node time as an UID generated before the time change. This results in duplicate UIDs. In our context, an object subsequently installed on the target node or restored to the node after the install may end up with the same UID as an object generated by the **invol** program.

However, because a UID's time stamp is accurate to within milliseconds, the possibility of duplicate UIDs is extremely remote; the danger in setting the time backward is minimal. To be absolutely certain that duplicate UIDs are not generated, after you finish running calendar, wait for the interval that you set the time backward before you perform any other actions on the node.

If you set the time forward more than five minutes, you are also asked to confirm your response:

```
The calendar is being set forward by more
than 5 minutes.
Is the above information correct?
```

You are asked to confirm your response to prevent the need for setting the time backward should you set the time forward erroneously.

6. After you specify a new date and time and, if necessary, confirm your response, **calendar** displays the following message and completes execution:

```
If running online, you should now shutdown
and reboot the system to run with the new
calendar setting.
```

Ignore this message and return to where you left off in Chapter 2 or Chapter 3.

Appendix C

Command Reference

This appendix provides a detailed description of each installation tool in the same format as that used for online help files. The command line options available for each tool are described.

NAME

cfgsa – create selection and override files

SYNOPSIS

cfgsa *AA*

DESCRIPTION

The **cfgsa** tool creates selection files and override files based on the release indexes in the Authorized Area at *AA*. Selection files are used with **distaa** to load subsets of products from distribution media into an Authorized Area. Override files are used with **config** or **install++** to constrain product configurations to a subset of those permitted by the product release indexes.

The **cfgsa** tool uses the release indexes for the products in the Authorized Area at *AA* as the source of its configuration questions. The **save** command to **cfgsa** creates an override file and places it in the **install/overrides** directory of this Authorized Area, making it an "active" override file.

The **cfgsa** tool is interactive and accepts the following commands at its `CFGSA>` prompt.

COMMANDS

Although they are shown here in lowercase, you can enter **cfgsa** commands in upper, lower, or mixed case. You can also abbreviate them to the point of uniqueness. The **cfgsa** tool accepts the following commands at its `CFGSA>` prompt:

available

Display a list of all the products in the Authorized Area at *AA*. This is the list of products you can choose from with the **select** command. The **cfgsa** tool executes an **available** command when it starts up.

constrain

Begin a constraint session for the selected product. (See the **select** command below). The **constrain** command causes **cfgsa** to present the series of configuration questions defined by the release index and constrained by the current active override file for the product, if one exists. After displaying each question, **cfgsa** presents three types of possible constraints:

user At configuration time, allow the user the full range of choices permitted by the release index for the product. This is the default constraint type selected by pressing `<RETURN>` at the constraint type prompt.

limit At configuration time, limit the user to a subset of the choices permitted by the release index. Choosing **limit** causes **cfgsa** to display the set of responses to the question allowed by the release index. You can then choose a subset of those responses to be presented at configuration time.

answer Answer the question for the user. Choosing **answer** causes **cfgsa** to display the set of responses allowed by the release index. The response

you choose becomes the answer to the configuration question posed by the release index; the user will not see the question at configuration time.

The CFGSA> prompt returns after all of the configuration questions posed by the release index have been presented for constraint. A constraint session can be terminated early by using the **abort** command.

exit Exit the **cfgsa** tool. If you have not yet saved all the constraints you have made, **cfgsa** asks for confirmation before terminating.

generate *template*
Create a selection file named **aa.***template* and a matching override file named **ov.***template*, in the current working directory for the selected product.

help Display a summary of the **cfgsa** commands accepted at the current prompt.

revert Remove all constraints imposed on the selected product during the current **cfgsa** session.

save Create an active override file for the product in the *AA*/**install/overrides** directory. If an active override file for the product already exists in the Authorized Area, **cfgsa** asks you to confirm that you want to overwrite the existing active override file.

select *product_name version_number*
Select version *version_number* of the product named *product_name* from the list of available products.

show List all of the questions in the release index for the selected product, together with the constraints imposed on the product configuration by the currently active override file and any choices made during the current **cfgsa** session.

EXAMPLES

```
cfgsa //sr10_node/authorized_area
```

This command begins an interactive configuration session for the products loaded into the Authorized Area at **//sr10_node/authorized_area**.

FILES

AA/**install/tools/cfgsa**
The executable for Domain/OS systems.

AA/**install/tools_sr9/cfgsa**
The executable for pre-SR10 systems.

AA/**install/help/cfgsa.hlp**
The help file for **cfgsa**.

AA/**install/ri.***company_name.product_name***.v.***version_number*
The product release directory for the product named *product_name* from the company named *company_name* in the Authorized Area at *AA*. The product

release directory is the source area containing the constituent objects of the product and the product release index. The product release index has the name **ri.***company_name*.*product_name*.**v.***version_number* in the product release directory.

AA/**install/overrides/ri.***company_name*.*product_name*.**v.***version_number*
The active override file for the product named *product_name* from the company named *company_name* in the Authorized Area at *AA*. If an active override file exists for a product, **cfgsa** will only ask questions consistent with the configuration restrictions defined in the active override file. You create an active override file with **cfgsa** by using the **save** command.

./**aa.***template*
The selection file created by the **generate** *template* command.

./**ov.***template*
The override file created by the **generate** *template* command.

SEE ALSO
distaa.hlp

NAME

config – create a configuration file

SYNOPSIS

config -s *AA* **-c** *configuration_file*

DESCRIPTION

Config is an interactive tool for creating and modifying configuration files. A configuration file defines a configuration of software products that can be installed. Configuration files are used to direct the **install** tool, which performs the actual installation.

OPTIONS

The **config** tool has two required command line options.

-s *AA* The -s option is used to specify the Authorized Area that contains the products to be configured. The **config** tool uses the release indexes for the products in the Authorized Area at *AA* and any constraints imposed by active override files in the *AA*/**install/overrides** directory as sources for the configuration questions it asks.

-c *configuration_file*

The **-c** option is used to specify the pathname of the configuration file that **config** creates or modifies. The product configuration you establish by using **config** is stored in *configuration_file*.

COMMANDS

Although they are shown here in lowercase, you can enter **config** commands in upper, lower, or mixed case. You can also abbreviate them to the point of uniqueness. The **config** tool accepts the following commands at its `CONFIG>` prompt:

abort Exit the **config** tool. Do not modify the configuration file with input from the **config** session. The **config** tool discards your input.

configure *product_name* [*version_number*]

Configure the product named *product_name*. The **configure** command causes the **config** tool to present configuration questions as defined in the product release index and constrained by the active override file, if one exists for the product. The **config** tool asks only the questions not already answered in the configuration file specified on the command line. You can abbreviate your responses to the configuration questions to the point of uniqueness.

deselect *product_name* [*version_number*]

Deselect a selected product. The **deselect** command removes the product named *product_name* from the configuration.

exit Exit the **config** tool and modify the configuration file named on the command line to reflect the results of the **config** session. At installation time, any unanswered questions for selected products will be assigned their default answers as defined in the release index.

help Display a summary of the commands accepted at the CONFIG> prompt.

install checking *product_name* [*version_number*] *check_type*
Set the type of configuration checking to perform at installation time for the product named *product_name* to *check_type*. There are three possible values for the *check_type* argument:

none Ignore any constituent objects of the product already installed on the target.

exist Install the constituent objects of the product named *product_name* only if they already exist on the target. Use this type of install checking to prevent replacing objects that have been deliberately deleted since the previous installation of a product.

version Install an object only if the version in the product named *product_name* differs from the version already installed on the target. This type of install checking implies **install checking exist**. An object is not installed if the object does not already exist on the target.

reconfigure *product_name* [*version_number*]
Reconfigure the product named *product_name*. Ignore any configuration information already in the configuration file, or established during the **config** session. The **reconfigure** command causes the **config** tool to present all the configuration questions defined in the product release index as constrained by the active override file, if one exists for the product.

select *product_name* [*version_number*]
Select version *version_number* of the product named *product_name*. You must select a product with the **select** command before you can configure it with the **configure** command. The selected product is added to the configuration.

select all
Select the latest version of all products in the Authorized Area at *AA* as displayed by the **show available** command.

show available
Display a list of products available for installation from the Authorized Area at *AA*. The **show available** command lists the version numbers of each product together with the product's name. The **config** tool executes a **show available** command when it starts up.

show queries *product_name* [*version_number*]
Display all configuration questions for the product named *product_name* together with their current answers. You can use the **configure** and **reconfigure** commands to change the answers shown by the **show queries** command.

show selections
Display the products selected for installation. Use the **select** command to add to the list of selected products; use the **deselect** command to remove a product from the list.

update *product_name* [*version_number*]
Deselect all previous versions of the product named *product_name*, and then select the latest version of the same product from the Authorized Area.

update all
Deselect all previous versions of all available products, and then select the latest versions of all products available in the Authorized Area.

validate *product_name* [*version_number*]
Determine whether installation of the product named *product_name* from the Authorized Area at *AA* will succeed with the current configuration.

EXAMPLES

```
config -s //sr10_node  -c //mynode/config_file
```

This command begins a configuration session for the products in the Authorized Area at **//sr10_node**. The **config** tool will modify the configuration defined in **//mynode/config_file**, or create a configuration file there if one does not already exist.

FILES

AA/**install/tools/config**
The executable for Domain/OS systems.

AA/**install/tools_sr9/config**
The executable for pre-SR10 systems.

AA/**install/help/config.hlp**
The help file for **config**.

AA/**install/ri.***company_name*.*product_name*.**v.***version_number*
The product release directory for the product named *product_name* from the company named *company_name* in the Authorized Area at *AA*. The product release directory is the source area containing the constituent objects of a product and the product release index. The product release index has the name **ri.***company_name*.*product_name*.**v.***version_number* in the product release directory.

AA/**install/overrides/ri.***company_name*.*product_name*.**v.***version_number*
The active override file for the product named *product_name* from the company named *company_name* in the Authorized Area at *AA*. If an active override file exists for a product, **config** will only ask questions consistent with the configuration restrictions defined in the active override file.

SEE ALSO

install++.hlp

NAME

distaa – load products from distribution media into an Authorized Area

SYNOPSIS

distaa [**-afv**] [**-e** *maximum_errors*] [**-m** *media_type*] *AA* [*selection_file*]

DESCRIPTION

The **distaa** tool loads software products from distribution media into the Authorized Area at *AA*. If *AA* is not the pathname of an existing Authorized Area, **distaa** will create an Authorized Area there.

You can load every product on the distribution media by using the **-a** option, or you can use a selection file to direct **distaa** to load only a subset of the products available on the distribution media. The *selection_file* argument can be the pathname of a selection file that ships with the product in the **install/templates** directory, or it can be the pathname of a file you created with the **cfgsa** tool. If you use a selection file to load only a subset of a product, move the override file associated with the selection file into the *AA*/**install/overrides** directory.

OPTIONS

-a You must specify either the **-a** option or the *selection_file* argument. The **-a** option causes **distaa** to load every product in its entirety from the distribution media into the Authorized Area. Without the **-a** option, **distaa** will only load the objects specified in *selection_file*.

-e *maximum_errors*

The **-e** option sets to *maximum_errors* the upper limit on the number of restoration errors that **distaa** will allow before aborting the load. The *maximum_errors* argument must be a decimal integer. If you don't use the **-e** option, **distaa** will not abort due to restoration errors.

-f The **-f** option forces **distaa** to load every object into the Authorized Area even if the object already exists in *AA*. Without the **-f** option, **distaa** copies only objects which are not already present in the Authorized Area.

-m *media_type*

Use the **-m** option to specify the type of distribution media that **distaa** is to load from. The argument *media_type* can be any one of the following:

c Cartridge tape.

f Floppy disk.

m Open-reel magnetic tape.

-v The **-v** option specifies verbose mode. The **distaa** tool lists each object as it is loaded. Without the **-v** option, **distaa** displays an abbreviated listing.

EXAMPLES

```
distaa -a //sr10
```

This command loads all the software that is on the release media into the Authorized Area located at **//sr10**.

FILES

AA/**install/tools/distaa**
The executable for Domain/OS systems.

AA/**install/tools_sr9/distaa**
The executable for pre-SR10 systems.

AA/**install/help/distaa.hlp**
The help file for **distaa**.

AA/**install/templates**/*company_name*/*product_name*.**v**.*version_number*/**aa**.*template*
and

AA/**install/templates**/*company_name*/*product_name*.**v**.*version_number*/**ov**.*template*
A selection file and override file pair that ships with the product named *product_name* from the company named *company_name*.

SEE ALSO

cfgsa.hlp, minst.hlp

NAME

fix_10_1_ri – correct the version number in ri.apollo.os.v.10.1

SYNOPSIS

fix_10_1_ri *AA*

DESCRIPTION

The **fix_10_1_ri** program corrects the version number in the **ri.apollo.os.v.10.1** release index in the Authorized Area at *AA*. You should run it at least once for every Authorized Area into which you have loaded version 10.1 of the Domain/OS product.

The **fix_10_1_ri** program has no effect if the Authorized Area at *AA* does not contain version 10.1 of the Domain/OS product, or if the release index has already been fixed. Consequently, it is not a bad idea to run **fix_10_1_ri** over every Authorized Area at your site if you have loaded 10.1 onto your network.

EXAMPLE

```
fix_10_1_ri //auth_area
```

This command fixes the version number in the release index: **//auth_area/install/ri.apollo.os.v.10.1/ri.apollo.os.v.10.1**

FILES

AA/**install/tools/fix_10_1_ri**
The executable for Domain/OS systems.

AA/**install/tools_sr9/fix_10_1_ri**
The executable for pre-SR10 systems.

AA/**install/ri.apollo.os.v.10.1/ri.apollo.os.v.10.1**
The release index for version 10.1 of the Domain/OS product.

NAME

inprot – modify object protection

SYNOPSIS

inprot -a *protection_file* **-t** *target_tree*

DESCRIPTION

The **inprot** tool modifies Access Control Lists (ACLs) for objects as described in *protection_file*. You can only use the **inprot** tool to apply ACLs to objects resident on Domain/OS nodes.

OPTIONS

-a *protection_file*

The file at the pathname following the **-a** option contains a list of object pathnames followed by the object ACLs.

-t *target_tree*

The **inprot** tool uses the pathname following the **-t** option as the working directory when evaluating any relative pathnames contained in *protection_file*. The pathname *target_tree* must refer to a directory on a Domain/OS node. Using **inprot** to set ACLs for objects on pre-SR10 nodes may yield unexpected results.

Format of an ACL Definition File

The ACL definition file contains a series of single-line records introduced by a single keyletter in the first column that defines the type of information on the line. The **inprot** tool currently recognizes the following keyletters:

Keyletter	Information on Line
D	the pathname of a directory
E	an extended ACL entry
F	the pathname of a file
G	a group ACL entry
O	an organization ACL entry
P	a person ACL entry
W	a world ACL entry

The keyletters are followed by one or more fields with the following format:

F	*path_name*		
P	*person_name*	*required_rights*	[-setuid | -setuid off]
G	*group_name*	*required_rights*	[-setuid | -setuid off]
O	*organization_name*	*required_rights*	[-setuid | -setuid off]
W	*world_name*	*required_rights*	
E	*subject_identifier*	*extended_rights*	
E	*subject_identifier*	*extended_rights*	
	.		
	.		
	.		
D	*path_name*		[-files | -dirs]
P	*person_name*	*required_rights*	[-setuid | -setuid off]
G	*group_name*	*required_rights*	[-setuid | -setuid off]
O	*organization_name*	*required_rights*	[-setuid | -setuid off]
W	*world_name*	*required_rights*	
E	*subject_identifier*	*extended_rights*	
E	*subject_identifier*	*extended_rights*	
	.		
	.		
	.		

Object Records (D and F)

Records that begin with the keyletters **D** or **F** are called **object records** because they define an object. The *path_name* field can be any pathname associated with a directory or file object. All succeeding records up to the next object record define modifications to an ACL associated with the object at *pathname*.

F records specify file objects and **D** records specify directory objects. There is only one type of **F** record, but there are three types of **D** records: object ACL records, initial directory ACL records, and initial file ACL records.

A directory's object ACL
: is similar to the **F** record for introducing a file object ACL. Succeeding ACL entry records define the permissions applied to the directory object itself. It is introduced by a **D** record without any modifier following the *path_name* field.

A directory's initial file ACL
: is the ACL associated with new files created within that directory. It is introduced by a **D** record with the **-files** modifier following the *path_name* field.

A directory's initial directory ACL
: is the ACL associated with new subdirectories created within that directory. It is introduced by a **D** record with the **-dirs** modifier following the *path_name* field.

ACL Entry Records
: Records that begin with the keyletters **P**, **G**, **O**, **W**, or **E** are called ACL entry records because they define entries for the ACL belonging to the object named in the last object record.

There are two types of entries in an ACL. The person, group, organization, and world entries are called **required** entries because they must be present in every ACL on the system. Required ACL entries are introduced by the **P**, **G**, **O**, and **W** keyletters in the ACL definition file, *protection_file*. Each required entry associates the name of a person, group, or organization with a set of rights. The user, group, and world entries in an ACL correspond to the UNIX model's user, group, and other permissions, respectively. The organization entry corresponds to the rights that can be granted to a user if he is in the named organization as defined in the **/etc/org** file; just as the group entry corresponds to the rights that can be granted to members of a group as defined in **/etc/group**.

An ACL can also contain **extended** entries. Extended ACL entries are introduced by the **E** keyletter in the ACL definition file, *protection_file*. Each extended entry associates a subject identifier (SID) with a set of rights. A SID specifies a person, group, and organization and has the following format:

person_name.group_name.organization_name

Each ACL entry has a set of rights associated with it. The set of rights available for use with required entries is **pwrxkjiu**:

p Grants the subject permission to change the ACLs associated with the object.

w Grants the subject permission to modify the object. The **w** right allows the subject to write to a file, or to add or remove objects from a directory.

r Grants the subject permission to read the object. The **r** right allows the subject to read a file, or to list the objects in a directory.

x Grants the subject permission to execute a file object or to resolve pathnames through a directory object. The **x** or "search" right on a directory allows the subject to use objects in the directory and to make the directory the working directory without granting the subject permission to list the directory's contents. Thus if a subject has only the **x** right on the directory **foo**, the subject cannot list the contents of **foo** or make **foo** its working directory, but if the subject knows beforehand that **foo** contains a file **bar** for which it has **w** and **r** rights, the subject can still read

or write **foo/bar**.

k Prevents the subject from removing the object or changing its name.

j Causes a required ACL entry to be ignored when determining rights to the object. The **j** right is incompatible with all but the **k** right.

i When specified as an initial file ACL or an initial directory ACL for a directory object, the **i** right causes the rights for a required entry to be inherited from the process that creates the object. The **i** right is not valid for extended ACL entries or for file object or directory object ACLs.

u When specified as an initial file ACL or an initial directory ACL for a directory object, the **u** right causes the rights for a required entry to be those requested by the process that creates the object as modified by the process's file creation mode mask. The **u** right is not valid for extended ACL entries or for file object or directory object ACLs.

Any valid set of the rights available for use with required entries may be used for the *required_rights* field. A **P**, **G**, or **O** record can include the **-setuid** option after its *required_rights* field. The **-setuid** option is used to add or remove set-user-ID rights from an executable binary object. If the **-setuid** option is omitted, the current state of the set-user-ID right is left unchanged. The **-setuid off** option removes the set-user-ID right for the entry. The **-setuid** option alone adds the set-user-ID right for the entry if it isn't present already.

The set of rights available for use with extended entries is **pwrx**. These rights have the same meanings as with required entries. Any valid set of the rights available for use with extended entries may be used for the *extended_rights* field.

If you omit a required ACL entry record from an ACL specification, the object defined by the last object record retains the current rights associated with that entry. To specify that no rights are to be associated with a required entry, leave the *required_rights* field of the ACL entry record blank.

Application Rules

The **inprot** tool applies the ACLs you specify in *protection_file* according to the following rules:

Required Entries

The following rules describe the application of **P**, **G**, **O**, and **W** records.

If the current ACL for the object grants permission, then **inprot** will assign to the entry the rights specified in *required_rights* and set the person, group, or organization component of the SID for that entry to the name specified in the *person_name*, *group_name*, or *organization_name*. (World ACL entries have no name associated with them.)

Extended Entries

If the current ACL for the object grants permission, then **inprot** will assign the rights specified in *extended_rights* to the ACL entry specified by *subject_identifier*. A new entry is created for the SID specified in the **E** record if it does not already exist in the ACL.

EXAMPLE

```
inprot -a protection -t //new_node
```

This command modifies the protections set on the files in **//new_node** and its subdirectories as described in the file **myfile**.

FILES

AA/**install/tools/inprot**

The executable for Domain/OS systems.

AA/**install/tools_sr9/inprot**

The executable for pre-SR10 systems.

SEE ALSO

In a SysV UNIX environment, consult the following manual entries for more information on ACLs: acl(5), chacl(1), chgrp(1), chmod(1), chown(1), cpacl(1), edrgy(1M), lsacl(1), org(4), passwd(4), salacl(1M), umask(1).

In a BSD UNIX environment, consult the following manual entries for more information on ACLs: acl(7), chacl(1), chgrp(1), chmod(1), chown(8), cpacl(1), edrgy(8), lsacl(1), org(5), passwd(5), salacl(8) umask(2).

In an Aegis environment, consult the following help entries for more information on ACLs: acl, acls, edacl, edrgy, protection, protection acls, protection rights, protection sids, salacl, umask.

NAME

install – install a product configuration

SYNOPSIS

install [**-aefhilmopruvx**] **-s** *AA* [**-c** *configuration_file*] **-n** *target_list*
install [**-aefhilmopruvx**] **-s** *AA* [**-c** *configuration_file*] *target* [*target*] ...

DESCRIPTION

The **install** tool installs a product configuration defined in one or more configuration files at one or more target pathnames. The constraints imposed by the active override files in the Authorized Area at *AA* take precedence over the configuration defined on the command line for any products selected in the product configuration.

OPTIONS

While the **install** tool has many command line options, you need to specify only two options to run **install**. You must use the **-s** option to specify an Authorized Area containing the products to be installed, and you must choose the **-a**, **-c**, or **-u** option to specify a configuration of those products.

-a Install on the target the latest version of all available products in the Authorized Area. The **-a** option is incompatible with the **-c** and **-u** options. The products are installed according to the default configurations defined in the release indexes.

-c *configuration_file*
Install the product configuration defined in the configuration file at the pathname *configuration_file*. You can use the **-c** option multiple times to specify multiple configuration files, which together define a configuration. The **-c** option is incompatible with the **-a** and **-u** options.

-e Check whether all files required for an installation are present in the Authorized Area before beginning the installation. In a multiple target installation, this test is done on a target-by-target basis.

-f Write out the new baseline file only after all products in the configuration are installed to a target. The default is to update the baseline file after each product is installed.

-h Ignore hardware compatibility checking. By default, the **install** tool will not install a product with an ISP type of **a88k** (that is, one released for a DN10000 series node) to a target directory that resides on a node with an ISP type of **m68k** (a non-DN10000 node), or vice versa. It will instead print an error message. The **-h** option forces the **install** tool to install all products in the configuration to all specified targets, regardless of ISP type incompatibilities.

-i Ignore product configuration size checking. By default, the **install** tool checks that the product configuration is smaller than the amount of free disk space on the target and, if it isn't, prompts you about whether to proceed with the installation. This option is most useful in unattended installations when you are

updating already installed products, since most of the space required by the new product versions is available after deleting the old versions.

-l When a target pathname is on the same node as the Authorized Area, install by creating hard links at the target to the objects in the product directories of the Authorized Area, rather than by copying the objects from the Authorized Area. This saves disk space on the Authorized Area node by not duplicating objects.

-m Replace any objects that were deleted from a product subsequent to its installation on a target. By default, **install** will not replace an object that was deleted from a product when that product is reinstalled or updated. The **-m** option also causes **install** to replace symbolic links on the target with file objects from the Authorized Area where called for in the configuration.

-n *target_list*
Specifies the pathname of a file containing a list of target directories. The file at *target_list* contains the pathnames of target directories—one pathname per line.

-o Install to each target as many times as the target is specified on the command line. By default, the **install** tool installs only once to each target, even if the target is repeated on the command line.

-p Purge all but the most recent baseline file for each target.

-r Run installation processes remotely on target nodes that have the Server Process Manager (**/sys/spm/spm**) running. The **-r** option is incompatible with the **-a** and **-u** options.

-s *AA* Specifies the pathname of the Authorized Area from which the **install** tool will install the product configuration. The **-s** option, together with the *AA* argument, is required. The Authorized Area at *AA* must contain the products defined in the configuration specified on the command line.

-t Test whether the specified configuration can be installed from the specified Authorized Area to one or more target directories, without actually installing products on the targets.

-u Update the target with the latest version available in the Authorized Area of each product specified in the baseline file at the target. The **-u** option is incompatible with the **-c** option.

-v Causes **install** to report more information while running. Without the **-v** option, **install** displays an abbreviated listing.

-x Continue if an error is encountered. An installation error can often be corrected without having to rerun **install**. By default, **install** will abort the installation when it encounters an error.

EXAMPLES

The following is a common **install** command line:

```
install -pvx -s //color -c //myhome/myfile //myhome
```

This command installs the product configuration defined in **//myhome/myfile** from the Authorized Area at **//color** to the node whose node entry directory is **//myhome**.

FILES

AA/**install/tools/install**
The executable for Domain/OS systems.

AA/**install/tools_sr9/install**
The executable for pre-SR10 systems.

AA/**install/help/install.hlp**
The help file for **install**.

AA/**install/ri.***company_name*.*product_name*.**v.***version_number*
The product release directory for the product named *product_name* from the company named *company_name* in the Authorized Area at *AA*. The product release directory is the source area containing the constituent objects of a product and the product release index.

AA/**install/overrides/ri.***company_name*.*product_name*.**v.***version_number*
The active override file for the product named *product_name* from the company named *company_name* in the Authorized Area at *AA*. If an active override file exists for a product, any restrictions defined in the file override the configuration for that product as specified on the command line.

target/**install/baseline/baseline.***number*
A file created in the target directory by **install** to record the configuration of products installed on the target.

target/**install/not_installed**
A file created by **install** to record the pathnames of any objects that could not be installed on the target. The **install** tool checks the **not_installed** file and, if possible, installs any objects named there from the Authorized Area at *AA*.

target/**install/doc/***company_name*/*product_name*.**v.***version_number*__**notes**
The release notes for the product named *product_name* from the company named *company_name* for a product in the configuration installed by the **install** tool.

SEE ALSO
install++.hlp

NAME

install++ – configure products and install

SYNOPSIS

install++ [**-adefhiklmopruvx**] **-s** *AA* [**-c** *configuration_file*] [**-C** *file_list*] **-n** *target_list*
install++ [**-adefhiklmopruvx**] **-s** *AA* [**-c** *configuration_file*] [**-C** *file_list*] *target* [*target*] ...

DESCRIPTION

The **install++** tool configures and installs software products to one or more target directories specified on the command line. The **install++** tool works by optionally invoking **config** to establish a product configuration, and then invoking **install** to install the product configuration. In effect, **install++** combines most of the functionality of **config** and **install** into one program.

OPTIONS

While there are many **install++** options, only the **-s** option is required.

The **install++** tool requires the **-s** option to specify an Authorized Area that is used by both the **config** and **install** programs. **install++** also requires at least one target pathname either specified directly on the command line, or listed in the file specified by the **-n** option. If no configuration file is specified via the **-c** option, **install++** invokes **config** to create a temporary configuration file, which it then passes to **install**. In addition to all the options accepted by **config** and **install**, **install++** takes the **-C**, **-d**, and **-k** options as described below. The **install++** tool is not generally suitable for unattended installations or for remote installations running on multiple targets.

-a Install the latest version of all products available in the Authorized Area. The **-a** option is incompatible with the **-c**, **-C**, **-d**, **-k**, and **-u** options. The products are installed according to the default configurations defined in the release indexes.

-c *configuration_file*
Install the product configuration defined in the configuration file at the pathname *configuration_file*. You can use the **-c** option multiple times to specify multiple configuration files, which together define a configuration. The **-c** option is incompatible with the **-a** and **-u** options.

-C *file_list*
Install the product configuration defined by the configuration files listed in the file at *file_list*. The file at *file_list* contains the pathnames of configuration files—one pathname per line. You can use the **-C** option multiple times to specify multiple configuration file lists, which together define a configuration. The **-C** option is incompatible with the **-a** and **-u** options.

-d Answer the configuration questions with the defaults defined in the product's release index, if the specified product configuration contains unanswered questions for a product. The **-d** option is incompatible with the **-a**, **-k**, and **-u** options.

-e Check whether all files required for an installation are present in the Authorized Area before beginning the installation. In a multiple target installation, this test is done on a target-by-target basis.

-f Write out the new baseline file only after all products in the configuration are installed to a target. The default is to update the baseline file after each product is installed.

-h Ignore hardware compatibility checking. By default, the **install** tool will not install a product with an ISP type of **a88k** (that is, one released for a DN10000 series node) to a target directory that resides on a node with an ISP type of **m68k** (a non-DN10000 node), or vice versa. It will instead print an error message. The **-h** option forces the **install** tool to install all products in the configuration to all specified targets, regardless of ISP type incompatibilities.

-i Ignore product configuration size checking. By default, the **install** tool checks that the product configuration is smaller than the amount of free disk space on the target and, if it isn't, prompts you about whether to proceed with the installation. This option is most useful in unattended installations when you are updating already installed products, since most of the space required by the new product versions is available after deleting the old versions.

-k Require configuration questions to be answered interactively, if the specified product configuration contains unanswered questions for a product. The **-k** option is incompatible with the **-a**, **-d**, and **-u** options.

-l When a target pathname is on the same node as the Authorized Area, install by creating hard links at the target to the objects in the product directories of the Authorized Area, rather than by copying objects from the Authorized Area. This saves disk space on the Authorized Area node by not duplicating objects.

-m Replace any objects that were deleted from a product subsequent to its installation on a target. By default, **install** will not replace an object that was deleted from a product when that product is reinstalled or updated. The **-m** option also causes **install** to replace symbolic links on the target with file objects from the Authorized Area where called for in the configuration.

-n *target_list*
The **-n** option specifies the pathname of a file containing a list of target directories. The file at *target_list* contains the pathnames of target directories—one pathname per line.

-o Install to each target as many times as the target is specified on the command line. By default, the **install** tool installs only once to each target, even if the target is repeated on the command line.

-p Purge all but the most recent baseline file for each target.

-r Run installation processes remotely on target nodes that have the Server Process Manager (**/sys/spm/spm**) running. The **-r** option is incompatible with the **-a** and **-u** options.

-s *AA* Specifies the pathname of the Authorized Area from which the **install** tool will install the product configuration. The **-s** option, together with the *AA* argument, is required. The Authorized Area at *AA* must contain the products defined in the configuration specified on the command line.

-t Test whether the specified configuration can be installed from the specified Authorized Area to one or more target directories, without actually installing products on the targets.

-u Update the target with the latest version available in the Authorized Area of each product specified in the baseline file at the target. The **-u** option is incompatible with the **-c**, **-C**, **-d**, and **-k** options.

-v Report more information while **install++** runs.

-x Continue if an error is encountered. An installation error can often be corrected without having to rerun **install**. By default, **install** will abort the installation when it encounters an error.

EXAMPLES

```
install++ -pvx -s //color //myhome
```

This command interactively creates an installable configuration of products from the Authorized Area at **//color**, and installs them to the node whose entry directory is **//myhome**.

FILES

AA/**install/tools/install++**
The executable for Domain/OS systems.

AA/**install/tools_sr9/install++**
The executable for pre-SR10 systems.

AA/**install/help/install++.hlp**
The help file for **install++**.

AA/**install/tools/install**
The **install** tool executable for Domain/OS systems.

AA/**install/tools_sr9/install**
The **install** tool executable for pre-SR10 systems.

AA/**install/help/install.hlp**
The help file for **install**.

AA/**install/tools/config**
The **config** tool executable for Domain/OS systems.

AA/**install/tools_sr9/config**
The **config** tool executable for pre-SR10 systems.

AA/**install/help/config.hlp**
The help file for **config**.

AA/**install/ri.***company_name.product_name*.**v.***version_number*
The product release directory for the product named *product_name* from the company named *company_name* in the Authorized Area at *AA*. The product release directory is the source area containing the constituent objects of a product and the product release index.

AA/**install/overrides/ri.***company_name.product_name*.**v.***version_number*
The active override file for the product named *product_name* from the company named *company_name* in the Authorized Area at *AA*. If an active override file exists for a product, any restrictions defined in the file override the configuration for that product as specified on the command line.

target/**install/baseline/baseline.***number*
A file created in the target directory by **install** to record the configuration of products installed on the target.

target/**install/not_installed**
A file created by **install** to record the pathnames of any objects that could not be installed on the target. The **install** tool checks the **not_installed** file and, if possible, installs any objects named there from the Authorized Area at *AA*.

target/**install/doc/***company_name/product_name*.**v.***version_number*__**notes**
The release notes for the product named *product_name* from the company named *company_name* for a product in the configuration installed by the **install** tool.

SEE ALSO
config.hlp, install.hlp, minst.hlp

NAME

minst – load and install products from distribution media

SYNOPSIS

minst [*AA*]

DESCRIPTION

Minst loads software products from distribution media into an Authorized Area at the pathname *AA*. After products have been loaded into the Authorized Area, **minst** optionally installs them from that Authorized Area to one or more target directories. It takes one optional argument to specify the pathname of an Authorized Area into which products are to be loaded. If the optional argument is omitted, you are prompted for the pathname.

The **minst** tool is interactive and presents a series of questions. Your answers to the questions determine whether the Authorized Area loading phase will be followed by node installation and whether multiple products will be loaded and installed.

EXAMPLES

```
minst //color
```

This command loads software into the Authorized Area located at the node entry directory **//color**.

```
minst //color/disk1
```

This command loads software into the Authorized Area located in the directory **/disk1** on the node whose entry directory is **//color**.

FILES

*AA***/install/tools/minst**
The executable for Domain/OS systems.

*AA***/install/tools_sr9/minst**
The executable for pre-SR10 systems.

*AA***/install/help/minst.hlp**
The help file for **minst**.

*AA***/install/minst/***
Files containing **minst** question prompts and answers from users.

*AA***/install/tools/last_time**
File recording the **minst** tool's last update of the Authorized Area structure from distribution media.

*AA***/install/doc/***company_name*/*product_name*.v.*version_number*__**notes**
The release notes for the product named *product_name* from the company named *company_name*.

*AA***/install/templates/***company_name*/*product_name*.v.*version_number*/**cf.***product_name*
The default configuration file that ships with the product named *product_name*

from the company named *company_name*.

AA/**install/templates**/*company_name*/*product_name*.**v**.*version_number*/**aa**.*template*

AA/**install/templates**/*company_name*/*product_name*.**v**.*version_number*/**ov**.*template*
A selection file and override file pair that ships with the product named *product_name* from the company named *company_name*.

AA/**install/overrides/ri**.*company_name*.*product_name*.**v**.*version_number*
The active override file for the product named *product_name* from the company named *company_name* in the Authorized Area at *AA*.

SEE ALSO
distaa.hlp, install++.hlp

NAME

mrgri – merge products and release indexes

SYNOPSIS

mrgri [**-i**] -s *source_AA* [**-t** *target_AA*] [**-v** *new_version*] [**-p** *new_name*] *primary_product secondary_product*

DESCRIPTION

The **mrgri** tool merges the two products, *primary_product* and *secondary_product*, in the Authorized Area at *source_AA* into a single product with a single release index.

The **mrgri** tool has two main uses. It can patch a product within its Authorized Area by merging a patch product with a previously released product. (In the RAI scheme, a patch is just another product. An Apollo "patch product" has the product name **patch** followed by some sequence number and a product version number.) In this case, installation of the merged product has the same result as would installing the product first, and then installing the patch to the same target. The advantage of merging the patch is that there is only one product in the Authorized Area to manage and install instead of two. We recommend that you do not use the **-p**, **-t**, or **-v** option when patching products. Each of these options causes the product to be patched to be copied. This increases the time for merging significantly, especially when you are patching Domain/OS.

The other main use is the creation of compound products. With the advent of the Series 10000 workstations and servers, Apollo now releases two versions of most products: a version that runs on nodes based on Motorola's 68000 series of microprocessors, and a version that runs on the Series 10000 *PRISM* (*P*arallel *R*educed *I*nstruction *S*et *M*ultiprocessor) nodes. We encapsulate these two different CPU architectures with the concept of an **ISP type**, where "ISP" stands for "Instruction Set Processor." Series 10000 nodes belong to the **a88k** ISP type, and all other nodes belong to the **m68k** ISP type.

We distinguish the two different ISP types in product releases by adding the extension **.p** on the version field of the release index name for products with the **a88k** ISP type. Thus, **ri.apollo.os.v.10.2.p** is the name of the release index for the version 10.2 of Domain/OS that runs on Series 10000 nodes, whereas **ri.apollo.os.v.10.2** is the name of the release index for the version that runs on nodes with the **m68k** ISP type. You can sometimes use **mrgri** to merge two products that differ only in ISP type into a single product that can be installed and run on nodes of both ISP types. (The release notes will tell you which products can be merged.) A product that is the result of merging versions with different ISP types is called a **compound product**. A compound product has the ISP type **cmpexe**, which is short for "compound executable."

Unless you use one or more of the **-p**, **-t**, or **-v** options, the **mrgri** tool overwrites *primary_product* in the Authorized Area at *source_AA* with the new merged product. Consequently, you should use at least one of the **-p**, **-t**, or **-v** options if you want to retain the ability to install *primary_product* from the Authorized Area at *source_AA*.

OPTIONS

-i Display more informational messages. With the **-i** option, **mrgri** will report each object copied to the merged product directory, and additional execution

milestones.

-s *source_AA*
Specifies the pathname of the Authorized Area containing the products named in *primary_product* and *secondary_product*.

-p *new_name*
Specifies a new product name to be applied to the merged product. There is rarely a need for this option, and its use is not recommended.

-t *target_AA*
Specifies the pathname of the Authorized Area in which the merged product is to be created. If you do not use the **-t** option, the **mrgri** tool creates the product in *source_AA*.

-v *new_version*
Specifies a new version number to be applied to the merged product. This is the recommended way to distinguish a merged product from its constituent products.

EXAMPLES

The following command line applies patch number 1 (the product whose release index is **ri.apollo.patch_p0001.v.1.0**) to version number 10.2 of the Domain/OS product (with the release index **ri.apollo.os.v.10.2**) in the Authorized Area at **//AA**.

```
//AA/install/tools/mrgri -s //AA
       ri.apollo.os.v.10.2 ri.apollo.patch_p0001.v.1.0
```

The patched product is left in **//AA** with the its name unchanged. After you have merged the patch with the product, you may wish to delete the patch product. A good way to do this in a UNIX environment is to use **/bin/rm** like this:

```
/bin/rm -R //AA/install/ri.apollo.patch_p0001.v.1.0
```

In the Aegis environment, you can use **/com/dlt** like this:

```
/com/dlt //AA/install/ri.apollo.patch_p0001.v.1.0
```

The following command line merges versions 10.2 and 10.2.p of the Domain/OS product in the Authorized Area at //AA to produce a compound product that can run on nodes of both ISP types:

```
//AA/install/tools/mrgri -s //AA -t //server/AA -v 10.2.p.cmpexe
       ri.apollo.os.v.10.2.p  ri.apollo.os.v.10.2
```

The resulting compound Domain/OS product is created in the Authorized Area at **//server/AA**, and its release index has the name **ri.apollo.os.v.10.2.p.cmpexe**.

After the merge is complete, you can install and run the new merged product, **os.v.10.2.p.cmpexe**, on a node of either ISP type. You can also boot nodes of either ISP type diskless off of a node running the merged Domain/OS.

For example, if after the merge you actually install the compound Domain/OS on the node whose node entry directory is **//server**, you can use the node at **//server** as a diskless server for a mixed network of **m68k** and **a88k** nodes.

In the above example we did not use the **-p** option. Consequently, **mrgri** uses the name **ri.apollo.os** as the new product name. We recommend that you do not use the **-p** option when merging products. Instead, we recommend that you keep the product name and use the **-v** option to add an extension that identifies the new product as a compound version. Adding **cmpexe** to the version number for the product whose ISP type is a88k, as shown in the example, is a good way to do this.

NOTES

The **mrgri** tool uses the release indexes of the products it merges to control how it combines the objects that constitute the products. For two versions of a product to merge successfully, they must be designed to merge successfully and their release indexes must reflect that design. Therefore, you should not indiscriminately merge any two products. The release notes for a product state whether the product can be successfully merged with **mrgri**, which versions should be merged together, and what restrictions you must follow when installing and using the merged product.

You should also be aware that a product created by **mrgri** cannot be "unmerged" to recover its constituent products. If you delete the individual product release directories after merging them, or if you use **mrgri** to overwrite one of the individual products with the new merged product, that product cannot be recovered from the merged product. You must reload it from the distribution media or copy it from another Authorized Area.

FILES

AA/**install/tools/mrgri**
The executable for Domain/OS systems.

AA/**install/tools_sr9/mrgri**
The executable for pre-SR10 systems.

AA/**install/help/mrgri.hlp**
The help file for **mrgri**.

SEE ALSO

distaa.hlp, fix_10_1_ri.hlp, minst.hlp

Appendix D

Installing Third–Party Software

You can install Domain/OS with any combination of the three operating environments: BSD, SysV, and Aegis. Therefore, there is no guarantee that any individual Domain/OS node contains a particular environment.

Before SR10, the installation scripts of many of our solution suppliers relied on the existence of the Aegis environment on every node. With SR10, the existence of an Aegis environment cannot be assumed.

To reduce the transition problems of our solution suppliers, a **/sys5.3/bin** directory is installed on every node when system software is installed, even on nodes on which the SysV environment is not installed. This directory contains a minimal set of SysV commands to support the use of SysV Bourne shell installation scripts. The shell is located in **/etc/sys_sh**. The commands supplied are

awk	**cpio**	**ls**	**sum**
cat	**diff**	**mkdir**	**tar**
chgrp	**find**	**mv**	**tr**
chmod	**expr**	**rm**	**uniq**
chown	**grep**	**rmdir**	**wc**
cmp	**id**	**sed**	
cp	**ln**	**sort**	

NOTE: We recommend that solution suppliers caution customers against deleting this tree.

On a node with a SysV environment installed, the full **/sys5.3/bin** directory is automatically installed instead of this minimal set.

Glossary

Authorized Area

A directory that acts as a source area for software. Software is loaded from the distribution media into the Authorized Area. The software in the Authorized Area can then be configured and installed on nodes across the network.

Base operating system

The set of software objects required by all three Domain/OS environments: Aegis, SysV, and BSD.

Baseline file

A file created by the **install** or **install++** program that records cumulatively the software that **install** or **install++** installs on a node. The baseline files reside in the directory **/install/baseline** on the node on which the software is installed.

Boot media

The cartridge tapes or floppy disks that contain the objects necessary for booting your system when you get a new release of Domain/OS.

calendar

A program that sets the calendar and clock on a node. An offline version (accessed at the Mnemonic Debugger level) resides in each of the **/sau** directories; an online vesion resides in the **/com** directory.

Canned

Refers to a configuration, override, or selection file supplied by Apollo with a software release. Canned software configurations usually match the configurations most commonly used by our customers.

cfgsa

An interactive program for creating selection and override files. Selection files are used with the **distaa** tool to load only subcomponents of a product from distribution media. Override files enable you to restrict in advance the range of choices presented during the configuration phase of subsequent installations. **cfgsa** resides in the directory *authorized_area*/**install/tools**.

Closed environment

An operating environment with a closed protection model.

Closed protection model

A model in which only system administrators have control over and access to system software.

config

An interactive program that uses the release indexes for products in an Authorized Area to create or modify a file that controls the software configuration installed on a node. **config** resides in the directory *authorized_area*/**install/tools**.

Configure

To determine the elements of software to be installed.

Configuration file

A file consisting of a list of software product descriptions. Each product description, including the one for the Domain/OS product containing three operating environments, indicates a selection of product options. The options determine which specific objects belonging to the product are installed on a node by the **install** or **install++** program.

ctnode

A program that specifies a name for its own or another node's node entry directory. **ctnode** resides in the **/com** directory.

distaa

A program that loads software products from media into an Authorized Area, which may be distributed over more than one node in a network. It can load a complete set of software, transferring all the objects on the distribution media, or it can create a sparse Authorized Area, with only a subset of the objects. **distaa** resides in the directory *authorized_area*/**install/tools**.

Distributed Authorized Area

An Authorized Area in which products or parts of products are located on more than one disk or storage module. For purposes of installation, a distributed Authorized Area presents the same user interface as one that is not distributed.

Distribution media

The cartridge tapes, floppy disks, or magnetic tapes on which you receive a software release.

dmtvol

A program that unmounts a volume from a disk or storage module in the Aegis environment. **dmtvol** resides in the **/com** directory.

Hard Link

Another entry in the naming tree for the same file on disk. For example, if you create a hard link called **my_link** to a file called **my_file**, the result is two names (**my_link** and **my_file**) for the same file object, identified internally by its UID (Unique Identifier).

inprot

A program that sets up a protection model for the directories and files on a node, according to the description in an SR10.x template file. This tool can only be run on an SR10.x node. **inprot** resides in the directory *authorized_area*/**install/tools**.

install

A noninteractive program that installs software on one or more nodes, using a previously created configuration file. **install** resides in the directory *authorized_area*/**install/tools**.

install++

A program that configures and installs software on one or more nodes. **install++** resides in the directory *authorized_area*/**install/tools**.

Interactive mode

A method of using the **install++** program in which you configure software by answering queries about software products after invoking the program.

invol

A program that initializes a disk or storage module; resides in each of the **/sau** directories and also in the **/com** directory.

mount

A program that mounts a volume on a disk or storage module in a SysV or BSD environment. **mount** resides in the /etc directory.

mtvol

A program that mounts a volume on a disk or storage module in an Aegis environment. **mtvol** resides in the /com directory.

Node entry directory

The top-level directory on a node; a subdirectory of the network root directory.

Node ID

A unique, unchangeable, hexadecimal ID assigned to each node by the manufacturer.

Open environment

An operating environment with an open protection model.

Open protection model

A model in which users as well as system administrators have control over and access to system software.

Optional product

A product that can be optionally purchased separate from the operating system, and that requires the operating system to run. Examples of optional products include Pascal, FORTRAN, and DSEE.

Override

A method by which you restrict in advance the configuration or product options that can be subsequently installed.

Override file

A file created by the **cfgsa** program and used by the installation program to determine permissible configurations for a given product at a specific site. Override files reside in the directory *authorized_area*/**install/overrides**.

Partner or partner node

A node that provides the Domain/OS operating system for a node that is booted diskless from it.

Protection/permissions

A method for restricting access to software objects to prevent either accidental or intentional corruption.

Pull installation

An installation in which products are installed on multiple nodes simultaneously as the result of a single command. The node at which the installation is first invoked, and the target nodes, must all be running the Server Process Manager (**spm**) process.

Push installation

An installation in which products are installed on multiple nodes serially as the result of a single command.

Release index

A binary file containing the compiled description of a product, including provision for all possible installation options for that product.

/sau directory

A directory that contains hardware-specific utilities and programs. Each node family has its own **/sau** directory.

Secure network

A network in which all nodes have closed environments. In a secure network, every user has a password-protected account, and every user can set the permissions for his or her personal directories and files.

Selection file

A file that the **distaa** tool uses to determine which software objects to load from distribution media into an Authorized Area.

Soft link

A name that points to another name of an object (directory or file) in the naming tree. For example, if you create a soft link named **my_link** to the directory **//another_node/some_subdirectory**, the system substitutes the directory name **//another_node/some_subdirectory** (the link text) whenever you use the name **my_link**. Also known as a symbolic link.

Software configuration

The set of software products and product options installed, or selected for installation, on one or more nodes. For example, the software configuration on a node used for CAD/CAM might be the Domain/OS operating system with the Aegis and SysV environments, and the optional software products GPIO, PAS, and FTN, with links to **/sys/help**, **/domain_examples**, and **/usr/man** directories on another node in the network.

Standard system software

The Domain/OS software that is needed to run a node. It includes at least one of the three operating environments (Aegis, SysV, or BSD). Standard software releases are numbered SRx, SRx.y, or SRx.y.z.

Supplementary software

A set of software products that are included on the distribution media with the standard system software but are not a required part of the standard software. The installation program allows you to choose which supplementary software to install. Supplementary software includes but is not limited to

- Hardware diagnostics
- Online help files and manual pages
- Include files for systems programmers
- A **/systest** directory for testing system software
- Sample C, Pascal, and FORTRAN programs (Domain examples)

System directories

Directories that contain the standard system software and reside at the entry directory level. Some examples of these directories are **/com**, **/bin**, **/sau2**, **/sau_sys**, **/sys**, **/bscom**, **/lib**, **/doc**, and **/domain_examples**. Access rights to system directories can be assigned through the **inprot** program. For self-protection and network security, users do not always have full access rights to modify their system directories; a system administrator should have full access to system directories on all nodes.

System software

The software comprising the Domain/OS operating system and the three environments it provides. Includes both base operating software (such as the **/sys** and **/etc** directories) and environment-specific software such as **/usr/man** files for BSD, **/usr/catman** files for SysV, and **/sys/help** files for Aegis.

Target (of an initialization)

The disk or storage module that you're initializing.

Target (of an installation)

The directory into which you're installing software. The target can be a node or server entry directory (for example, **//color**) or any subdirectory (for example, **//color/pas** or **//color/disk1**).

uctnode

A program that removes a node name from its own or another node's node entry directory. **uctnode** resides in the /**com** directory.

umount

A program that unmounts a volume from a disk or storage module in a UNIX environment. **umount** resides in the /**etc** directory.

Work node (for a disk initialization)

The node at which you perform the initialization procedure. The work node and the target are identical unless you are initializing a disk or storage module connected to a Domain Storage Processor (DSP).

Work node (for an installation)

The node at which you perform the installation procedure. The target node and the work node can be the same.

Index

Symbols are listed at the beginning of the index.

Symbols

A

B

C

D

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

W

Reader's Response

Please take a few minutes to send us the information we need to revise and improve our manuals from your point of view.

Document Title: *Installing Software with Apollo's Release and Installation Tools*
Order No.: 008860-A02

Your Name _____ Date _____

Organization _____

Street Address _____

City _____ State _____ Zip _____

Telephone number (_____) _____

When you use the Apollo system, what **job(s)** do you perform?

- ☐ Programming
- ☐ Application End User
- ☐ Hardware Engineering
- ☐ System Administration
- ☐ Other (describe)_____

How many years of **experience** do you have in using the Apollo system:

_____

What programming **languages** do you use with the Apollo system?

_____

How would you **evaluate** this book?

	Excellent		Average		Poor
Completeness	1	2	3	4	5
Accuracy	1	2	3	4	5
Usability	1	2	3	4	5

Additional Comments: _____

_____

_____

No postage necessary if mailed in the U.S.

cut along dotted line

fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 78 CHELMSFORD, MA 01824

POSTAGE WILL BE PAID BY ADDRESSEE

APOLLO COMPUTER INC.
Technical Publications
P.O. Box 451
Chelmsford, MA 01824

fold

Reader's Response

Please take a few minutes to send us the information we need to revise and improve our manuals from your point of view.

Document Title: *Installing Software with Apollo's Release and Installation Tools*
Order No.: 008860-A02

Your Name _____ Date _____

Organization _____

Street Address _____

City _____ State _____ Zip _____

Telephone number (____) _____

When you use the Apollo system, what **job(s)** do you perform?

- ☐ Programming
- ☐ Application End User
- ☐ Hardware Engineering
- ☐ System Administration
- ☐ Other (describe)_____

How many years of **experience** do you have in using the Apollo system:

_____

What programming **languages** do you use with the Apollo system?

_____

How would you **evaluate** this book?

	Excellent		Average		Poor
Completeness	1	2	3	4	5
Accuracy	1	2	3	4	5
Usability	1	2	3	4	5

Additional Comments: _____

_____

_____

No postage necessary if mailed in the U.S.

fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 78 CHELMSFORD, MA 01824

POSTAGE WILL BE PAID BY ADDRESSEE

APOLLO COMPUTER INC.
Technical Publications
P.O. Box 451
Chelmsford, MA 01824

fold