Installing Domain Software

First Printing: March 1992

Warranty

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties or merchantability and fitness for particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

WARNING

Electrical Shock Hazard

Do not remove system covers. To avoid electrical shock, use only supplied power cords

Explosion Hazard

Do not operate the system in the presence of flammable anesthetics

Hewlett-Packard Co.
Learning Products
P.O. Box 451
Chelmsford, MA 01824

Installing Domain Software describes how to install software products on Domain systems using the Domain installation tools. Products installable in this way have the characters **RAI** on the label of their distribution media. They include the Domain/OS operating system, Products Support Kits (PSKs), optional products, and patches (bug fixes).

This manual documents the installation tools that are released with version 10.4 of Domain/OS (SR10.4). You can use these tools to install any **RAI** product, including **RAI** products released before SR10.4. Also, you can use these tools to install products on a node running any SR10.*x* version of Domain/OS; the target of the installation does not have to be running SR10.4.

The installation tools are shipped on the Domain/OS distribution media. The section "How Do I Restore the Installation Tools from Media?" in the introduction to Part I of this manual tells you how to get the new tools from the OS media onto a node.

This manual contains general installation information for all Domain software products. The release notes shipped with each product contain product–specific installation information.

This manual is divided into three parts:

Part 1, "Installation Tasks," provides step-by–step procedures for the most essential and commonly used installation tasks. The introduction to Part 1 describes the concepts you need to know to perform these tasks.

Part 2, "Advanced Installation Tasks," provides procedures for several optional and more advanced installation tasks. It is intended primarily for system administrators responsible for the installation and control of products at a site.

Part 3, "Installation Reference," provides a detailed description of the installation tools and the components of an Authorized Area. It also describes some common installation error messages. Unlike Parts 1 and 2, which provide step-by-step instructions for performing specific tasks, Part 3 describes the bits and pieces of the Domain installation model in a comprehensive and more traditional, expository fashion. You can read Part 3 as you would a text book or consult it as a reference when you perform tasks in Parts 1 and 2.

The individual chapters are organized as follows:

Part I **Installation Tasks**

Chapter 1 How to install the Domain/OS operating system from cartridge- or magnetic-tape distribution media on a new node or an existing node whose disk you want to initialize.

Chapter 2 How to install Domain/OS across the network from an Authorized Area to a new node or an existing node whose disk you want to initialize.

Chapter 3 A detailed description of the **invol** utility, provided as an optional supplement to Chapters 1 and 2.

Chapter 4 A detailed description of the **calendar** utility, provided as an optional supplement to Chapters 1 and 2.

Chapter 5 How to load products from media into an Authorized Area, and configure, install, and deinstall products.

Summary of Technical Changes

Installing Domain Software documents technical changes to the Domain installation tools that have been made since the release of version SR10.2 of the Domain/OS operating system. The new features of the installation tools enable you to:

- Deinstall an entire product, using the **-D** option of the **install** or **install++** tool.
- Deinstall parts of a product, using the **-d** option of the **install** or **install++** tool. The former **install++ -d** option (answer unanswered configuration questions with their default values) is now the **-j** option.
- Reset an installed product's Access Control Lists (ACLs) to their original settings (those defined by the product's release index), using the **-A** option of the **install** or **install++** tool.
- Reanswer a single configuration question for a product, rather than have to reconfigure the entire product. You do this with the **config** tool's new **reanswer** command.
- Specify default link-to text when configuring products, rather than have to repeatedly enter the same link-to destination. You do this with the **config** tool's new **set linkprompt** command.
- Merge PSKs with Domain/OS in an Authorized Area, using the **mrgri** tool. Prior to SR10.3, **mrgri** supported only the merging of patches with products and the merging of the a88k and m68k versions of a product. To accommodate this extension, **mrgri** has two new command-line options: **-merge** and **-cmpexe**.

Additionally, the functionality of the **install** program's **-m** option (ignore product customization) has been extended. When you reinstall or update a product, and the resident (previously-installed) product has local copies of directories that have been manually changed to links (or vice versa), **-m** now causes **install** to overwrite

these directories or links, as called for by the product configuration. Formerly, **install -m** did this for files, but not for directories.

The **inprot** tool, which enabled you to modify the Access Control Lists (ACLs) of installed objects, is no longer provided. **inprot** has been superceded by the **setprot** tool, which enables you to modify ACLs in a much easier way. **setprot** is *not* part of the Domain installation tool set; it is provided as part of the Domain/OS command set and documented in the *Domain/OS System Administration Reference.*

Related Manuals

The file **/install/doc/apollo/os.v.***latest software release number***__manuals** lists current titles and revisions for all available Domain manuals.

For example, at SR10.4 refer to **/install/doc/apollo/os.v.10.4__manuals** to check that you are using the correct version of 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 topics related to software installation, refer to the following documents:

- *Administering the Domain/OS Registry* *(015363)*
- *Aegis Command Reference* *(002547)*
- *Configuring and Managing TCP/IP* *(008543)*
- *Domain/OS BSD Command Reference* *(005800)*
- *Domain/OS SysV Command Reference* *(005798)*

- *Domain/OS System Administration Guide* *(019001)*
- *Domain/OS System Administration Reference* *(019208)*
- *Domain System Software Release Notes* *(017957)*
- *Getting Started with Domain/OS* *(002348)*
- *HP Visual User Environment Configuration Guide for Domain/OS Systems* *(B1171–90046)*
- *Managing NCS Software* *(011895)*
- *Network Computing System Reference Manual* *(010200)*
- *Using Your Aegis Environment* *(011021)*
- *Using Your BSD Environment* *(011020)*
- *Using Your SysV Environment* *(011022)*

You can order Domain documentation by calling **1–800–225–5290**. Outside the USA, please contact your local sales office.

Does This Manual Support Your Software?

This manual was released with version 10.4 of the Domain/OS operating system (SR10.4). It documents the installation tools released with SR10.4. You use this manual to install SR10.4 and later versions of Domain/OS. You can also this manual to install any RAI-installable optional products, patches, and PSKs on SR10.*x* based systems, provided you use the SR10.4 version of the installation tools. To install a pre–SR10.4 version of Domain/OS or to install other RAI products with pre–SR10.4 versions of the installation tools, use the previous version of this manual: *Installing Software with Apollo's Release and Installation Tools* (008860–A02).

To verify which version of operating system software you are running, type:

bldt

If you are running Domain/IX on a release of the operating system earlier than SR10.0, then type:

/com/bldt

If you are using a later version of software than that with which this manual was released, use one of the following ways to check if this manual was revised or if additional manuals exist:

- Read Chapter 3 of the release document that shipped Domain/OS. The release document is online: **/install/doc/apollo/os.v.10.*x*__notes**, where *x* is the version of Domain/OS. Check with your system administrator if you cannot find the release document.
- Telephone **1-800-225-5290**. Outside the USA, contact your local sales office.
- Refer to the lists of manuals described in the preceding section, "Related Manuals."

To determine which of two versions of the same manual is newer, refer to the part number that is printed on the title page or cover. Every part number has a 3-digit suffix; for example, **-A00**. A higher suffix number indicates a more recently released manual. For example, a manual with suffix **-A02** is newer than the same manual with suffix **-A01**.

Problems, Questions, and Suggestions

If you have any questions or problems with our hardware, software, or documentation, please contact either your HP Response Center or your local HP representative.

You may call the Tech Pubs Connection with your questions and comments about our documentation:

- In the USA, call 1–800–441–2909
- Outside the USA, call (508) 256–6600 extension 4965

The recorded message that you will hear when you call includes information about our new manuals.

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

Documentation Conventions

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

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.
sample user input	In interactive examples, information that the user enters appears in color.

`output/source code`	Information that the system displays appears in this `typeface`. Examples of source code also appear 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/ or ^ 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.

Contents

Part 3 Installation Reference

Tables

Part 1

Installation Tasks

This part of the manual (Part 1) provides step-by-step procedures for performing some essential and common installation tasks: installing the Domain/OS operating system from media and across the network on a new or initialized node, and installing other software products after the operating system is installed. Part 2 describes more advanced and optional installation tasks. The Part 1 contents include:

Chapter 1 how to install the Domain/OS operating system from cartridge- or magnetic-tape distribution media on a new node or on an existing node whose disk you want to initialize.

Chapter 2 how to install the Domain/OS operating system across the network from a source area on disk, called an Authorized Area, to a new node or an existing node whose disk you want to initialize.

Chapter 3 a detailed description of the **invol** (initialize volume) utility, provided as an optional supplement to the instructions in Chapter 1 and 2.

Chapter 4 a detailed description of the **calendar** utility, provided as an optional supplement to the instructions in Chapter 1 and 2.

Chapter 5 how to load products — Domain/OS updates, optional products, patches, and Product Support Kits (PSKs) — from distribution media into an Authorized Area, and then configure and install products from the Authorized Area to other nodes.

The remainder of this introduction briefly describes some important aspects of the Domain installation model — things you need to know before you perform any of the Part 1 procedures. It also provides an overview of the installation tools. If you are responsible for managing the installation of Domain products in a network or anticipate installing products frequently, we recommend you read Chapters 10 and 11, which describe the concepts and the installation tools in more detail, in addition to this chapter.

The Authorized Area: A Key Concept

The concept of an **Authorized Area** is central to the Domain installation model. This section provides a brief description of Authorized Areas. For a detailed description, see Chapter 11.

What is an Authorized Area?

An Authorized Area is a directory on a disk volume or storage module that acts as a source area for product installations. Before you can install products on a node, you must first load them from distribution media into an Authorized Area. Products are then installed from the Authorized Area to other nodes in the network (or the Authorized Area node). In addition to containing the actual products, an Authorized Area contains the installation tools and other administrative objects required to manage the Authorized Area and the products it contains.

Four aspects of an Authorized Area are important to keep in mind:

- Products in an Authorized Area are not operational — they just sit there in dormant storage.

- Getting an operational configuration of a product on a node from distribution media is always a two-step process: first, you load the product from media into an Authorized Area; then you install an operational configuration of the product from the Authorized Area to one or more target nodes. You cannot install a product directly from media and bypass an Authorized Area.

- An Authorized Area is not a special construct. It is simply a directory, like any other Aegis or UNIX directory, with a number of subdirectories and objects contained in the subdirectories. Consequently, you can copy, move, and manipulate Authorized Areas using standard Aegis and UNIX commands (Chapter 9). The only required feature of an Authorized Area directory is that it contain a subdirectory named **install**, which in turn contains the other objects and

subdirectories. When we ask you to provide the name of an Authorized Area, be sure to give the pathname that contains the **install** subdirectory, not *authorized_area_pathname*/**install**.

- Every node contains an **install** subdirectory in its node entry directory (//*node_name*/**install**). This directory contains objects created and used by the installation programs (see "install" in Chapter 10). Do not confuse this directory with the **install** subdirectory of an Authorized Area. Although a node's **install** subdirectory can double as the **install** subdirectory of an Authorized Area, the objects it contains and their functions are different.

What is the Purpose of an Authorized Area?

The primary purpose of an Authorized Area is to enable you to install products *across the network* — from one node to another — rather than perform the more tedious and time-consuming process of installing products from distribution media on every node. In theory, you only have to load a product from media once, into an Authorized Area. Thereafter, you can install the product across the network on other nodes from the Authorized Area.

Additionally, because an Authorized Area centralizes products and the installation tools in one location, it enables system administrators to more easily control, manage, and restrict the installation of products in a network.

Note that the ACLs (Access Control Lists) of an Authorized Area directory are initially *open*, meaning all users have read, write, and execute privileges. If greater security at your site is required, you should modify the ACLs of the Authorized Area and its objects to restrict access in the way most appropriate for your site.

How do Authorized Areas Relate to Distribution Media?

A set of product distribution media may contain a single product (Domain/OS, for example) or several products (a group of compilers, for example). In addition to and separate from the products, the distribution media contains a number of administrative objects used in an Authorized Area. The administrative objects reside on the first physical file — file 1 — on the first media volume.

The administrative objects include: the installation tools and their help files; online release notes and other release documentation for the products; and files that enable you to control the configurations of products that get installed.

The installation tools are currently released with the Domain/OS operating system and PSKs only. Formerly, the installation tools were released with all products. Therefore, older optional products that have not been re-released since approximately version 10.3 of Domain/OS may contain older versions of the installation tools on their media.

How is an Authorized Area Created?

You do not have to perform a special, separate procedure to create an Authorized Area: the process of installing Domain/OS from distribution media on a new or initialized node from distribution media (Chapter 1) automatically creates an Authorized Area on that node's disk. This Authorized Area contains the Domain/OS product, the installation tools, and all of the other administrative objects required in an Authorized Area. Once this initial Authorized Area is created, you can can load other products into it from media and install products from this Authorized Area across the network to other nodes.

If you want, you can create other Authorized Areas in your network by simply copying the initial Authorized Area (Chapter 9). Also, you can create a new Authorized Area when you load a Domain/OS update or a PSK from media onto an existing node that you don't want to initialize (Chapter 5).

How Do I Restore the Installation Tools from Media?

The installation tools are currently released with the Domain/OS operating system and PSKs only. You don't have to perform any special, separate procedure to restore them from the media into an Authorized Area.

As we already mentioned, the process of installing Domain/OS from distribution media on a new or initialized node from distribution media (Chapter 1) automatically creates an Authorized Area that contains the installation tools. Specifically, when you boot the node from media during this procedure prior to installing Domain/OS, the boot program restores the tools; the new tools are then used to restore and install the Domain/OS product.

Also, when you load a Domain/OS update or a PSK from media into an Authorized Area on an existing, non-initialized node (Chapter 5), the procedure tells you how to restore the installation tools prior to restoring the products.

What is a Product?

In the Domain installation model, a product is a set of objects (files and directories) assigned a name and version number recognized by the installation tools. For example, the name and version number of version 10.4 of Domain/OS (SR10.4) are **os** and **10.4**. You use this name and version number in conjunction with several of the installation tools.

Product Version Numbers and ISP Types

Most Domain products are released in two versions: a version that runs on workstations based on Motorola's 68000 series of microprocessors, and a version that runs on the Series 10000 workstation. Two versions are required because the Series 10000 workstation has a different ISP (Instruction Set Processor) type, called PRISM (Parallel Reduced Instruction Set Multiprocessor). We refer to the Series 10000 ISP type as a88k, and the Motorola 68000 ISP type as m68k.

The first character after the right most period in a product's version number usually identifies the product's ISP type. If the character is a digit, the product is for m68k machines. If the character is the letter "p", the product is for a88k (Series 10000) machines. For example, the version number of the m68k version of SR10.4 is **10.4**, whereas the a88k version has a version number of **10.4.p**.

Product Directories

In an Authorized Area, the objects that comprise a product reside in a directory named **ri.apollo.***product_name.version_number*. For example, the name of the SR10.4 product directory is **ri.apollo.os.10.4**. When you install a product, the installation tools copy the product's objects from its product directory in the Authorized Area to the target node and build an operational configuration of the product.

Product Release Indexes

Every product also contains a binary file called a **release index**. The release index defines configuration options for the product and is used internally by several of the installation tools to control the installation of the product. In an Authorized Area, a product's release index resides in the product directory and has the same name as the product directory (**ri.apollo.***product_name.version_number*).

Types of Products

The installation tools handle all products in the same way. However, in practice it is useful to group Domain products into four classes:

- The Domain/OS operating system
- Optional products
- Product Support Kits (PSKs)
- Patches (bug fixes)

A PSK is a subset of the Domain/OS file set that provides support for a new hardware platform or peripheral device, or that provides new functionality. PSKs are released as separate products and installed *on top of* a particular version of Domain/OS. Similarly, each patch is released and handled as an individual product with its own product name, version number, and product directory. Unless otherwise stated, whenever we use the term "product" in this manual, the term applies to all four types of products.

Types of Distribution Media

Domain products are potentially distributed on three types of distribution media: cartridge tape, magnetic tape, and floppy disks. As of SR10.4, Domain/OS is distributed on cartridge tape and magnetic tape only, with a cartridge tape boot volume for both media types. Prior to SR10.4, Domain/OS was distributed on floppy disks as well as cartridge and magnetic tape, with floppy disk boot volumes for the magnetic tape and floppy product media. Patches are distributed on cartridge tape only. Optional products may be distributed on all three media types; it depends on the particular product.

On all media types, the products are in **wbak** format. **wbak** format is a format created by the **wbak** command, which HP uses to write Domain products on the media.

The Installation Tools: An Overview

This section provides a brief overview of the Domain installation tools. For a complete description, see Chapter 10.

Core Tools

The set of installation tools contains five core tools: **distaa**, **config**, **install**, **cfgsa**, and **mrgri**. Each of these tools handles a single, discrete aspect of the installation process. The **distaa**, **install**, and **mrgri** tools are strictly command–line tools with no interactive interface. The **config** and **cfgsa** tools have a non–graphical, interactive interface as well as command-line options.

distaa The **distaa** (distribute Authorize Area) tool loads products from distribution media into an Authorized Area on disk. **distaa** does not install products; it only places the products in the Authorized Area in preparation for subsequent installation with other tools. You can optionally use an HP-supplied **selection file** with **distaa** to select a single product from the media for loading (when the media contains more than one product). You can also use an HP-supplied or user–created selection file to load a subset of a particular product, rather than the entire product. **distaa** loads products only — it does not load any of the administrative objects, such as release documentation, associated with a product or required in an Authorized Area. You must manually restore these objects with the **rbak** command. The **minst** tool, which provides an alternate means of loading products, automatically loads the administrative objects.

config With most Domain software products, you are given configuration options that enable you to specify precisely how you want a product or group of products installed. The **config** tool is used to define a configuration of one or more products resident in an Authorized Area prior to installing the product(s). For a given product, **config** presents you with a series of configuration questions defined in the product's release index file. The most common configuration questions ask you which optional subcomponents of a product you want to install, whether you want to install the subcomponents as links to other nodes or as local copies, and to specify other operational aspects of the product. **config** saves the configuration in a **configuration file**. To actually install the product configuration, you supply the pathname of the configuration file as a command line argument to the **install** tool. Using **config** is optional; every product ships with at least one predefined configuration file that you can use with **install** instead of creating your own.

install The **install** tool installs an operational configuration of one or more products from an Authorized Area to one or more nodes. The configuration of products installed is defined in one or more configuration files, whose pathnames you supply as command–line arguments to **install**. You create the configuration files beforehand with the **config** tool, or you can use default configuration files shipped with the products. You can also use **install** to deinstall (remove) an installed product; to deinstall specified subcomponents of an installed product; and to reset the permissions of an installed product to their original settings.

cfgsa The **cfgsa** (configure system administrator) tool has two uses. First, you can use **cfgsa** to restrict the configuration choices presented to a user when a user configures a product with the **config** tool. The restricted configuration also overrides any configuration that a user defines prior to the restrictions. Second, you can use **cfgsa** to create a selection file that defines a subset of product. You can then use this selection file with **distaa** to load the subset of the product from distribution media into an Authorized Area. The use of **cfgsa** is entirely optional. It is intended primarily for system administrators who want to exercise greater control over the installation of products and save space in their Authorized Area.

mrgri The **mrgri** (merge release index) tool merges two products in an Authorized Area into a single product. With **mrgri** you can merge a patch with the product it patches or merge a PSK with Domain/OS. You can then install the merged product as a single entity, rather than install the constituent products separately. **mrgri** also enables you to merge the version of a product that runs on the Series 10000 workstation (the a88k version) with the version that runs on m68k workstations. The merged product can run on workstations of either ISP type. The use of **mrgri** is entirely optional. It is intended primarily for system administrators.

Layered Tools

We also provide two other interactive tools — **install++** and **minst** — that are *layered on top* of the core tools; that is, they transparently invoke some of the basic tools to combine discrete installation tasks into a single process. The layered tools provide an optional approach to using some of the core tools individually. They are more interactive than the core tools and require less prior knowledge to run, but are less flexible.

install++ The **install++** tool combines the functionality of the **config** and **install** tools into a single process. It optionally invokes the interactive **config** program, enabling you to define or modify a configuration of one or more products in an Authorized Area. It then invokes the **install** program to install the product configuration from the Authorized Area to one or more nodes. Like the **install** tool, **install++** can also deinstall an entire product, deinstall product subcomponents, and reset the permissions of an installed product. Functionally, using **install++** is nearly equivalent to using **config** and **install** separately. **install++** is more convenient for one-time installations of a product, especially since it allows you to *not* save the product configuration you define. **install** is better suited for incorporation in scripts (because it not interactive) and for running installations remotely on other nodes. Also, **install** allows you to separate the installation process in time from the configuration process, which many administrators find desirable.

minst The **minst** (media install) tool is an interactive program that combines the functionality of the **distaa** and **install++** tools into a single process. It is used to load products from distribution media into an Authorized Area, and then to optionally configure and install the products from the Authorized Area to one or more nodes. The most notable feature of **minst** is that it leads you through the loading and installation processes step-by-step, providing detailed instructions and numerous prompts. Also, **minst** performs certain tasks — such as restoring the installation tools, release documentation and other administrative objects from the media — automatically. With the other tools, you do these tasks manually with standard Aegis or UNIX commands. Therefore, **minst** is well suited for new and infrequent users of the installation tools and for the first-time installation of Domain/OS from media, a fairly complex task. Many users, however, prefer the flexibility and economy of the core tools and separating the loading and installation processes.

Two Important Terms: Load and Install

Throughout this manual we use the terms *load* and *install* in a particular way. To *load* a product means to restore the product from distribution media into an Authorized Area on a storage device, using the **distaa** or **minst** tool. To *install* a product means to install an operational configuration of a product from an Authorized Area to one or more target nodes, using the **install**, **install++**, or **minst** tool. In a few exceptional cases (such as in the titles of Chapters 1 and 2 and the part titles), for brevity, we've used the term *install* to apply to the overall process of loading and installing a product.

1

Installing Domain/OS from Media

How to install Domain/OS from cartridge- or magnetic-tape distribution media on a new node or an existing node whose disk you want to initialize with the **invol** utility

This chapter describes how to install the Domain/OS operating system from cartridge tape and magnetic tape distribution media on a new node or an existing node whose disk you want to initialize. The target of the installation can be a workstation or a Domain Server Processor (DSP). A workstation is a processor equipped with a monitor and a keyboard; a DSP has no monitor or keyboard.

If you are installing Domain/OS from cartridge tape, you can use this procedure to install any SR10.*x* version of Domain/OS. If you are installing from magnetic tape, you can use this procedure to install version SR10.4 or later. Pre–SR10.4 versions of Domain/OS distributed on magnetic tape required booting from floppy disks, a process not described in this manual. SR10.4 and later versions of Domain/OS distributed on magnetic tape boot from a cartridge–tape boot volume, a process this manual does describe.

Creation of an Authorized Area

In addition to installing an operational configuration of Domain/OS, the procedure creates an Authorized Area at the target node's node entry directory (*//node_name*). This Authorized Area contains the Domain/OS product, the installation tools, and other administrative objects (the help files for the installation tools, predefined Domain/OS selection and override files, and the Domain/OS release documentation).

When Do I Use This Procedure?

Stand-Alone Nodes

If the node is a stand-alone node (not in a network), use this procedure to install Domain/OS for the first time. If the stand-alone node is already running Domain/OS (SR10.*x*), also use this procedure to install any version of Domain/OS that requires initializing the node's disk. If disk initialization is not required, use the load, configure, and install procedures in Chapter 5.

Networked Environments

In a network, you can use this procedure to install Domain/OS on any node to be initialized. However, in theory, you only have to use this procedure once — for the first node in a network that you are setting up for the first time. Once you've successfully installed Domain/OS on the first node, you can install Domain/OS *across the network* from the Authorized Area on the first node to other nodes, rather than installing from media (see Chapter 2). Network installs are faster and easier.

In an existing, SR10.*x*-based network, you can install a new version of Domain/OS that requires disk initialization in two ways. You can use the procedure in this chapter to install the version on one node, and then use network installs (Chapter 2) to install the version on other initialized nodes. Alternately, you can load the new version from media into a new or existing Authorized Area without initializing the Authorized Area node (using the load procedure in Chapter 5); install the version across the network from the Authorized Area node to other initialized nodes; and eventually, to install the new version on the Authorized Area node, copy the Authorized Area to another node and install from that node to the initial Authorized Area node.

To load and install a new version of Domain/OS that does *not* require disk initialization in an existing SR10.*x* network, use the load, configure, and install procedures in Chapter 5.

SR10.4 and Disk Initialization

Version 10.4 of Domain/OS (SR10.4) does not require initializing the node's disk. But we recommend disk initialization to take advantage of performance enhancements and new functionality, such as disk quotas.

Workstations versus DSPs

Loading and installing Domain/OS from media on an initialized node is easier if the node is a workstation rather than a DSP. To initialize and install Domain/OS on a DSP, we recommend that you do the installation over the network (Chapter 2). If you want to set up an Authorized Area on a DSP, you can install Domain/OS on a workstation, perform a network install of Domain/OS from the Authorized Area on the workstation to the DSP, and then copy or move the Authorized Area to the DSP.

Summary of Procedure

This procedure is broken down into seven major steps:

Step 1 Prepare for the installation: prerequisites and things you should check before you being the installation.

Step 2 Back up any files or directories that you want to save, if you are installing Domain/OS on an existing node.

Step 3 Initialize the node's disk using the **invol** utility, and boot the node from the cartridge–tape boot volume. As of SR10.4, a cartridge-tape boot volume is used for both cartridge- and magnetic-tape distribution media.

Step 4 Load Domain/OS from media into an Authorized Area and install Domain/OS on the target node, using the **minst** tool.

Step 5 Reboot the node to make the new operating system operational.

Step 6 Restore any files or directories that you backed up before initializing the disk.

Step 7 Perform post–install administrative tasks, such as creating a registry database, as required by the specific needs of the node and your site.

Step 1. Prepare for the Installation

This section discusses prerequisites and things you should do and check before you begin the actual installation process.

- Install hardware.

  The procedure in this chapter assumes that you've already physically installed all requisite hardware. If you have not already done so, physically install the workstation or DSP and all applicable peripheral devices, such as a magnetic-tape drive. Use the instructions in the appropriate unpacking and installation manuals that you receive with the hardware. The node does not have to be connected to the network.

  The workstation or DSP must have a cartridge-tape drive to perform this procedure. This is so even if you are installing Domain/OS from magnetic tape, since (as of SR10.4) you must boot the node from a cartridge-tape boot volume.

  If you are installing Domain/OS on a DSP, also note:

  - The DSP must have an internal Winchester disk, even if the DSP is also connected to an SMD drive.
  - You must attach a dumb terminal to serial I/O line number 1 (SI01), an RS-232 port, of the DSP to act as the system console. Alternately, you can run a null-modem cable from SI01 of the DSP to an RS-232 port of a workstation, and run a terminal emulator on the port at the workstation. You can use the Apollo terminal emulator **emt** or any program that can emulate a dumb terminal.

    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.

    See the DSP's operating guide for more information about connecting a terminal or workstation to a DSP.

- Check machine-type compatibility.

  Versions 10.4 and greater of Domain/OS do not support all Domain machine types. Make sure the target workstation or DSP is a type that is supported. Table 1-1 shows the machine types supported by the first release of SR10.4. Later versions of Domain/OS may support additional machine types. Check the release notes of later versions for machine compatibility information. Table 1-2 shows the machine types *not* supported by SR10.4 and later versions of Domain/OS. Note that SR10.4 does not support any of the machine types that support the use of storage modules.

Table 1-1 **Machine Types Supported by SR10.4**

/sau #	Machine Type
/sau7	DN3500, DN3550, DN4000, DN4500, DSP3500, DSP3550, DSP4000, DSP4500
/sau8	DN3000, DSP3000
/sau9	DN2500
/sau10	DN10000
/sau11	9000/425S, 9000/425T
/sau12	9000/400S, 9000/400T
/sau14	DN5500

Table 1-2 **Machine Types Not Supported by SR10.4 and Later**

/sau #	Machine Type
/sau2	DN300, DN320, DN330
/sau3	DSP80, DSP80A, DSP90
/sau4	DSP160, DN460, DN660
/sau5	DN550, DN560, DN570, DN580, DN590
/sau6	DSP500-T, DN560-T, DN570-T, DN580-T, DN590-T

The **sau** (stand alone utility) directories listed in the tables are directories that contain machine-specific utilities. They are part of the Domain/OS file set. You can identify the **sau** number of an existing node with the **bldt** command (**/com/bldt** or **/usr/apollo/bin/bldt**). **bldt** displays the **sau** number of the machine in parentheses following the text, `Domain/OS kernel`. You can display the machine type (for example, DN3500) of an existing node with the command **/com/netstat -config** or **/etc/nodestat -config**.

- Determine the number of logical volumes on the disk.

  If the target node is an existing (versus new) node, we recommend that you determine the number of logical volumes on the node's disk. To do this, shutdown and use option 5 of the **invol** program. 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 boot volume and preserve other existing volumes, using option 2 or 3 of the **invol** program.

- Read the *Domain System Software Release Notes*.

  Pay particular attention to the installation chapter of the release notes. It contains Domain/OS-specific installation information that is not addressed in this manual and documents any changes to and errors in this manual.

  Notably, the installation chapter describes different configurations of Domain/OS (for example, an **aegis_bsd4.3_medium** configuration) that you can load and install. The configurations differ in the operating-system environments provided (some combination of Aegis, SysV, and BSD4.3) and their sizes (small, medium, or large). The release notes list the subcomponents included in each configuration and their disk space requirements.

  The **minst** program, which you'll use to load and install Domain/OS, requires you to select a configuration. (**minst** calls these configurations **templates**). Although **minst** pauses so you can read the release notes online and choose a template at that time, we recommend that you select one now. Make sure the target disk is large enough for the configuration you select. Note that you need at least 12 MB of free disk space in addition to the actual size of the template to allow for space required by the installation processes while they execute. On an existing node, you can determine the size of the disk volume with the **/com/lvolfs** or **/etc/nodestat -config** command.

Step 2. Back Up Files

If you are installing Domain/OS on a new workstation or DSP, skip this section and proceed to Step 3.

Initializing a disk destroys all data on it. Therefore, before you initialize the disk you should back up any files and directories you want to save. Typically, you back up user directories and any site- or node-specific configuration and database files. You can then restore the archived objects to the node after you install Domain/OS.

You can back up objects to removable media (cartridge tape, magnetic tape, or floppy disk) using the **wbak** command. Or you can simply copy the objects to another node in the network using the Aegis **cpt** and **cpf** commands, or the UNIX **cp** command. Copying files to another node is faster, but the node's disk must have sufficient space. Also, if you copy files, be sure to preserve any desired attributes of the files — such as ACLs and date and time stamps— by using the appropriate command-line options.

Do *not* back up and restore any system directories like **/sys** or **/com** — you want to use the new versions of these that get installed with Domain/OS.

The following list identifies objects that are commonly backed up and provides sample **wbak** command lines for doing so. The list is not comprehensive — the objects you back up on any given node ultimately depends on the node, the node's user, and the site. Also, the **wbak** command lines are suggestive only. For more information about **wbak** and its command-line options, see the Aegis, SysV, or BSD *Command Reference* or the online manual pages for **wbak.**

In all of the **wbak** command lines, the variable *device* is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape. Also, use the command-line option **–f 1** instead of **–f end** if the backup volume is blank. **wbak** resides in the **/com** and **/usr/apollo/bin** directories. You may have to be logged in as **root**, depending on the ACLs (Access Control Lists) of the objects to be backed up.

- Back up all user directories and files, using this command line:

  wbak -dev *device* **-l -nhi -f end -fid user_trees** *pathname1 ... pathnameN*

  where

  pathname1 ... pathnameN are the pathnames of user directories and files. We recommend that you use full, not relative, pathnames and designate the node entry directory with a single slash, rather than with two slashes and the node name (**/my_dir** rather than *//node_name*/**my_dir** or **my_dir**, for example). If you use two slashes, the restore (**rbak**) commands we provide later in this chapter won't work properly if the node is cataloged under a different name. If you use relative pathnames, the **rbak** commands we provide restore the objects to the current directory, which can complicate the process.

- If the node is an **ns_helper** database site, back up the **ns_helper** database files.

  To determine if the node is an **ns_helper** site, run **/etc/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.

  Instead of backing up the **ns_helper** files, you can reinitialize the files with **edns** after you install Domain/OS. Reinitializing the files is not that difficult because **edns** can reconstruct the databases with little intervention on your part.

  If you choose to back up the **ns_helper** files, first stop **ns_helper** using the **sigp** (Aegis) or the **kill** (UNIX) command. Then back up the files, using this command line:

  wbak -dev *device* **-l -nhi -f end -fid ns_helper**
  /sys/ns/helper_data/'ns_helper.?*'
  /sys/node_data/system_logs/ns_helper.err_log

- If the node is the master registry site, back up the registry database and the **/etc/passwd**, **/etc/group**, and **/etc/org** files:

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

  Before you back up the registry, make sure you stop or put the registry servers in maintenance mode using the **rgy_admin** tool. Otherwise, some files will be busy and will not be backed up properly.

- Also consider backing up the following types of files:
  - User-modified startup files in the directories **/sys/node_data** and **/sys/dm**. If other nodes use the 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 node. When you install Domain/OS, you get a new set of startup files with appropriate defaults. However, the information from the old startup files may help you get the node up and running the way you like it more quickly.
  - Printer configuration files in the directory **/sys/print**, if the node is attached to a printer.
  - Customized font files, if any, in the directory **/sys/dm/fonts**.
  - Local extensions, if any, to the standard UNIX commands in the directories **/usr/local**, **/bin**, or **/com**.
  - UNIX system configuration files, such as **/etc/rc**, **/usr/lib/crontab**, and **uucp** configuration files, if the node is a **uucp** administrative site. When you install Domain/OS, you get a new set of these files with appropriate defaults.
  - TCP/IP administrative files, if the node is running TCP/IP: **/etc/hosts**, **/etc/hosts.equiv**, **/etc/networks**, **/etc/gateways**.

  To back up these files, you can use the following **wbak** command line:

  wbak –dev *device* **–l –nhi –f end –fid** *file_id* *pathname1 ... pathnameN*

  where

  file_id is a character string of your choice that labels the physical file on the backup media in which the groups of objects are archived. You can use this label with **rbak** to easily restore the same group of objects.

  pathname1 ... pathnameN are the pathnames of the files or directories you want to back up.

Step 3. Initialize the Disk and Boot From Media

Perform the following steps to initialize and boot the target node in preparation for installing Domain/OS. You initialize the target's disk using the **invol** program at the Mnemonic Debugger (MD) level. You then boot the node from the cartridge-tape boot volume. If you are installing Domain/OS (versions 10.4 and greater) from magnetic-tape distribution media, you also boot from the cartridge-tape boot volume provided with the OS distribution media.

1 Make sure the node is in NORMAL (versus SERVICE) mode.

The mode is usually controlled by a toggle switch on the back panel of the CPU. See the node's operating guide for more information.

2 Shut down the node to the Mnemonic Debugger (MD) level.

- If the node is a workstation running the Display Manager (DM), press the < CMD > key and enter **shut** at the DM prompt:

  `Command:` **shut**

  You can also enter **shut** at the DM login prompt.

- If the node is a workstation that is not running the DM, log in as **root** and issue the appropriate UNIX **shutdown** command at a shell prompt:

  /etc/shutdown -y -g0 -i5 (SysV UNIX)

  /etc/shutdown -h now (BSD UNIX)

- If the target node is a DSP, enter **shut** in the **emt** or **crp** window.

In all cases, wait for the message `SHUTDOWN SUCCESSFUL` and for the Mnemonic Debugger (MD) prompt to appear. The prompt depends on the node firmware, but it always ends in a >.

3 Reset the node.

To reset the node, enter the command **RE** (or **RE W** for Series 10000 workstations), followed by a carriage return at the next prompt. For example,

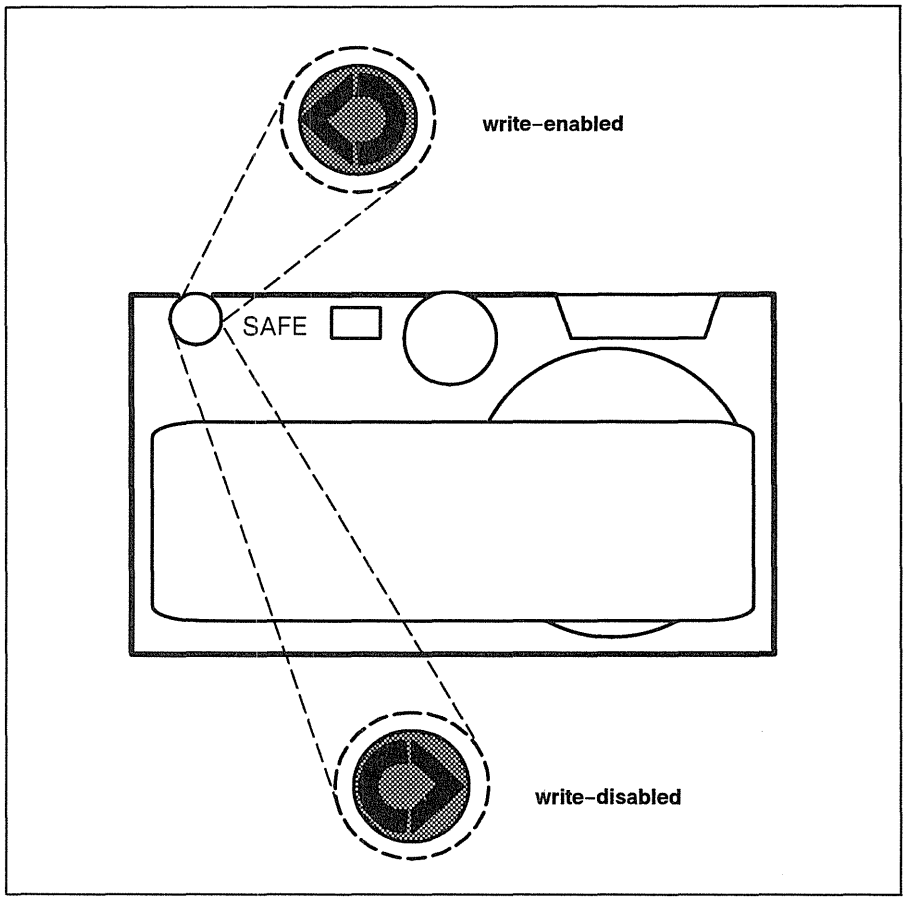
```
> RE <RETURN>
> <RETURN>
MD7C  Rev. 6.0, 1989/05/25.15:18:03
>
```

If the MD message does not appear after you press <RETURN> the second time, repeat the process.

4 Write–disable and physically mount the cartridge boot tape.

The cartridge-tape boot volume is labeled **crtg_std_sfw_boot_1**. Make sure the boot tape is write–disabled. Then insert the tape in the drive.

To check that the tape is write–disabled, hold the tape cartridge so that the word "SAFE" embossed in the plastic casing is visible in the upper left corner. 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 so it points to "SAFE."



Write-disabling a Cartridge Tape

5 Select the cartridge-tape drive as the device from which to read.

To select the cartridge-tape drive on all nodes types except the Series 2500, enter the command

> **DI C**

On Series 2500 nodes, enter

> **DI T**

6 Run the interactive **calendar** program to set the node's date, time, and time zone.

To start **calendar**, enter the command

> **EX CALENDAR**

Respond to the series of prompts. See Chapter 4 for a detailed description of the prompts. Running **calendar** at this point ensures that the disk initialization program (**invol**) creates correct Unique Identifiers (UIDs) for the objects it creates on the disk.

7 When **calendar** completes, reset the node again:

> **RE** <RETURN> [or **RE W** for Series 10000 workstations]
> <RETURN>
```
MD7C  Rev. 6.0, 1989/05/25.15:18:03
>
```

8 Select the cartridge-tape drive again:

> **DI C** [or **DI T** for Series 2500 nodes]

9 Run the interactive **invol** program to initialize the node's disk and set the size of the OS paging file.

To start **invol**, enter the command

> **EX INVOL**

The main **invol** menu appears. If you are unfamiliar with the **invol** program, turn now to Chapter 3, which provides a detailed description of running **invol** in this context. When you finish with Chapter 3, go to step 10. If you don't use Chapter 3, use the following summarized account:

1 On the **invol** menu, select option 1 (`initialize virgin physical volume`) to initialize the entire disk (the typical response). Or, if you are initializing an existing disk that 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`).

2 Respond to the subsequent prompts and wait for the disk initialization to complete. When asked,

`Anything more to do?`

enter **y**.

3 Select option 8 (`create or modify an os paging file`) on the **invol** menu, and respond to the subsequent prompts. Unless you have special paging size requirements, accept the default paging size. As of SR10.4, Domain/OS dynamically allocates paging file disk blocks as needed. The size you specify with **invol** is the initial size of the paging file, which then *grows* or *shrinks* as required.

4 When asked,

```
Anything more to do?
```

enter **n**.

10 Reset the node again:

```
> RE <RETURN>                [or RE W for Series 10000 workstations]
> <RETURN>
MD7C  Rev. 6.0, 1989/05/25.15:18:03
>
```

11 Select the cartridge-tape drive again:

```
> DI C                [or DI T for Series 2500 nodes]
```

12 Run the **calendar** program again:

```
> EX CALENDAR
```

Respond to the series of prompts. See Chapter 4 for a detailed description of the prompts. Running **calendar** at this point sets the correct time for objects subsequently installed on the disk.

13 Boot the node from the cartridge-tape boot volume.

To boot:

1 Enter the command

```
> EX DOMAIN_OS
```

2 After a few messages, this confirmation prompt appears:

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

Answer **y** in response.

The boot program then loads a subset of the operating system, enough to run the DM or Server Process Manager (SPM) process. A series of messages appears during this process.

3 When the) prompt appears, 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. Several messages appear during this process. When the boot completes, remove the boot tape from the drive.

The node is now named //**node_***node_id*, where *node_id* is the hexadecimal node ID assigned to the node during the manufacturing process.

The boot procedure creates a minimal Authorized Area on the node. The Authorized Area is created at the node's entry directory (/). It contains the following components: the installation tools (in the **/install/tools** directory), the help files for the installation tools (**/install/help** directory), the Domain/OS release documentation (**/install/doc/apollo** directory), the Domain/OS selection and override files (**/install/templates/apollo/os.v.10.4** directory), and the media's TOC file (**/install/toc** directory). (See Chapter 11 for more information about these components.) The Domain/OS product is not loaded into the Authorized Area at this time.

Step 4. Load and Install Domain/OS

This section describes how to load and install an operational configuration of Domain/OS from cartridge or magnetic tape after you've initialized and booted the node. The procedure instructs you to use the interactive **minst** tool. Instead of **minst**, you can use the **distaa**, **config**, and **install** tools separately to load, configure, and install Domain/OS, respectively. We've chosen **minst** because its easier to use in this situation.

1 Log in as **user**; press < RETURN > at the password prompt.

The **minst** program, identifiable by the `MINST>` prompt, starts automatically upon login. **minst** starts automatically whenever you log in after booting from distribution media.

If the system does not accept the **user** login, and the node is physically connected to a network, disconnect the node and try logging in again. A network registry that excludes **user** as a valid login will override the local **user** login created by the boot process.

2 Respond to the **minst** prompts.

For the most part, running **minst** is self-explanatory; it displays detailed, explicit instructions and prompts that lead you through the entire process of loading and installing a product step-by–step. Therefore, we don't provide detailed information about the individual **minst** prompts here. For more information, see "minst" in Chapter 10.

We do provide a few notes specific to installing Domain/OS that you may find useful:

- **minst** prompts you to enter the pathname of an Authorized Area. It provides the node's entry directory (//**node_***node_id*, where *node_id* is the system-supplied node ID) as the default response. If you want to create the Authorized Area at some other directory that does not yet exist (//**node_***node_id*/**aa**, for example), you must first create a shell (by pressing the < CMD > key and entering **cp /com/sh** at the `Command:` prompt) and then create the desired directory (by entering **wd** *pathname* at the shell prompt). Remember, however, that the boot process already created an Authorized Area (minus the Domain/OS product) at //**node_***node_id*.

Therefore, we recommend that you accept the default; and if you want the Authorized Area at another directory, copy or move the node-entry Authorized Area to the desired directory after you install Domain/OS. (See "Copying or Moving an Authorized Area" in Chapter 9).

- If you run **minst** in expert mode, be sure to request that Domain/OS be installed with hard links to the Authorized Area. Otherwise, the space required will be almost twice that of the selected OS configuration. Novice mode installs the OS with hard links by default.

  Note that if you subsequently modify any Domain/OS object (such as a user-modifiable configuration file) that is hard linked to the Authorized Area, the modifications also apply to the object "in" the Authorized Area as well. (A hard link is just another reference to the same physical object.) When Domain/OS is subsequently installed from this Authorized Area to other nodes, the modified object — not the original version — gets installed. So use caution when modifying any hard-linked Domain/OS objects.

- When asked to enter the pathname of the installation target, accept the default (//**node_***node_id*).

- When **minst** prompts you to read the on-line Domain/OS release notes, read at least the installation chapter of the release notes and select which *template* (**bsd4.3_medium**, for example) you want to load into the Authorized Area. **minst** asks you to specify this choice when you finish reading the release notes. Each template is a different configuration of Domain/OS. The configurations vary in the operating system environments they include — some combination of Apollo's proprietary Aegis environment, Berkeley's Software Distribution 4.3 of UNIX (bsd4.3), and AT&T's System V Release 3 of UNIX (sys5.3) — and the size (small, medium, or large) of each of these environments. The release notes list the subcomponents included in each template and their disk space requirements. Make sure you choose a template that fits on the disk. Note that you need at least 12 MB of free disk space *in addition to* the actual size of the template. The extra 12 MB allows for space required by the installation processes while they execute.

- If there is insufficient disk space for the Domain/OS template you select, **minst** eventually displays a warning message and asks if you wish to continue. If this happens, answer **no. minst** stops execution and the transcript pad is closed. Create a new shell by entering **cp /com/sh** at the DM `Command:` prompt. Then rerun **minst** by entering **/install/tools/minst** at the shell prompt. This time, select an appropriately sized template.
- If one of the Domain/OS distribution tapes does not contain any objects that are part of the template you select, **minst** does not prompt you to insert that particular tape volume in the drive.
- If you are running novice mode or running expert mode with the default installation configuration selected, **minst** asks the following question when it finishes loading Domain/OS from the tape into the Authorized Area:

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

  Enter **s** (select) in response to this prompt. This selects Domain/OS for installation and begins the installation process.
- If for some reason you quit **minst** or **minst** aborts before the installation is complete, rerun **minst** by entering the command **/install/tools/minst** at a shell prompt. (If necessary, you can create a new shell by entering **cp /com/sh** at the DM `Command:` prompt.) Repeat the entire **minst** procedure. You do not need to initialize the disk again before you rerun **minst**.

3 Check for error and warning messages.

When **minst** completes execution, check the messages displayed by **minst** for any errors or warnings. Errors messages are prefixed with the label `ERROR:` and warning messages with the label `WARNING:`. You can easily locate the messages by searching the transcript for these character strings. Chapter 12 explains some common error and warning messages potentially displayed during the installation phase of **minst**. If you find errors, rectify the error conditions and, if necessary, rerun **minst** by executing the command **/install/tools/minst**.

Step 5. Reboot the Node

Once you successfully install Domain/OS, you must shut down the node and boot it from its own disk. This causes the node to begin running the newly installed operating system.

1 Shut down the node.

If the node is a workstation running the DM, press the < CMD > key and enter **shut** at the DM prompt:

Command: **shut**

If the target node is a DSP, enter **shut** in the **emt** or **crp** window.

Wait for the message `SHUTDOWN SUCCESSFUL` and for the Mnemonic Debugger (MD) prompt to appear. The prompt depends on the node firmware, but it always ends in a >.

2 Reset the node.

To reset the node, enter the command **RE** (**RE W** for Series 10000 workstations), followed by a carriage return at the next prompt. For example,

```
> RE <RETURN>
> <RETURN>
MD7C  Rev. 6.0, 1989/05/25.15:18:03
>
```

If the MD message does not appear after you press < RETURN > the second time, repeat the process.

3 Boot the node.

Enter the command

> EX DOMAIN_OS

If you are at a workstation, after a series of messages, the DM is started and the login prompt appears. If you are at a DSP, the SPM is started. The node is now running the newly installed version of Domain/OS.

Step 6. Restore Files

If you installed Domain/OS on a new node, skip this step. If you installed Domain/OS on an existing node and backed up objects before you initialized the disk, restore them now to their original location.

If you backed up the objects to removable media using the **wbak** command lines in "Step 2. Back Up Files," you can use the following **rbak** command lines to restore them. For a complete description of **rbak**, see the Aegis, SysV, or BSD *Command Reference* or the online manual pages for **rbak**.

In all of the **rbak** command lines, the variable *device* is **ct0** for cartridge tape, **f0** for floppy disk, or **m0** for magnetic tape. **rbak** resides in the **/com** and **/usr/apollo/bin** directories. You may have to be logged in as **root**, depending on the ACLs (Access Control Lists) of the directory into which you are restoring objects. Also note that if you backed up the objects using relative, not full, pathnames, the objects are restored to the current directory.

- Restore user directories and files:

  rbak -dev *device* **-l -sacl -pdt -force -du -fid user_trees -all**

- Restore **ns_helper** database files:

  rbak -dev *device* **-l -sacl -pdt -force -du -fid ns_helper -all**

- Restore the master registry database and the **/etc/passwd**, **/etc/group**, and **/etc/org** files:

  rbak -dev *device* **-l -sacl -pdt -force -du -fid registry -all**

- Restore any other objects you backed up, using a command line of the form:

  rbak -dev *device* **-l -sacl -pdt -force -du -fid** *file_id* **-all**

  where *file_id* is the file identifier that you specified with the **-fid** option when you backed up the objects with **wbak**.

Step 7. Perform Administrative Tasks

In this section, we list some administrative tasks you should consider doing after you successfully install Domain/OS. Where necessary, we provide cross-references to the manuals that describe these tasks in detail. This list is suggestive only: the administrative tasks you need to perform ultimately depends on the specific requirements of the node and your site.

- Physically connect the node to the network, if it isn't currently connected.
- Catalog the node.

  As a result of disk initialization, the node is now named //**node_***node_id*, where *node_id* is the system-supplied, hexadecimal node ID. Therefore, you should now use the following commands to catalog the node with a name of your choice and add the name to the master network root directory (the first command); and update the node's local registry with the names of other nodes on the network (the second command).

  ctnode *node_name node_id* **–l –r –root**

  ctnode –l –update

  where

  node_name is a node name of your choice; this becomes the name of the target node's entry directory (*//node_name*). Do not proceed the name with any slashes.

  node_id is the system–supplied node ID of the target node. You can display this ID with the **lcnode –me** or the **bldt** command. **bldt** displays the node ID in the form **** Node *net_id.node_id*. **** .

  For more information about cataloging nodes, see the *Domain/OS System Administration Guide*, the *Domain/OS System Administration Reference*, and the online manual page for **ctnode**.

- Recover disk space.

  Check how much free disk space remains, using the **lvolfs** or **df** command. If the disk has little free space, consider using the following methods to recover some disk space:

  - Run **/etc/salacl** or **/com/salacl** to merge duplicate Access Control Lists (ACLs) into a single copy and to delete unused ACLs.
  - Move the Authorized Area to another node with more disk space.

    See "Copying or Moving an Authorized Area" in Chapter 9.
  - Distribute the Authorized Area on more than one disk using links.

    See "Distributing an Authorized Area with Links" in Chapter 9.
  - Remove any unneeded objects from the Authorized Area.

    For example, you might delete any release documentation in the **/install/apollo/doc** directory for which you have hard copy, such as the Domain/OS release notes. You can delete objects in an Authorized Area using the Aegis **dlt** and **dlf** commands or the UNIX **rm** command. (See Chapter 11 for more information about the contents of an Authorized Area.)
  - Remove any unneeded **sau** directories from the Authorized Area.

    Each machine type has a corresponding **sau** (stand alone utility) directory that contains several machine-specific utilities. For example, the name of the **sau** directory for Series 2500 nodes is **/sau9**. The **sau** directories are part of the Domain/OS file set. When you run **minst** in either novice or expert mode, it unconditionally loads the **sau** directories for *all* machine types supported by that release into the Authorized Area (*authorized_area*/**install/ri.apollo.os.v.10.***x*/**sau***x*). Table1-1 lists the **sau** directories and the corresponding machine types provided with SR10.4.

    If the node is a stand-alone node, you can delete all **sau** directories except the one for that machine type. If the node is in a network, consider deleting those **sau** directories for machine types that are not and will not be in the network. You can delete **sau** directories in an Authorized Area using the Aegis **dlt** command or the UNIX **rm** command.

    sau directories are the only product subcomponents that you should remove from an Authorized Area in this way. But note that the Domain/OS configuration question (displayed by **config**) that asks the user to select which **sau** directories to install will still show and allow the selection of the removed **sau** directories.

- Delete the entire Authorized Area.

  Deleting the entire Authorized Area is a radical step. Generally, it is advisable only when you have another Authorized Area elsewhere in the network that at least contains the same version of Domain/OS and the same version of the installation tools. Single-node users or very small sites with limited disk space that do not anticipate installing products frequently might also consider deleting the Authorized Area.

  If you choose to delete the Authorized Area, do so only after you finish loading and installing all software, including optional products, on all nodes for the time being. If you delete the Authorized Area on a stand-alone node or delete the only Authorized Area in your network, you cannot install any products until you create a new Authorized Area and load products from media into it.

  If Domain/OS or any optional products are installed on the Authorized Area node with hard links to the Authorized Area, you can still delete the Authorized Area. However, the space saved is minimal.

  To delete the Authorized Area, use the procedure "Removing an Authorized Area" in Chapter 9.

- If the node is the first Domain/OS node in a new network, create and configure a new registry database and then start the registry server processes.

  See *Administering the Domain/OS Registry*. (Note: you create a new registry database with the *authorized_area*/**install/tools/rgy_create** utility. The "Creating a New Registry Database" section of the 015363-A00 version of *Administering the Domain/OS Registry* refers you to an earlier version of this manual for information about **rgy_create**. Instead, refer to the manual page for **rgy_create** in the *Registry* manual, which provides sufficient information.)

- If the node is a **ns_helper** site and you did not back up the **ns_helper** databases before you initialized the node, reinitialize the **ns_helper** databases with **edns** and start the **ns_helper** server processes.

  See the *Domain/OS System Administration Guide* and the *Domain/OS System Administration Reference*.

- Configure TCP/IP and HP VUE, if desired. TCP/IP must be running to use HP VUE.

  See *Configuring and Managing TCP/IP* and the *HP Visual User Environment Configuration Guide for Domain/OS Systems*.

2

Installing Domain/OS Across the Network

How to install Domain/OS across the network from an Authorized Area to a new node or a node whose disk you want to initialize with the **invol** utility

This chapter describes how to install any SR10.*x* version of the Domain/OS operating system *across the network* from an Authorized Area to another node in the network whose disk you want or need to initialize. At least one node in your network must be running Domain/OS (SR10.*x*). Also, an Authorized Area must exist that contains the version of Domain/OS you want to install. (Chapter 5 explains how to load products into an Authorized Area.)

The target of the installation can be an existing node or a new node that you are adding to the network. Also, the target can be a Winchester disk volume connected to a workstation or to a Domain Server Processor (DSP). A workstation is a processor equipped with a monitor and a keyboard; a DSP has no monitor or keyboard. (SR10.4 does not support any of the machine types that support storage module devices, so this manual does not include instructions for installing Domain/OS on storage modules.) The first section in this chapter describes how to install Domain/OS on a workstation; the second, on a DSP.

To install a version of Domain/OS that does *not* require disk initialization on a node already running Domain/OS (SR10.*x*), use the procedures in Chapter 5 rather than the procedures in this chapter. Version 10.4 of Domain/OS (SR10.4) does not require disk initialization. But we recommend initializing the disk prior to installation to take advantage of performance enhancements and new functionality, such as disk quotas.

Installing Domain/OS across the network on an initialized node is faster and easier than installing it from distribution media (Chapter 1). Also, unlike the media installation procedure, a network installation of Domain/OS does not create an Authorized Area on the target node, which is usually not wanted.

The procedure in this section describes how to install any SR10.*x* version of Domain/OS across the network from an Authorized Area to a new workstation or to an existing workstation whose disk you need or want to initialize. We refer to the workstation and its disk collectively as the **target node**. We refer to the target node's disk, or the logical volume on the disk on which you install Domain/OS, as the **target volume**.

Role of Partner Node

During this procedure, you sit at the target node and enter commands at its keyboard. You shut down the target node to the Mnemonic Debugger (MD) level and then select another node in the network, called the **partner node**. The partner node provides the various utilities and commands, such as **invol**, that you'll use throughout this procedure. Also, the partner node is the node from which you'll boot the target node after initializing its disk (a process called *booting diskless*) and the node on which you'll mount the target volume in order to install Domain/OS. The partner node must be running Domain/OS and meet other criteria, as detailed in "Step 1. Prepare for the Installation."

The Authorized Area containing the installation tools and the version of Domain/OS you want to install can reside on the partner node or on some other node in the network.

Summary of Procedure

This procedure is broken down into four major steps:

Step 1 Prepare for the installation: things to do and check before you begin the actual installation process.

Step 2 Prepare the target node for the installation: back up user files and directories, shut down, initialize the target volume using the partner node's **invol** program, boot the target node diskless from the partner, and mount the target volume on the partner node's file system.

Step 3 Configure and install Domain/OS from an Authorized Area to the target node, using the **install++** program.

Step 4 Perform final steps: unmount the target volume, shut down, boot the target node from its own boot volume, catalog the target node, restore any objects backed-up prior to initialization.

Step 1. Prepare for the Installation

This section discusses things you should do and check before you begin the actual process of installing Domain/OS on the workstation.

- Install hardware.

  The procedure assumes that you've already physically installed all requisite hardware. If you are installing Domain/OS across the network on a new node, physically install the workstation and any applicable peripheral devices. Connect the node to the network, if you have not already done so. Use the instructions in the appropriate unpacking and installation manuals that you receive with the hardware.

- Check machine-type compatibility.

  Version 10.4 of Domain/OS (SR10.4) and later versions do not support all Domain machine types. Make sure the target workstation is a type that is supported. See Tables 1-1 and 1-2 for the machine types supported and not supported, respectively, by SR10.4. Later versions of Domain/OS may support additional machine types. Check the release notes of later versions for machine compatibility information.

  The **sau** (stand alone utility) directories listed in the tables are directories that contain machine-specific utilities. You can identify the **sau** type of an existing node with the **bldt** command (**/com/bldt** or **/usr/apollo/bin/bldt**). **bldt** displays the **sau** number of the machine in parentheses following the text, `Domain/OS kernel`. You can display the machine type (for example, DN3500) of an existing node with the command **/com/netstat -config** or **/etc/nodestat -config**.

- Read the installation chapter of the *Domain System Software Release Notes*.

  The installation chapter contains Domain/OS-specific installation information that is not addressed in this manual and documents any changes to and errors in this manual. Notably, the installation chapter lists the subcomponents included in the different configurations of Domain/OS and their disk space requirements. The disk space requirements for any given Domain/OS installation vary, depending on which subcomponents you select for installation

(when you run **config** or **install++**) and which subcomponents you install as links.

However, we recommend that you make at least a rough assessment at this time of whether the target disk is large enough for the configuration you'd like to install. Note that you need at least 12 MB of free disk space in addition to the actual size of the configuration to allow for space required by the installation processes while they execute. On an existing node, you can determine the size of the disk volume with the **/com/lvolfs** or the **/etc/nodestat -config** command.

The release notes are provided in hard-copy and on-line. The pathname of the on-line release notes is *AA*/**install/doc/apollo/os.v.10.***x***__notes**, where *AA* is the pathname of the Authorized Area containing the version of Domain/OS to be installed, and *x* is the version number of Domain/OS.

- Determine the number of logical volumes on the disk.

  If the target node is an existing (versus new) node, we recommend that you determine the number of logical volumes on the node's disk. To do this, shutdown and use option 5 of the **invol** program. 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 boot volume and preserve other existing volumes, using option 2 or 3 of the **invol** program.

- Know the target node's name.

  If the target node is an existing node, identify its current name (the name of its node entry directory). The **bldt** command returns the node's name (*node_name*) in the form

  `**** Node` *net_id.node_id* `****` `"//`*node_name*`"`

- Select a partner node that satisfies the following criteria:

  ☐ The partner node must be running Domain/OS (SR10.*x*).

  We recommend that the partner is running the same version of Domain/OS that you are going to install. This ensures that the versions of the **invol** and **calendar** programs, which the partner provides to initialize the target's disk and to set the target's date and time, are consistent with the version of Domain/OS to be installed. The command **bldt** *//partner_node_name* displays the version of Domain/OS running on the partner.

  ☐ The partner node must be on the same local network as the target node.

  Use the commands **bldt** and **bldt** *//partner_node_name* to determine the network IDs of the target node and partner node, respectively. **bldt** displays the network ID *(net_id)* in the form

  `**** Node` *net_id.node_id* `****` `"//`*node_name*`"`

  ☐ The partner node must be running **netman**.

  To check this, enter one of the following commands and look for `netman` in the displayed list of processes:

  pst –n *//partner_node_name* (Aegis)

  ps –e –n *//partner_node_name* (SysV)

  ps –ax –n *//partner_node_name* (BSD)

  If **netman** is not running, you can start it by entering the command **cps /sys/net/netman** at the partner's DM command prompt or **/sys/net/netman &** at a shell prompt on the partner.

  ☐ The **/sau***x* directory for the target node's machine type must be installed on the partner node.

  See Table 1-1 for a list of the machine types supported by SR10.4 and their corresponding **/sau** directories. List the directories (using **ld** or **ls**) in the partner's node entry directory to ensure the appropriate **sau***x* directory exists.

- Know the node ID of the partner node.

  Identify and jot down the partner's node ID (the hexadecimal ID assigned to the node during the manufacturing process). The command **bldt** *//partner_node_name* displays the node ID (*node_id*) in the form

  `**** Node` *net_id.node_id* `****` `"//`*node_name*`"`

Step 2. Initialize, Boot, and Mount the Target

The following steps prepare the target node for the installation of Domain/OS. You back up any objects you want to save; shutdown the target; use the partner node's **calendar** and **invol** programs to set the target's date and time and to initialize the target volume; boot the target diskless from the partner; and, finally, log in and mount the target volume on the partner's file system.

1 Back up files and directories, as necessary.

If you are installing Domain/OS on a new workstation, skip this step. If you are installing Domain/OS on an existing workstation, back up any files and directories you want to save, since initializing the disk destroys all data on it. See "Step 2. Back Up Files" in Chapter 1 for a discussion of backup methods, a list of files and directories commonly backed-up, and suggested **wbak** command lines for backing up the objects to removable media.

2 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.

3 Shut down the target node to the Mnemonic Debugger (MD) level.

- If the target node is running the Display Manager (DM), press the < CMD > key and enter **shut** at the DM prompt:

  `Command:` **shut**

  You can also enter **shut** at the DM login prompt.

- If the target node is not running the DM, log in as **root** and issue the appropriate UNIX **shutdown** command at a shell prompt:

  /etc/shutdown -y -g0 -i5 (SysV UNIX)

  /etc/shutdown -h now (BSD UNIX)

In all cases, wait for the message `SHUTDOWN SUCCESSFUL` and for the Mnemonic Debugger (MD) prompt to appear. The prompt depends on the node firmware, but it always ends in a >.

4 Reset the target node.

To reset the node, enter the command **RE** (or **RE W** for Series 10000 workstations), followed by a carriage return at the next prompt. For example,

```
> RE <RETURN>
> <RETURN>
MD7C  Rev. 6.0, 1989/05/25.15:18:03
>
```

If the MD message does not appear after you press <RETURN> the second time, repeat the process.

5 Select the partner node as the boot device.

If the target node is connected to an Apollo Token Ring network, enter the command

> **DI N 0***xxxx*

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

If the target node is connected to an ETHERNET network, enter the command

> **DI E 0***xxxx*

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

6 Set the target node's date, time, and time zone with the interactive **calendar** program.

To start **calendar**, enter the command

> **EX CALENDAR**

Respond to the series of **calendar** prompts. See Chapter 4 for a detailed description of the prompts. Running **calendar** at this point ensures that the **invol** program creates correct Unique Identifiers (UIDs) for the objects it creates on the disk.

7 When **calendar** completes, reset the target and select the partner node again:

```
> RE <RETURN>                [or RE W for Series 10000 workstations]
> <RETURN>
MD7C  Rev. 6.0, 1989/05/25.15:18:03
> DI N 0xxxx
```

8 Initialize the target volume and set the size of the OS paging file with the interactive **invol** program.

To start **invol**, enter the command

> **EX INVOL**

The main **invol** menu appears. If you are unfamiliar with the **invol** program, turn now to Chapter 3, which provides a detailed description of running **invol** in this context. When you finish with Chapter 3, go to step 9. If you don't use Chapter 3, use the following summarized account:

1 On the **invol** menu, select option 1 (`initialize virgin physical volume`) to initialize the entire disk (the typical response). Or, if you are initializing an existing disk that 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`).

2 Respond to the subsequent prompts and wait for the disk initialization to complete. When asked,

`Anything more to do?`

enter **y**.

3 Select option 8 (`create or modify an os paging file`) on the **invol** menu, and respond to the subsequent prompts. Unless you have special paging size requirements, accept the default paging size. As of SR10.4, Domain/OS dynamically allocates paging file disk blocks as needed. The size you specify with **invol** is the initial size of the paging file, which then *grows* or *shrinks* as required.

4 When asked,

`Anything more to do?`

enter **n**.

9 Reset the target and select the partner node again:

> **RE** <RETURN> [or **RE W** for Series 10000 workstations]
> <RETURN>
`MD7C Rev. 6.0, 1989/05/25.15:18:03`
> **DI N 0***xxxx*

10 Run **calendar** again:

> **EX CALENDAR**

See Chapter 4 for a detailed description of the **calendar** prompts. Running **calendar** at this point sets the correct time for objects subsequently installed on the disk.

11 Reset and select the partner node again:

> **RE** <RETURN> [or **RE W** for Series 10000 workstations]
> <RETURN>
MD7C Rev. 6.0, 1989/05/25.15:18:03
> **DI N 0***xxxx*

12 Boot the target node from the partner node, by entering the command:

> **EX DOMAIN_OS**

After a series of messages, the DM login prompt appears.

13 Log in to the target node.

If the Aegis environment is not installed on the partner node, you must log in as **root**, since the SysV and BSD UNIX mount commands (which you'll use later in this procedure) require you to be **root**.

14 Mount the target volume on the partner node's file system.

The mount command you use depends on which Domain/OS environments are installed on the partner node. The UNIX SysV and BSD mount commands (**/etc/mount**) require that you be logged in as **root**. The Aegis mount command (**/com/mtvol**) does not.

- In an Aegis environment enter:

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

  where:

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

  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 disk as a single logical volume.

pathname is a unique directory pathname of your choice that becomes the mount point of the target volume (for example, **target**).

- In a UNIX environment enter:

  mkdir */pathname*

  and then one of the following commands:

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

  /etc/mount /dev/wn*N***a** */pathname* (BSD)

  where:

  pathname is a unique directory pathname of your choice that becomes the mount point of the target volume (for example, **target**).

  N is the unit number of the target volume.

Step 3. Install Domain/OS on the Target

You can now install an operational configuration of Domain/OS from an Authorized Area to the target volume. The following procedure instructs you to use the **install++** tool to install Domain/OS. Instead of **install++**, you can use the **config** and **install** tools. We've chosen **install++** because it is somewhat easier in this situation, where you'll want to install the operating system immediately after configuring it. Functionally, the two methods are pretty much the same. See Chapter 10 for a complete description of these tools.

1 Invoke the **install++** tool.

To invoke **install++**, enter the command line

AA/**install/tools/install++** **-vxs** *AA* [**-c** *configuration_file*] *target*

where

AA is the pathname of the Authorized Area containing the version of Domain/OS that you want to install. The Authorized Area can reside on the partner node or on some other node.

configuration_file is the pathname of a file (of your choice) in which to save the configuration of Domain/OS that you'll define during the configuration phase of **install++**; for example, *//node_name*/**config.os.v.10.4**. You can place the file anywhere on the network. If you don't want to save the configuration for later use, omit the **-c** switch and the configuration file pathname.

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.

This command line is suggestive only. -v (verbose) and **-x** (continue on error) are two recommended options. See "install++" in Chapter 10 for a complete description of **install++** command-line options.

When you invoke **install++**, it invokes the interactive **config** tool. **config** displays a list of the products available for installation from the specified Authorized Area and displays a `CONFIG>` prompt.

2 Select Domain/OS for installation.

Enter the command

CONFIG> **select os 10.*x***

where *x* is the version of Domain/OS to be installed, as displayed in the list of available products. (Although this procedure is self-contained, see "config" in Chapter 10 for more information about **config** and its commands.)

3 Configure Domain/OS.

Enter the command

CONFIG> **configure os 10.*x*** [or **co os 10.*x***]

where *x* is the version of Domain/OS that you selected in Step 2.

This starts a configuration session for Domain/OS. You are presented with a series of configuration questions. Generally, for each subcomponent of Domain/OS, you are asked if you want to install the subcomponent as a local copy on the target node, as a link to another node on which the component is already installed, or not at all.

Answer all the questions. When you finish, **config** redisplays the CONFIG> prompt. Alternately, you can enter **STOP** in response to any configuration question. This returns you to the CONFIG> prompt. You can then resume where you left off later by entering **configure os 10.*x*** again at the CONFIG> prompt.

Note that you can change your answer to a single configuration question with the **reanswer** command. Also, you can start the entire configuration process over, throwing out all of your previous answers, with the **reconfigure** command.

4 Exit the configuration phase.

When you finish configuring Domain/OS, exit the configuration phase of **install++** by entering the command

CONFIG> **exit**

install++ now invokes the **install** tool, which performs the actual installation of Domain/OS. **install** displays a series of messages as it installs the software. When the installation finishes, you are returned to the shell prompt. Upon completion, **install++** displays a message that instructs you to shut down and reboot the target node. Ignore this message and continue with this procedure.

5 Check for error and warning messages.

Check the messages displayed by the installation phase of **install++** for any errors or warnings. Errors messages are prefixed with the label `ERROR:` and warnings messages with the label `WARNING:`. You can easily locate these messages by searching the installation transcript for these character strings. Chapter 12 explains some common error and warning messages. If you find errors, rectify the error conditions and, if necessary, reinstall Domain/OS with the **install++** or **install** tool. We recommend you save the transcript in a file for future reference.

Step 4. Perform Final Steps

After you successfully install the Domain/OS software on the target node, perform the following steps to complete the process. These include unmounting, rebooting, and recataloging the target, and restoring any objects backed up prior to initializing the target.

1 Unmount the target volume.

Unmount the target volume from the partner node's file system, using the appropriate unmount command. Use the same command line variables that you used when you mounted the volume earlier in this procedure.

/com/dmtvol {**w**|*wx:y*} [*logical_volume_number*] */pathname* (Aegis)

/etc/umount /dev/dsk/W*N***d0s1** (SysV)

/etc/umount /dev/wn*N***a** (BSD)

2 Shut down the target node.

Press the < CMD > key and enter **shut** at the DM prompt:

```
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 always ends in a >.

3 Reset the target node.

To reset the node, enter the command **RE** (or **RE W** for Series 10000 workstations), followed by a carriage return at the next prompt. For example,

```
> RE <RETURN>
> <RETURN>
MD7C  Rev. 6.0, 1989/05/25.15:18:03
>
```

If the MD message does not appear after you press < RETURN > the second time, repeat the process.

4 Boot the target node from its own boot volume by entering the command:

> **EX DOMAIN_OS**

After a series of messages, the DM login prompt appears. Note that if the network has a net ID other than 0, the system hangs when it attempts to start the standard daemons and does not finish booting. It this happens, physically disconnect the node from network, change the net ID, and reboot.

5 Log in to the target node.

6 Catalog the node.

As a result of disk initialization, the target node is now named //**node_***node_id*, where *node_id* is the system-supplied, hexadecimal node ID. Therefore, you should now use the following commands to catalog the node with a name of your choice and add the name to the master network root directory (the first command); and update the node's local registry with the names of other nodes on the network (the second command).

ctnode *node_name node_id* **-l -r -root**

ctnode -l -update

where

node_name is a node name of your choice; this becomes the name of the target node's entry directory (//*node_name*). Do not proceed the name with any slashes. You can name the node with its original name or some other name.

node_id is the system-supplied node ID of the target node. You can display this ID with the **lcnode -me** or the **bldt** command. **bldt** displays the node ID in the form **** Node *net_id.node_id*. **** .

For more information about cataloging nodes, see the *Domain/OS System Administration Guide*, the *Domain/OS System Administration Reference*, and the online manual page for **ctnode**.

7 Restore objects to the target volume.

Restore any directories or files that you backed up prior to initializing the target volume. If you backed up objects to removable media using the **wbak** command lines we provided (in the section "Step 2. Back Up Files" in Chapter 1), you can restore the objects using the **rbak** command lines in the section "Step 6. Restore Files" in Chapter 1.

The procedure in this section describes how to install Domain/OS across the network from an Authorized Area to a Domain Server Processor (DSP) whose disk you need or want to initialize. The DSP can be a new or existing DSP. A DSP is a processing unit without a keyboard or monitor attached to it, so the installation procedure is different from installing Domain/OS on a workstation.

During this procedure, you perform all your work at a **partner node**. You sit at the partner node and enter commands at its keyboard. You boot the DSP *diskless* from the partner node's boot volume and use the partner node's utility programs to set the DSP's date and time, initialize the DSP's disk, mount the DSP on the partner's file system, and unmount the DSP after the installation of Domain/OS is incomplete.

The Authorized Area containing the installation tools and the version of Domain/OS you want to install can reside on the partner node or some other node in the network.

Step 1. Prepare for the Installation

This section discusses things you should do and check before you begin the actual process of installing Domain/OS on the DSP.

- Install hardware.

  The procedure assumes that you've already physically installed all requisite hardware. If you are installing Domain/OS across the network on a new DSP, physically install the DSP and connect the DSP to the network, if you have not already done so. Use the instructions in the appropriate unpacking and installation manuals that you receive with the hardware.

- Check machine-type compatibility.

  Version 10.4 of Domain/OS (SR10.4) and later versions do not support all Domain machine types. Make sure the DSP is a type that is supported. See Tables 1-1 and 1-2 for the machine types supported and not supported, respectively, by SR10.4. Later versions of Domain/OS may support additional machine types. Check the release notes of later versions for machine compatibility information.

  The **sau** (stand alone utility) directories listed in the tables are directories that contain machine-specific utilities. You can identify the **sau** type of an existing node with the **bldt** *//node_name* command (**/com/bldt** or **/usr/apollo/bin/bldt**). **bldt** displays the **sau** number of the machine in parentheses following the text, `Domain/OS kernel`. You can display the machine type (for example, DN3500) of an existing node with the command **/com/netstat -config -n** *//node_name* or **/etc/nodestat -config -n** *//node_name*.

- Set up partner node.

  Set up a node as a partner node to the DSP, using the information in the *Domain/OS System Administration Guide*. Also, make sure the partner node satisfies the following criteria:

☐ The partner node must be running Domain/OS (SR10.*x*).

We recommend that the partner is running the same version of Domain/OS that you are going to install. This ensures that the versions of the **invol** and **calendar** programs, which the partner provides to initialize the DSP's disk and to set the DSP's date and time, are consistent with the version of Domain/OS to be installed. Use the **bldt** command to determine the version of Domain/OS running on the partner.

☐ The partner node must be running **netman**.

To check this, enter one of the following commands at the partner node and look for `netman` in the displayed list of processes:

pst	(Aegis)
ps -e	(SysV)
ps -ax	(BSD)

If **netman** is not running, you can start it by entering the command **cps /sys/net/netman** at the partner's DM command prompt or **/sys/net/netman &** at a shell prompt on the partner.

☐ The **/sau** directory for the DSP's machine type must be installed on the partner node.

See Table 1-1 for a list of the machine types supported by SR10.4 and their corresponding **/sau** directories. List the directories (using **ld** or **ls**) in the partner's node entry directory to ensure the appropriate **sau***x* directory exists.

- Know the DSP's node ID and node name.

Identify and jot down the DSP's node ID (the hexadecimal ID assigned to the DSP during the manufacturing process) and, if the DSP is not new, its current name (the name of its node entry directory). The **bldt** command displays the node ID and current node name in the form

`**** Node` *net_id.node_id* `****` `"//`*node_name*`"`

If the DSP is new, its node ID is listed on the node identification slip.

- Read the installation chapter of the *Domain System Software Release Notes*.

  The installation chapter contains Domain/OS-specific installation information that is not addressed in this manual and documents any changes to and errors in this manual. Notably, the installation chapter lists the subcomponents included in the different configurations of Domain/OS and their disk space requirements. The disk space requirements for any given Domain/OS installation vary, depending on which subcomponents you select for installation (when you run **config** or **install + +**) and which subcomponents you install as links.

  However, we recommend that you make at least a rough assessment at this time of whether the target disk is large enough for the configuration you'd like to install. Note that you need at least 12 MB of free disk space in addition to the actual size of the configuration to allow for space required by the installation processes while they execute.

  The release notes are provided in hard-copy and on-line. The pathname of the on-line release notes is *AA*/**install/doc/apollo/os.v.10.***x***_ _notes**, where *AA* is the pathname of the Authorized Area containing the version of Domain/OS to be installed, and *x* is the version number of Domain/OS.

Step 2. Initialize and Mount the DSP

The following steps prepare the DSP for the installation of Domain/OS. You back up any objects you want to save; boot the DSP diskless from the partner node; use the partner node's **calendar** and **invol** programs to set the DSP's date and time and to initialize the DSP's disk; and mount the DSP's disk volume on the partner's file system.

1 Back-up files and directories, as necessary.

Initializing a disk destroys all data on it. Therefore, if you are installing Domain/OS on an existing (versus new) DSP, back up any files or directories you want to save. Typically you back up user directories and files and user-modified startup files in the **/sys/node_data** and **/sys/node_data***node_id* directories. You can restore archived objects to the DSP after you install Domain/OS. Do *not* back up any system directories like **/sys** or **/com**, since you'll want to use the new versions that get installed with Domain/OS.

2 Log in to the partner node.

3 Boot the DSP diskless from the partner node.

Use your DSP operating guide as a reference. You should have already set up a node as a partner to the DSP, as directed in "Step 1. Prepare for the Installation."

4 Create a process on the DSP.

To create a process, enter the following command at the partner node:

crp -on *node_id*

where *node_id* is the node ID of the DSP.

5 When the **crp** banner appears on the partner node, log in to the DSP.

If the Aegis environment is installed on the partner node, you can log in as **user** or yourself. If the Aegis environment is not installed on the partner node, log in as **root**, since the SysV and BSD UNIX mount commands (which you'll use later in the procedure) require you to be **root**.

6 Set the DSP's date and time with the interactive **calendar** program.

To start **calendar**. enter the following command in the **crp** window:

/com/calendar

Respond to the series of **calendar** prompts. See Chapter 4 for a detailed description of the prompts. Running **calendar** at this time ensures that the **invol** program creates correct Unique Identifiers (UIDs) for the objects it creates on the DSP's disk.

7 Initialize the DSP's disk and set the size of the OS paging file with the interactive **invol** program.

To start **invol**, enter the following command in the **crp** window:

/etc/invol

The main **invol** menu appears. If you are unfamiliar with the **invol** program, turn now to Chapter 3, which provides a detailed description of running **invol** in this context. When you finish with Chapter 3, go to Step 8. If you don't use Chapter 3, use the following summarized account:

1 On the **invol** menu, select option 1 (`initialize a virgin physical volume`) to initialize the entire disk (the typical response). Or, if you are initializing an existing disk that 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`).

2 Respond to the subsequent prompts and wait for the disk initialization to complete. When asked,

```
Anything more to do?
```

enter **y**.

3 Select option 8 (`create or modify an os paging file`) on the **invol** menu. Respond to the subsequent prompts. Unless you have special paging size requirements, accept the default paging size. As of SR10.4, Domain/OS dynamically allocates paging file disk blocks as needed. The size you specify with **invol** is the initial size of the paging file, which then *grows* or *shrinks* as required.

4 When asked,

```
Anything more to do?
```

enter **n**.

8 Run **calendar** again:

/com/calendar

Respond to the series of prompts. See Chapter 4 for a detailed description of the **calendar** prompts. Running **calendar** at this point sets the correct time for objects subsequently installed on the disk.

9 Mount the DSP's disk volume on the partner node's file system.

The mount command you use depends on which Domain/OS environments are installed on the partner node. The UNIX SysV and BSD mount commands (**/etc/mount**) require that you be logged in as **root**. The Aegis mount command (**/com/mtvol**) does not. Enter the appropriate command(s) in the **crp** window of the partner node.

- In an Aegis environment enter

  /com/mtvol w [*logical_volume_number*] *//partner_node/pathname*

  where:

  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 DSP's disk as a single logical volume.

  partner_node is the node name of the partner node.

  pathname is a unique directory pathname of your choice that becomes the mount point of the disk volume (for example, **target**).

- In a UNIX environment enter:

  mkdir */pathname*

  and then one of the following commands:

  /etc/mount /dev/dsk/WN*N*d0s1 */pathname* (SysV)

  /etc/mount /dev/wn*N*a */pathname* (BSD)

  where:

  pathname is a unique directory pathname of your choice that becomes the mount point of the disk volume (for example, **target**).

  N is the unit number of the disk volume.

Step 3. Install Domain/OS on the DSP

You can now install an operational configuration of Domain/OS from an Authorized Area to the DSP. The following procedure instructs you to use the **install++** tool to install Domain/OS. Instead of **install++**, you can use the **config** and **install** tools. We've chosen **install++** because it is somewhat easier in this situation, where you'll want to install the operating system immediately after configuring it. Functionally, the two methods are pretty much the same. See Chapter 10 for a complete description of these tools.

1 Invoke the **install++** tool.

To invoke **install++**, enter the following command line in the **crp** window on the partner node:

AA/**install/tools/install++** **-vxs** *AA* *//partner_node/pathname*

where

AA is the pathname of the Authorized Area containing the version of Domain/OS that you want to install.

partner_node is the node name of the partner node.

pathname is the pathname at which you mounted the DSP's disk volume on the partner node.

This command line is suggestive only. **-v** (verbose) and **-x** (continue on error) are two recommended options. If you want to save the configuration of Domain/OS that you'll define with **install++** in a file for subsequent use, use the **-c** *configuration_file* option. See "install++" in Chapter 10 for a complete description of **install++**.

When you invoke **install++**, it invokes the interactive **config** tool. **config** displays a list of the products available for installation from the specified Authorized Area and and displays the `CONFIG>` prompt.

2 Select Domain/OS for installation.

Enter the command

CONFIG> **select os 10.*x***

where *x* is the version of Domain/OS to be installed, as displayed in the list of available products. (Although this procedure is self-contained, see "config" in Chapter 10 for more information about **config** and its commands.)

3 Configure Domain/OS.

Enter the command

CONFIG> **configure os 10.*x*** [or **co os 10.*x***]

where *x* is the version of Domain/OS that you selected in Step 2.

This starts a configuration session for Domain/OS. You are presented with a series of configuration questions. Generally, for each subcomponent of Domain/OS, you are asked if you want to install the subcomponent as a local copy on the target node, as a link to another node on which the component is already installed, or not at all.

Answer all the questions. When you finish, **config** redisplays the CONFIG> prompt. Alternately, you can enter **STOP** in response to any configuration question. This returns you to the CONFIG> prompt. You can then resume where you left off later by entering **configure os 10.*x*** again at the CONFIG> prompt.

Note that you can change your answer to a single configuration question with the **reanswer** command. Also, you can start the entire configuration process over, throwing out all of your previous answers, with the **reconfigure** command.

4 Exit the configuration phase.

When you finish configuring Domain/OS, exit the configuration phase of **install++** by entering the command

CONFIG> **exit**

install++ now invokes the **install** tool, which performs the actual installation of Domain/OS. **install** displays a series of messages as it installs the software. When the installation finishes, **install++** displays a message that instructs you to shut down and reboot the target node. Ignore this message and continue with this procedure.

5 Check for error and warning messages.

Check the messages displayed by the installation phase of **install++** for any errors or warnings. Errors messages are prefixed with the label `ERROR:` and warnings messages with the label `WARNING:`. You can easily locate these messages by searching the installation transcript for these character strings. Chapter 12 explains some common error and warning messages. If you find errors, rectify the error conditions and, if necessary, reinstall Domain/OS with the **install++** or **install** tool. We recommend you save the transcript in a file for future reference.

Step 4. Perform Final Steps

After you successfully install the Domain/OS software on the DSP, perform the following steps to complete the process. These include unmounting, rebooting, and recataloging the DSP, and restoring any objects backed up prior to initializing the DSP's disk.

1 Unmount the DSP from the partner node.

Unmount the DSP from the partner node's file system, by entering one of the following commands in the **crp** window of the partner node. Use the same command line variables that you used when you mounted the DSP earlier in this procedure.

/com/dmtvol w [*logical_volume_number*] */pathname* (Aegis)

/etc/umount /dev/dsk/W*N***d0s1** (SysV)

/etc/umount /dev/wn*N***a** (BSD)

2 Log out from the DSP.

To log out, enter the following command in the **crp** window of the partner node:

shutspm

3 Reboot the DSP.

Reset and reboot the DSP by entering the following commands in an **emt** window over a serial line or using a dumb terminal connected to serial line:

```
> RE <RETURN>
> <RETURN>
MD7C  Rev. 6.0, 1989/05/25.15:18:03
> EX DOMAIN_OS
```

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

4 Log back on to the partner and create a process on the DSP, by entering

crp –on *node_id*

where *node_id* is the node ID of the DSP.

5 Catalog the DSP.

As a result of disk initialization, the DSP is now named //**node_***node_id*, where *node_id* is the system-supplied, hexadecimal node ID. Therefore, you should now use the following commands to catalog the DSP with a name of your choice and add the name to the master network root directory (the first command); and update the DSP's local registry with the names of other nodes on the network (the second command).

ctnode *node_name node_id* **–l –r –root**

ctnode –l –update

where

node_name is a node name of your choice; this becomes the name of the DSP's node entry directory (*//node_name*). Do not proceed the name with any slashes. You can name the DSP with its original name or some other name.

node_id is the DSP's system-supplied node ID. You can display this ID with the **lcnode –me** or the **bldt** command. **bldt** displays the node ID in the form **** Node *net_id.node_id.* **** .

For more information about cataloging nodes, see the *Domain/OS System Administration Guide*, the *Domain/OS System Administration Reference*, and the online manual page for **ctnode**.

6 Restore objects to the DSP.

Restore any directories or files that you backed up prior to initializing the DSP's disk.

3

Running invol: A Detailed Account

A detailed description of how to initialize a disk and set the OS paging file size with the **invol** program — a supplement to the Domain/OS installation procedures in Chapters 1 and 2

This chapter provides a detailed description of how to initialize a disk and set the size of the OS paging file with the **invol** program. This information is intended only as an optional supplement to the procedures in Chapters 1 and 2, which describe these steps in summary fashion only and refer you here for detailed information.

This chapter is not a comprehensive description of **invol**. For a comprehensive description, see the online manual pages for **invol** or the hard copy manual pages in the *Aegis Command Reference* or the *Domain/OS System Administration Reference.*

Step 1. Initialize the Disk

1 Select option 1 or 3 on the **invol** menu.

When you invoke **invol**, the following menu appears.

```
invol (initialize_volume), revision 10.4, April 22, 1991 1:20:13 pm

Options are:
0             -  EXIT.
1  [-fnb5um]  -  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
      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         -  OBSOLETE
11         -  remove from existing badspot list
12         -  configure disk quota table

Option:
```

The invol menu

On the **invol** menu, enter option **1** to initialize the entire disk (the typical response). Or, if you are initializing a disk previously formatted with more than one logical volume, and you want to initialize and install Domain/OS on only one of the volumes, enter option **3**.

As the menu indicates, you can optionally follow the **1** or **3** with other flags: -fnb5um or -fnb5; for example, **1 -f**. The **-b** (`apply bsd unix acls`) and **-5** flags (`apply sys5 unix acls`) are superfluous as of SR10.4, since a SysV protection inheritance scheme is unconditionally applied to system software when you install SR10.4 or greater. We don't recommend using the **-u** flag (`use defaults`) or the **-n** flag (`make a non-bootable volume`).

Other flags to consider using:

f By default, **invol** performs the very time-consuming task of reformatting every track on the disk. The **-f** flag (`don't reformat disk`) causes **invol** to bypass this operation and execute more quickly, especially if you select Verification option 1 (described later in this section). Normally, you only need to reformat a disk that is new or when you suspect that a disk's physical formatting is damaged or corrupted.

m The **-m** flag (`build a multi-disk group`) is used to group multiple physical disks together so they logically function as a single, large disk. The **-m** flag causes **invol** to display somewhat different prompts than those described in the remainder of this section. For a description of the **-m** prompts, see the online manual pages for **invol** or the hard copy manual pages in the *Domain/OS System Administration Reference* or the *Aegis Command Reference*.

2 Specify disk type.

After you select option **1**, **invol** displays this prompt :

```
Select disk:
[w=Winch|f=Floppy|q=Quit][ctrl#:][unit#]
```

Enter **w**, followed by a disk controller unit and disk unit number, as required. The default controller and unit numbers are both 0. Therefore, if the node has a single, Winchester disk, you can just enter **w**, omitting the other designations.

A single digit and colon character (**:**) following the **w** specifies a controller number (which is optionally followed by a unit number specification). A single number alone following the **w** specifies a unit number on the first (0'th controller).

For example:

w5:	Winchester disk, controller #5, unit #0
w5:0	Winchester disk, controller #5, unit #0
w1	Winchester disk, controller #0, unit #1

3 Specify physical volume name.

You are prompted for a name for the disk you are initializing:

```
Physical volume name:
```

Enter a character string of your choice, such as **apollo**.

If you are initializing a SCSI (Small Computer Systems Interface) disk on a Series 2500 or Series 400 workstation, **invol** formats the entire disk immediately after you enter the physical volume name. The message `Formatting` appears to indicate this. Do not stop the **invol** program while the format is in progress. On other workstation types, **invol** does not format the disk until after you request badspot information in step 7.

4 Choose disk verification method.

You are asked to choose one of three possible methods that **invol** uses 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:
```

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 2, which causes **invol** to write to each block on the disk, is slower but safer than option 1. Option 3 is the safest, but also the slowest; initializing a large disk can take a considerable period of time.

5 Specify average file size.

You are prompted for the expected average size of the files that will subsequently 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 (by pressing <RETURN>). Specifying 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 allocated more efficiently. The **salvol** program tells you the average size of all files on a disk.

6 Specify logical volumes.

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 specified. 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 the `volume 1:` prompt. This is the typical response. To partition the disk 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, press <RETURN> to terminate input.

The logical volume size must be at least 30 blocks and must be a multiple of the track size for the disk. If you specify a logical volume size that is not a multiple of the track size, **invol** rounds it up to the next multiple track size and informs you. The physical volume label occupies the first block on the volume. Thus, the size of the first logical volume is always one less than a multiple of the track size. Logical volume names are optional. They are used only when **invol** lists the logical volumes on the disk (**invol** option 5).

7 Request pre-recorded badspot information.

You are asked if you want to reuse the prerecorded badspot information shipped with the disk:

```
Use pre-recorded badspot info?
```

Enter **y** in response. **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 initialization, **invol** displays a series of messages that inform you of the status of the initialization.

8 Request to continue.

When the initialization completes, you are asked,

```
Anything more to do?
```

Enter **y**. The **invol** menu reappears. Continue with the instructions in the next section.

Step 2. Set the Size of the OS Paging File

You must now set (or reset) the size of the Domain/OS paging file. As of SR10.4, Domain/OS dynamically allocates paging file disk blocks as needed. The size you specify with **invol** is the initial size of the paging file. The OS then *grows* or *shrinks* the paging file as it executes. Therefore, the **invol** setting isn't as important as with earlier Domain/OS releases, which did not dynamically allocate paging blocks. We recommend you just accept the default.

To set the paging file size,

1 On the **invol** menu, select option 8 (`create or modify an os paging file`).

2 Specify disk type.

When you select option **8**, you are prompted for the disk type:

```
Select disk:
[w=Winch|f=Floppy|q=Quit][ctrl#:][unit#]
```

Enter **w**, followed by the controller and unit numbers (as described in step 2 of the previous section ,"Initialize the Disk").

3 Select logical volume.

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     xxxxxx(d)

Enter logical volume number:
```

Enter the number of the volume on which you plan to install Domain/OS. If the disk contains only one logical volume, enter **1**.

4 Specify OS paging file size.

You are prompted for the size of the paging file:

```
Size in kB for the OS paging file (CR for default value =
640):
```

Enter the desired size, or press <RETURN> to accept the default value. For SR10.2, SR10.3, and SR10.4, the recommended size is 2048 blocks for Series 10000 and Series 400 workstations, and 640 blocks for all other workstation types. For SR10.1 and SR10.0, the recommended size is 590 blocks.

5 Exit **invol**.

After you specify the paging file size, **invol** 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 1 or Chapter 2.

Example

To initialize an entire Winchester disk as a single logical volume and to set the size of the OS paging file,

```
invol (initialize_volume), revision 10.4, April 22, 1991  1:20:13 pm


Options are:
  0            - EXIT.
  1 [-fnb5um] - 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
       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            - OBSOLETE
 11            - remove from existing badspot list
 12            - configure disk quota table

Option: 1

Select disk: [w=Winch|f=Floppy|q=Quit][ctrl#:][unit#] w

Physical volume name: beeza2

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: 3
Expected average file size, in kB (CR for default, 5 kB): <RETURN>

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 specified.  Terminate input with a blank line.
```

```
Specifying a size of "all" will use all remaining blocks.

There are 146879 kB  available.

volume 1: all

Use pre-recorded badspot info? y
The pre-recorded badspot list contains    5(d) entries

Writing logical volume 1.
   Formatting...  % complete:
       20
       40
       60
       80
      100

   Writing all blocks... % complete:
       20
       40
       60
       80
      100

   Reading all blocks... % complete:
       20
       40
       60
       80
      100

Initialization complete.

Anything more to do?  y
```

```
Options are:
  0             - EXIT.
  1 [-fnb5um] - 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
       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             - OBSOLETE
 11             - remove from existing badspot list
 12             - configure disk quota table

Option: 8


Select disk: [w=Winch|f=Floppy|q=Quit][ctrl#:][unit#] w

Volume built by Invol version "revision 10.4"  on  Oct. 15, 1991

Physical volume "beeza2".  Logical volumes:

 #  size (kB)     name
 1  146879(d)

Enter logical volume number:  1
Size in kB for the OS paging file (CR for default value = 640) <RETURN>

Done.

Anything more to do? n
```

Running calendar: A Detailed Account

A detailed description of how to reset a node's date, time, and time zone with the **calendar** program — a supplement to the Domain/OS installation procedures in Chapters 1 and 2

This chapter provides a detailed description of how to reset a node's date, time, and time zone with the **calendar** program. This information is intended only as an optional supplement to the procedures in Chapters 1 and 2, which describe these steps in summary fashion only and refer you here for more detailed information. For more information about the **calendar** program, see the Aegis online manual page for **calendar** or the *Aegis Command Reference*.

Running calendar

1 Specify the disk type.

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**, followed by a disk controller unit, disk unit number, and logical volume number, as required. The default controller, unit, and logical volume numbers are 0, 0, and 1, respectively. Therefore, if the target is a single disk formatted as one logical volume, you can just enter **w**, omitting the other designations.

A single number and colon character (**:**) following the **w** specifies a controller number (which is optionally followed by a unit number specification). A single number alone following the **w** specifies a unit number on the first (0'th controller).

Examples:

w5:	Winchester disk, controller #5, unit #0
w5:0	Winchester disk, controller #5, unit #0
w1	Winchester disk, controller #0, unit #1

2 Optionally request to reset the time zone.

You are asked if you want to reset the target node's current time zone. For example,

```
The time-zone is set to 0:00 (UTC). 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 Specify a new time zone.

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 Coor-
    dinated Time in the form "hour:minutes" or
    "-hour:minutes" (e.g. 9:00, -3:00). Time-zone differences
    west of Greenwich are negative and those east of Green-
    wich 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 (for example, **-5:00** for Eastern Standard Time). Note that GMT (Greenwich Mean Time) is the same as UTC (Universal Coordinated Time).

4 Optionally request to reset the date and time.

The program displays the node's current date and time settings, and asks if you want to reset it. For example,

```
The calendar date/time is 1991/09/20 09:07:38 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 1 or Chapter 2. Enter **y** if the date or time is incorrect.

5 Specify a new date and time.

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): 1991/09/20
Please enter the local time in 24 hour format (hour:minute):
09:10
```

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?
```

You are given this warning because, if you set a node's time backward, it is *theoretically* possible that an object's unique identifier (UID) that is generated after the time change may be generated at the same node time as another object's UID that was generated before the time change. This results in two objects with the same UID. In our context, an object subsequently installed on the target node or restored to the node after you install Domain/OS 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.

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 1 or Chapter 2.

Loading and Installing Products

How to load products from media into an Authorized Area, configure and install products, and deinstall products

This chapter describes how to load, configure, install, and deinstall products. Specifically, it provides four essential procedures:

- Loading products from distribution media (cartridge tape, magnetic tape, or floppy disk) into a disk Authorized Area, using the **rbak** and **distaa** tools.
- Configuring products in an Authorized Area, using the **config** tool.
- Installing products from an Authorized Area to one or more nodes, using the **install** tool.
- Deinstalling an installed product, using the **install** tool.

When to Use These Procedures

To get a product from distribution media to an installed state on a node, perform the first three procedures (load, configure, and install). To install a product that already resides in an Authorized Area, ignore the first procedure (load) and perform the second (configure) and third (install).

The load, configure, and install procedures can be used for optional products, patches, Product Support Kits (PSKs), and Domain/OS updates. By *Domain/OS update* we mean a version of Domain/OS that does not require disk initialization prior to installation and that is installed on a node already running an earlier version of Domain/OS. To install a version of Domain/OS on a new or initialized node, use the procedure in Chapter 1 (media install) or Chapter 2 (network install). The deinstall procedure applies to all products except Domain/OS.

Requirements

If you are loading Domain/OS or a PSK, you can load the product into an existing Authorized Area, or you can use the load procedure to create a new Authorized Area. If you are loading a patch or an optional product, the node must already contain an Authorized Area. In all cases, the node onto which you are loading or installing products must be running the Domain/OS operating system (SR10.*x*).

Alternate Methods

The first three procedures (loading, configuration, and installation) instruct you to use the **distaa**, **config**, and **install** tools, respectively. An alternate approach is to use the **minst** tool (see "minst," Chapter 10). **minst** enables you to just load products from distribution media, or to load, configure, and install products in one interactive session.

Also, instead of using the **config** and **install** tools separately, you can use the **install++** tool (see "install++," Chapter 10). **install++** invokes **config** and **install**, combining configuration and installation into a single process. Functionally, using **install++** is essentially equivalent to using **config** and **install** separately. The primary difference is that **install++** enables you to define a temporary product configuration that is saved only for the duration of the installation session, whereas invoking **config** directly always saves the product configuration in a file for subsequent use.

Loading Products from Media into an Authorized Area

The following procedure describes how to load products — optional products, patches, Product Support Kits (PSKs) and Domain/OS — from distribution media into an Authorized Area. You must load a product into an Authorized Area before you can configure and install an operational configuration of the product on a node. The node on which you want to load products must be running an SR10.*x* version of Domain/OS and be connected to the appropriate type of drive (cartridge tape, magnetic tape, or floppy disk).

If you are loading Domain/OS or a PSK, you can load the product into an existing Authorized Area or use this procedure to create a new Authorized Area. The new Authorized Area will contain the installation tools shipped on the Domain/OS or PSK distribution media. If you are loading an optional product or patch, you must load the product into an existing Authorized Area.

This procedure uses the **distaa** tool. Alternately, you can use the **minst** tool and quit **minst** after its load phase completes (see "minst," Chapter 10).

1 Log in as **root**.

You must be **root** for the ACLs (Access Control Lists) of the restored objects to be set correctly.

2 Change the current directory to the Authorized Area directory, using the Aegis **wd** command or the UNIX **cd** command.

If you are loading Domain/OS or a PSK and want to create a new Authorized Area, change the current directory to the directory that you want to use as the Authorized Area. This directory must already exist, but the **install** subdirectory required by an Authorized Area doesn't have to (it gets automatically created). You can use any directory for the Authorized Area, but we recommend that you do not use the node entry directory (*//node_name*). Also, we recommend that you create a directory to be used exclusively as an Authorized Area (for example, *//node_name*/**aa_sr10.4**).

3 Physically mount the media.

Insert the first volume of the distribution media into the drive, or mount the magnetic tape. If you are loading Domain/OS from cartridge tape, be sure to insert the first product volume, not the boot volume.

4 Load administrative Authorized Area components, using the **rbak** command.

If you are loading Domain/OS, a PSK, or patches, enter this command:

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

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

rbak -dev *dev* **-ms -sacl -pdt -force -du -f 1**
install/templates -as install/templates
install/doc -as install/doc
install/toc -as install/toc

In both command lines,

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

These commands load administrative components from the media into the Authorized Area; they do not load the actual products. (See "Relation of Authorized Areas to Distribution Media" in Chapter 11 for more information). The first command line loads *all* administrative components into the Authorized Area: the installation tools (presently provided with Domain/OS and PSKs only) and their help files, the product release documentation, the predefined override and selection files for the product(s), and the media Table of Contents (TOC).

The second command line loads the same components as the first command line, except it does not load the installation tools, if they are resident on the media. Some older optional products contain old (pre-SR10.4) installation tools on the media. You don't want to load these tools because you'll overwrite the later versions of the tools already in your Authorized Area.

5 Read the product release notes.

The release notes for each product on the media now reside online. The pathname of each set of notes is *authorized_area*/**install/doc/apollo**/*product_name*.**v**.*version*__**notes**. At least read the installation chapter (usually Chapter 2) of the release notes for each product you intend to load. Read the installation chapter to:

- Find out about any product–specific installation issues, limitations, or dependencies.
- Determine which patches you want to load, if you are loading patches from a patch tape.
- Determine which HP-supplied selection file(s) you want to use, if any, with the **distaa** tool (in Step 7). Every product ships with at least one selection file that selects the entire product for loading. You use this type of selection file when the media contains more than one product (a patch tape, for example) and you want to select only some of the products, not all of them, for loading. Some products, most notably Domain/OS, also ship with selection files that select a subset of the product for loading. A product's selection files are located in the Authorized Area directory **install/templates/apollo**/*product_name*.**v**.*version*. They are named **aa**.*name*; for example, **aa.aegis_small**. See "Selection Files" in Chapter 11 for more information.
- Determine the size of the product(s), especially if the product is Domain/OS. Use the Aegis **lvolfs** or the UNIX **df** command to make sure you have enough available disk space for the product(s). If you don't, you can use a selection file to load a subset of a product, or you can distribute the product among more than one disk (as explained in the next step).

6 Optionally edit a selection file to distribute the product among more than one disk.

When you load a product into the Authorized Area with the **distaa** tool (in Step 7), by default all product components are loaded into the Authorized Area that you specify on the **distaa** command line. If a product is too large to fit on one disk, you can distribute the product among more than one disk as you load it from media. To do this, you edit one of the product's selection files and then supply the pathname of this selection file as a command line argument to **distaa**. Selection files are ASCII files which you can edit with any text editor. You can edit the selection file that selects the entire product for loading or edit a selection file that specifies a product subset.

A selection file consists of a series of move commands, one for each component of the product. The third field in each move line specifies the component; the last field specifies the destination of the component. Initially, the value of the destination field is either **-rootaa** (root Authorized Area) or **-nil**. If the value is **-rootaa**, **distaa** loads the component into the Authorized Area specified on the **distaa** command line. If the value is **-nil**, **distaa** does not load the component at all.

```
/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      -rootaa
move   ri.apollo.dialog.v.3.4   /examples -nil
```

Sample Selection File

To distribute the product, replace **-roota** with the pathname of the desired Authorized Area for those components you want to distribute. Do not specify the Authorized Area that you specify on the **distaa** command line. The directory you specify must exist, although no Authorized Area components, including the **install** subdirectory, need already reside in the directory. Do not change any destination fields to **-nil**, and do not edit any of the other fields.

For those components for which you change **-rootaa**, **distaa** creates a link from the root Authorized Area to the component in the specified Authorized Area (to *AA*/**install/ri.apollo.***product_name*.**v.***version*/*component_name*).

7 Load the product(s) into the Authorized Area, using the **distaa** tool.

- To load all products in their entirety from the media, enter this command:

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

  where

  AA is the pathname of the Authorized Area.

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

- To load a single product (if the media contains more than one product) or to load a subset of a product, enter this command:

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

  where

  AA is the pathname of the Authorized Area.

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

  selection_file is a pathname of the selection file that specifies the desired product or product subset (*AA*/**install/templates/apollo**/*product_name*.**v.***version*/**aa.***name*).You can only specify one selection file on the command line. If you want to load more than one product (but not all), you must invoke **distaa** separately with the appropriate selection file pathname for each product.

These commands load each product and its release index into the directory **install/ri.apollo.***product_name*.**v.***version* in the specified Authorized Area. If the products reside on more than one piece of media, you are prompted during the load process to remove media from the drive and replace it with the next piece.

If a superset (a larger set of components) of the same version of the product you are loading already resides in the Authorized Area, we recommend that you remove the resident product before loading the subset. (See "Removing Products from an Authorized Area" in Chapter 9). This isn't essential because once you activate the override file for the newly loaded subset (Step 8), the installation tools behave as if only the subset resides in the Authorized Area. However, removing the superset can save significant disk space. If a subset of the product you are loading already resides in the Authorized Area, no special steps are required.

The command lines are suggestive only. For a complete description of **distaa** and its command line options, see "distaa" in Chapter 10.

8 Make the override file active, if necessary.

If you used a selection file (in Step 7) to load a subset of a product, or if you loaded a different configuration of a product that already resides in the Authorized Area, enter one of the following commands:

Aegis environment:

cpf *override_file* *AA*/**install/overrides/ri.apollo.***product_name***.v.***version*

UNIX environment:

cp *override_file* *AA*/**install/overrides/ri.apollo.***product_name***.v.***version*

where

override_file is the pathname of the override file associated with the selection file that you used in Step 7, or the pathname of the override file that corresponds with a full set of product components. (For more information about override files, see "Override Files," Chapter 11). You use the latter type of override file if you loaded an entire product into an Authorized Area that already contained a subset of the product. Override files have the same pathname as the corresponding selection file, except the prefix **ov** is used in the file name instead of **aa (install/templates/apollo/***product_name***.v.***version***/ ov.***name***)**.

AA is the pathname of the Authorized Area.

product_name is the name of the product (as used in the name of the product directory and the product's release index).

version is the version number of the product (as used in the name of the product directory and the product's release index).

This makes the override file for the product active and overwrites any previously active override file. Making the override file active causes the configuration options for the product to be consistent with the set of product components in the Authorized Area.

Examples

To load a small Aegis subset of version 10.4 of Domain/OS from cartridge tape into the Authorized Area **//server/aa**,

% cd //server/aa

% install/tools/rbak_sr10 -dev ct0 -ms -sacl -pdt -force -du -f 1 -all

```
Label:
   Volume ID:      ST0168
   Owner ID:       apollo
   File number:    1
   File section:   1
   File ID:        force
   File written:   1991/09/18 11:03:03 EDT

Starting restore:
Restore complete.
```

% install/tools/distaa -f -m c //server/aa
//server/aa/install/templates/apollo/os.v.10.3/aa.aegis_small

```
000000   00
CRTG_std_sfw_1
std_sfw         10.4         68


Retensioning cartridge tape... Please wait.

Label:
   Volume ID:      ST0168
   Owner ID:       apollo
   File number:    2
   File section:   1
   File ID:        ri.apollo.os.v.10
   File written:   1991/09/18 11:08:52 EDT

Starting restore:
Restore complete.

Label:
   Volume ID:      ST0168
   Owner ID:       apollo
```

```
   File number:    3
   File section:   1
   File ID:        /base_sysboot
   File written:   1991/09/18 11:09:47 EDT

Starting restore:
Restore complete.

...
```

% cp //server/aa/install/templates/apollo/os.v.10.4/ov.aegis_small //server/aa/install/overrides/ri.apollo.os.v.10.4

To load all products in their entirety from the 1990 Compilers cartridge tape (a tape that contains several cc, FORTRAN, and Pascal compiler products) into the Authorized Area **//server/aa,**

% cd //server/aa

% install/tools/rbak_sr10 -dev ct0 -ms -sacl -pdt -force -du -f 1 install/templates -as install/templates install/doc -as install/doc install/toc -as install/toc

...

% install/tools/distaa -a -f -m c //server/aa

...

To load patch pd91_m0223 from a patch tape into the Authorized Area **//server/aa,**

% cd //server/aa

% install/tools/rbak_sr10 -dev ct0 -ms -sacl -pdt -force -du -f 1 -all

...

% install/tools/distaa -f -m c //server/aa //server/aa/install/templates/apollo/pd91_m0223.v.1.0/aa.pd91_m0223

...

Configuring Products in an Authorized Area

The following procedure describes how to define or modify a configuration of one or more products in an Authorized Area, using the interactive **config** tool.

To define a product configuration, you invoke **config**, select the products you want to include in the configuration, and then answer a series of predefined configuration questions for each of the selected products. The most common configuration questions ask whether you want to install a particular subcomponent of a product and whether you want to install the subcomponent as a local copy on the target node or as a link to some other node.

When you exit **config**, the configuration you define is saved in a configuration file. To actually install the product configuration, you supply the pathname of the configuration file as a command-line argument to the **install** tool (as described in the next section). Although configuring products is recommended, it is optional. Instead of configuring a product, you can supply the pathname of the default configuration file that ships with every product as an argument to the **install** tool.

This procedure describes how to invoke the **config** tool directly. An alternate approach is to use **install+ +**, which invokes the **config** and **install** programs as subroutines, combining product configuration and installation into a single process (see "install+ +," Chapter 10).

1 Invoke the **config** tool.

Invoke the **config** tool by entering the 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 configure.

configuration_file is the pathname of the configuration file that you want to create or modify. You can create a configuration file anywhere on disk; it does not have to reside in an Authorized Area.

When you invoke **config**, an interactive configuration session is started. **config** displays a list of all products available in the specified Authorized Area and a CONFIG> prompt appears. At the CONFIG> prompt, you can enter any one of a number of **config** commands. (Although this procedure is self-sufficient, you may wish to see "config" in Chapter 10 for a comprehensive description of the **config** commands. You can also enter **help** at the CONFIG> prompt to display a list of the commands and a brief description of each.)

You can abbreviate these commands to the point of uniqueness. For example, to redisplay the list of available products, you can enter **show available** or **s a**. You can quit the configuration session at any time and return to the shell prompt by entering **abort** or **exit**. **abort** does not save any of the configuration changes you have made; **exit** saves changes.

2 Display selected products (**s s**).

At the CONFIG> prompt, enter **show selection** or **s s**. **config** displays all products currently selected. If you are creating a new configuration file, no products are selected initially. If you are modifying an existing configuration file, the products in that configuration are automatically selected initially.

3 Select products and/or modify the list of selected products.

Use the **select**, **select all**, **update**, **update all**, or **deselect** command to select products and modify the current list of selected products, as desired. A product must be selected before you can configure it (using **configure**) or reconfigure it (using **reconfigure**). If you exit **config** without configuring a selected product, the product is added to the configuration with all configuration questions set to their default values.

In the descriptions of these and subsequent commands, *product_name* and *version_number* are the name and version number of a product as they appear in the list of available products (**show available**) or the list of selected products (**show selections**), whichever applies.

select *product_name* [*version_number*] Add the specified product to the list of currently selected products. If you omit *version_number*, **config** selects the latest version of the product available in the Authorized Area.

select all Select the latest version of all products in the Authorized Area. If an earlier version of one of the products is already selected, this version is *not* removed from the list of selected products.

update *product_name* Replace all versions of the specified product in the list of selected products with the latest version of the product available in the Authorized Area. At least one version of the the specified product must already be selected. When a product is updated, any previously-supplied configuration answers for that product are retained, provided the configuration questions in the latest version of the product are the same.

update all Replace all products currently selected with the latest versions of these products available in the Authorized Area. When a product is updated, any previously-supplied configuration answers for that product are retained, provided the configuration questions in the latest version of the product are the same.

deselect *product_name* [*version_number*] Remove the specified product from the list of currently selected products and, if you are modifying an existing configuration file, remove the product from the configuration file (upon exit from **config**). If you omit *version_number*, **config** deselects the latest version of the product in the list of selected products.

Note that when both the m68k and a88k versions of a product reside in an Authorized Area, **config** considers the a88k version *later than* the m68k version when it must select the latest version of a product. (An a88k version is indicated by a **.p** extension in the product's version number.) For example, if you use the **select all** command, and the products **ri.apollo.os.v.10.4** and **ri.apollo.os.v.10.4.p** reside in the Authorized Area, only **ri.apollo.os.v.10.4.p** (the a88k version) is selected.

4 Optionally enter **show selections** or **s s** to verify the list of selected products.

5 Optionally specify a default link destination, using **set linkprompt**.

Many products have configurations questions that ask if you want to install a product subcomponent as a link to another node, rather than as a local copy on the target node. To keep from having to repeatedly enter the same link destination, you can specify a default link destination. To do this, enter **set linkprompt** *text*, where *text* is the default link destination, typically a node entry directory. This is particularly useful when you configure Domain/OS because of the large number of Domain/OS subcomponents that you can install as links.

If you specify a default link destination, the configuration questions that ask if you want to install a subcomponent as a link still prompt you for a link destination; you can accept the default or enter another link destination at that time. The default link text appears for all products configured during the current configuration session; it doesn't change until you issue another **set linkprompt** command.

To remove a default link prompt, enter **set linkprompt** without any *text*. To display the current default link destination, enter **show linkprompt.**

6 Optionally configure or reconfigure the selected products.

Configuring a selected product — responding to the configuration questions defined for the product — is optional, but recommended. If you are creating a new configuration file and do not configure a selected product, that product is added to the configuration (upon exit from **config**) with the answers for all of the product's configuration questions set to their default values. If you are modifying an existing configuration file and do not configure a selected product, the answers currently defined in the configuration file are retained. If Domain/OS is selected, configure it before you configure any other selected products.

To configure a selected product, use one of the following commands or an appropriate combination of these commands, at the `CONFIG>` prompt:

configure *product_name version_number* The **configure** command displays only those questions not answered for the product in the current or a previous configuration session. Use **configure** when you are configuring a product for the first time (for the specified configuration file). Also use **configure** if you stopped configuring a product earlier in the configuration session before you responded to all of the questions, and now want to resume where you left off. Some products have no configuration questions. In this case, **configure** displays the message "All queries answered for the product."

reconfigure *product_name version_number* The **reconfigure** command displays all questions for the product, regardless of whether they have already been answered. Upon invocation, **reconfigure** resets all configuration questions to their default values; any answers already defined in the configuration file or specified during the current configuration session are thrown out. Use **reconfigure** when you are modifying an existing configuration file or configured a product earlier in the current configuration session and want to change all or several of the previously supplied configuration answers for the product.

reanswer *product_name version_number query_name* Use the **reanswer** command to change a previous answer to a single configuration question (*query_name*) for a product. To identify the *query_name* for a configuration question, use the **show queries** command or review the configuration transcript produced by **configure** or **reconfigure**. In the configuration transcript, the *query_name* for each configuration question appears before the actual text of the question as `** Name: query_name **`. **reanswer** is especially useful when you want to change one or just a few configuration answers for Domain/OS, which has a large number of configuration questions. If there are other configuration questions conditionally related to the one you reanswer, **config** displays these as well.

All configuration questions have a default answer, indicated by a `(D)`. You can select the default by pressing <RETURN> at the ==> prompt. When you finish responding to all configuration questions for a product, the `CONFIG>` prompt returns.

You can stop configuring a product before you respond to all configuration questions and return to the `CONFIG>` prompt by entering **STOP** at the ==> prompt. You can resume configuring the product later in the same configuration session with the **configure** command. If you exit the configuration session without answering all of a product's configuration questions, the unanswered questions are automatically set to their their default values.

You can review the current answers for a product's configuration questions with the **show queries** *product_name version_number* command.

7 Optionally set the type of installation object checking to perform for each selected product.

For each selected product, you can specify the type of object checking that the **install** program performs when you install the product on a node. The object checking is relevant only when you are reinstalling the product; installing the product as an update to an earlier, installed version of the product; or when another installed product has objects in common with the product you are installing.

To specify the object checking, enter the following command at the CONFIG> prompt:

install checking *product_name version_number check_type*

check_type has three possible values:

version If an object in the product being installed already exists on the target node, reinstall it only if it is a different version (has a different date and time stamp). Also, do not reinstall objects that were manually deleted, manually changed from a local copy to a link, or manually changed from a link to a copy on the target node (unless the **-m** switch, which overrides product customization, is used with the **install** tool). **version** checking is the default and the type of object checking we recommend.

none Install all objects of the product as called for by the configuration, regardless of whether any of the objects already exist on the target or have been manually changed by the user in any way; a *force* install. This overrides the default action of the **install** tool, which is to *not* reinstall objects that have been manually deleted, manually changed from a local copy to a link, or manually changed from a link to a copy.

exist Install all objects as called for by the configuration, except do *not* reinstall objects that were manually deleted, manually changed from a local copy to a link, or manually changed from a link to a copy on the target node. Unlike **version** checking, an object that already exists on the target node is reinstalled, even if it the same version as the object in the product being installed.

8 Exit the **config** program.

When you finish configuring and setting the object checking for selected products, exit the **config** program by entering **exit** (or **e**) at the CONFIG> prompt. **config** saves the product configuration in the specified configuration file and returns you to the shell prompt. Any unanswered configuration questions for selected products are set to their default values.

Example

In the following sample configuration session, we configure version 3.5 of the product **dpce** and then reanswer one of the configuration questions. The Authorized Area name is **//server/aa** and the configuration is saved in the configuration file **//server/cf.dpce_3.5**.

```
% //server/aa/install/tools/config -s //server/aa -c //server/cf.dpce_3.5

Scanning Authorized Area in //server/aa ...

            RAI Config Tool --- Version 2.01  09 Aug 90

CONFIG> s a
The following is a list of products/versions available for
selection

 1. cc            6.7.m
 2. cc            6.8.m
 3. dpce          3.5
 4. dsee          3.3.2
 5. hpvue         1.0
 6. lisp          4.0
 7. omniback      1.2
 8. os            10.4
 9. pas           8.7.m
10. pas           8.8.m


Type 'help' for command information

CONFIG> show selections
No products selected.

CONFIG> select dpce 3.5

Loading release index for dpce 3.5 ...
Product 'dpce 3.5' has been added to the configuration file.

CONFIG> s s
The following is a list of products/versions/install-checking
currently selected
 1. dpce 3.5    version     == Not all queries answered ==
```

```
CONFIG> set linkprompt //server
Link prompt set to: //server

CONFIG> configure dpce 3.5

Configure:  Type 'help' for information
            To take the default [marked (D)] for a query,
            type <RETURN>.

** Name: dpcc_help_dir **
/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 DOMAIN PC Coprocessor (or other PC
        Compatability product). If /sys/help IS a link from
        the target to another node, then you must install the
        DOMAIN PC Coprocessor 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 DOMAIN PC Compatability products
help files (/sys/help) a link to another node or neither?
: [ copy(D) link none ]
==> link
Enter the name of the node which has the DOMAIN PC Compat-
ability help files
(/sys/help) installed:: [ //server (D) ]
Link Text>>>> <RETURN>

** Name: sys5.3_usr_catman_install **
/SYS5.3/USR/CATMAN
     The /sys5.3/usr/catman directory contains man pages for
     the sys5.3 environment.  If this directory exists and is
     NOT a link on the target, you may want to install the
     man pages for DOMAIN PC Compatability Products. If
     /sys5.3/usr/catman IS a link from the target to another
     node, then you must install DOMAIN PC Compatability
     Products 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 DOMAIN PC Compatability Products
man pages, a link to another node or neither?
: [ copy(D) link none ]
==> link
Enter the name of the node which has the DOMAIN PC Compat-
ability Products man pages
installed:: [ //server (D) ]
Link Text>>>> <RETURN>

** Name: bsd4.3_usr_man_install **
/BSD4.3/USR/MAN
     The /bsd4.3/usr/man directory contains man pages for the
     bsd4.3 environment. If this directory exists and is NOT
     a link on the target, you may want to install the man
     pages for DOMAIN PC Compatability Products. If
     /bsd4.3/usr/man IS a link from the target to another
     node, then you must install DOMAIN PC Compatability
     Products 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 DOMAIN PC Compatability Products
man pages, a link to another node or neither?
: [ copy(D) link none ]
==> copy

All queries answered for product: dpce  3.5
NOTE: Only previously unanswered queries were asked.
      To answer all queries for this product, use the
      command:
      reconfigure dpce 3.5

CONFIG> reanswer dpce 3.5 bsd4.3_usr_man_install

Configure:  Type ´help´ for information
            To take the default [marked (D)] for a query,
            type <RETURN>.

** Name: bsd4.3_usr_man_install **
/BSD4.3/USR/MAN
     The /bsd4.3/usr/man directory contains man pages for the
     bsd4.3 environment. If this directory exists and is NOT
```

```
        a link on the target, you may want to install the man
        pages for DOMAIN PC Compatability Products. If
        /bsd4.3/usr/man IS a link from the target to another
        node, then you must install DOMAIN PC Compatability
        Products 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 DOMAIN PC Compatability Products
man pages, a link to another node or neither?
: [ copy(D) link none ]
==> link
Enter the name of the node which has the DOMAIN PC Compat-
ability Products man pages
installed:: [ //server (D) ]
Link Text>>>> <RETURN>

All queries answered for product: dpce  3.5

CONFIG> exit

Validating selected products, please wait ...
New configuration file saved in //server/cf.dpce_3.5
```

Installing Products from an Authorized Area

The following procedure describes how to install a configuration of products from an Authorized Area to one or more target nodes using the **install** tool. The procedure assumes you already created a configuration file that defines the configuration of products to be installed (as described in the preceding section), unless you want to install the default configuration of one or more products.

Instead of using the **install** tool, you can use the **install++** tool (see "install++," Chapter 10). **install++** invokes the **config** and **install** programs, combining product configuration and installation into a single process.

1 Prepare for the installation.

To prepare for the installation,

- ☐ Read the release notes for each product you are installing. Online release notes are located in the Authorized Area subdirectory **install/doc/apollo** and are named *product_name*.v.*version*__**notes**. Pay particular attention to the installation chapter in each set of release notes (usually Chapter 2), which discusses product-specific installation issues and dependencies and gives the product size. The release notes for Domain/OS and PSKs also describe any errors in this manual and any changes to the installation tools made since the release of this manual.
- ☐ If you are installing an optional product, patch, or Product Support Kit (PSK), make sure the target nodes (the nodes on which you are installing software) are running the version of Domain/OS required by the product. Use the **/com/bldt** or **/usr/apollo/bin/bldt** program to determine the version of Domain/OS running on a node.
- ☐ Use the Aegis **lvolfs** or UNIX **df** command to check the free disk space on each target node. Make sure there is enough free space to contain the product(s). If you are installing Domain/OS or a large optional product, and the target node is the node on which you invoke the **install** tool, also make sure there is at least approximately 12 MB of free space in addition to the actual size of the product. This allows for disk space required by the installation processes as they execute.

- ☐ Make sure the target nodes are not running the **lprotect** program. The **install** tool does not install software on nodes running **lprotect**.

2 Install the product configuration.

To install the product configuration, enter the following command line. This is a suggested command line only. See "install" in Chapter 10 for a complete listing of **install**'s command-line options.

AA/**install/tools/install** **-vx** **-s** *AA* **-c** *configuration_file* *target* [*target* ...]

where

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

configuration_file is the pathname of the configuration file that defines the configuration of products to be installed. This can be a configuration file you created previously with the **config** tool, or the default configuration file for a single product. Default configuration files reside in the Authorized Area subdirectory **install/templates/apollo/***product_name***.v.***version* and are named **cf.***product_name*. You can list more than one configuration file on the same command line, preceding each with the **-c** switch.

target [*target* ...] is a list of installation targets. Although you can supply the pathname of any directory, you usually supply the name of a node's entry directory (*//node_name*) or the mount point of a mounted disk volume (as when the target is booted diskless from another node). Instead of, or in addition to, specifying targets explicitly on the command line, you can create a file containing a list of targets (one per line) and specify the pathname of this file on the command line preceded by the **-n** switch.

As **install** executes, it displays various informational messages. When **install** completes execution, it displays one of the following messages:

```
RAI install has successfully completed

RAI install has completed with errors
```

3 Check the installation transcript for errors and warnings.

When the **install** program completes execution, check the installation transcript (the series of messages displayed by **install**) for error messages and warning messages. Errors messages are prefixed with the label `ERROR:`; warnings messages with `WARNING:`. Chapter 12 explains some common error and warning messages.

If the transcript contains errors, correct any problems that exist. Then, if necessary, rerun **install**. Errors often result from transient network problems during the installation. You can usually correct such problems by running the installation again. Another run of **install** is usually much faster than the first because, by default, **install** copies only those objects that it did not successfully install during the first run.

4 Reboot the target node, if necessary.

If the installation transcript displays a message instructing you to reboot the target node, perform the following steps. Rebooting is required when you install Domain/OS or software that changes the system libraries.

1 Shut down the target node.

If the target node is a workstation running the Display Manager (DM), enter **shut** at the DM command prompt.

If the target node is a workstation that is not running the DM, log in to the node as **root** and issue the UNIX **shutdown** command:

/etc/shutdown -y -g0 -i5 (SysV environment)

/etc/shutdown -h now (BSD environment)

If the target node is a DSP, issue the **shut** command to the Server Processor 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 ends in a >.

2 Reset the node.

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
>
```

3 Boot the node.

To boot the node, enter the command

> EX DOMAIN_OS

Wait for the login prompt to appear.

Example

In the following example, we install the configuration of products defined in the user-created configuration file **//server/cf.dpak_4.1**. This file defines a configuration of one product: version 4.1 of **dpak**. The Authorized Area is **//server/aa** and the product is installed on the nodes **//bsa** and **//bsa2**.

```
% //server/aa/install/tools/install -vx -s //server/aa -c //server/cf.dpak_4.1 //bsa //bsa2
RAI Install Tool V2.T11 12 Dec 90

Install order:
          1) ri.apollo.dpak.v.4.1

Authorized area is on //server/aa
The selected switch settings are:
    Existence of files in the AA will not be checked
    //server/cf.dpak_4.1 is the configuration file for all target nodes
    Fast installation is not selected
    Object customization is on
    Hard links are not enabled
    Remote installations are inactive
    Continue on error has been selected

Installation is for:
    //bsa
    //bsa2

Installing //bsa
Writing new baseline file //bsa/install/baseline/baseline.00000005
Using baseline file baseline.00000005 for node //bsa
Checking status of baseline file entry ri.apollo.dpak.v.4.1
Checking status of configure file entry ri.apollo.dpak.v.4.1
WARNING://bsa/usr/apollo/bin contained in multiple installed products - item
is ignored
Computing installable set for //bsa
WARNING:Using existing soft link for //bsa/usr/apollo/bin instead of changing
to a local copy
New baseline file //bsa/install/baseline/baseline.00000006 contents:
```

```
        from BF: ri.apollo.os.v.10.3
        from BF: ri.apollo.dwb.v.2.0
        from BF: ri.apollo.dpss_mail.v.2.3.1
        from BF: ri.apollo.kb_reader.v.1.1
        from CF: ri.apollo.dpak.v.4.1
Installation requires 20 blocks
WARNING: No Release Notes were found under //server/aa/install/doc
         No Release Notes will be installed for //bsa
Writing new baseline file //bsa/install/baseline/baseline.00000006
Directory region:
File Region:
    File //bsa/lib/dpatlib installed
Link Region:
    Soft link for //bsa/com/dpat to ../usr/apollo/bin/dpat created
    Soft link for //bsa/com/hpc to ../usr/apollo/bin/hpc created
    Soft link for //bsa/com/dspst to ../usr/apollo/bin/dspst created
Writing new baseline file //bsa/install/baseline/baseline.00000006
Installing //bsa2
Writing new baseline file //bsa2/install/baseline/baseline.00000009
WARNING:Value //bleeding_edge.15 not found for query usr_apollo_bin in
ri.apollo.os.v.10.3
Using baseline file baseline.00000009 for node //bsa2
Checking status of baseline file entry ri.apollo.dpak.v.4.1
Checking status of configure file entry ri.apollo.dpak.v.4.1
Computing installable set for //bsa2
New baseline file //bsa2/install/baseline/baseline.00000010 contents:
        from BF: ri.apollo.dpss_mail.v.2.3.1
        from BF: ri.apollo.dwb.v.2.0
        from BF: ri.apollo.os.v.10.3
        from CF: ri.apollo.dpak.v.4.1
Installation requires 0 blocks
WARNING: No Release Notes were found under //server/aa/install/doc
         No Release Notes will be installed for //bsa2
Writing new baseline file //bsa2/install/baseline/baseline.00000010
Directory region:
File Region:
    File //bsa2/usr/apollo/bin/hpc installed
    File //bsa2/usr/apollo/bin/dspst installed
Writing new baseline file //bsa2/install/baseline/baseline.00000010
RAI install has successfully completed
%
```

Deinstalling Products

The following procedure describes how to remove or *deinstall* an entire product that has been installed on a node. You can also deinstall selected subcomponents of a product rather than the entire product, using the **-d** switch of the **install** (or **install++**) tool (see "install" or "install++" in Chapter 10). Deinstallation is best reserved for optional products. Do *not* use this method to remove Domain/OS. To remove Domain/OS, initialize the disk using the **invol** utility.

1 Deinstall the product using the **-D** switch with the **install** tool.

To deinstall a product, enter the following command line:

AA/**install/tools/install -D** *product_name version_number* **-vxs** *AA target* [*target* ...]

where

AA is the pathname of the Authorized Area containing the products you want to deinstall. At least the release index of the product being deinstalled must reside in the specified Authorized Area.

product_name is the name of the product you want to deinstall (as used in the name of the product's release index).

version_number is the version number of the product (as used in the name of the product's release index).

target [*target* ...] is a list of deinstallation targets. For a target, you usually supply the name of a node entry directory (*//node_name*) or the mount point of a mounted disk volume (as when the target is booted diskless from another node). Instead of, or in addition to, specifying targets explicitly on the command line, you can create a file containing a list of targets (one per line) and specify the pathname of this file on the command line, preceded by the **-n** switch.

This is a suggested command line only. For a description of other command line options that you can use when you deinstall products, see "install" in Chapter 10. Note that you can also use the **-D** switch with the **install++** tool to deinstall products, but doing so is functionally equivalent to using **install**.

As **install** executes, it displays various informational messages. When **install** completes execution, it displays one of the following messages:

```
RAI install has successfully completed
```

```
RAI install has completed with errors
```

Note:

- To deinstall a product, at least the release index of *all* products currently installed on the target node must reside in the specified Authorized Area.
- **install** does *not* deinstall an object (a file, directory, or link) that is part of the product in two cases: first; if the object is part of another product installed on the target; second, if the object (judging by its name) is not an object that was originally released with the product. Such objects include user-created files or files preserved by the **install** tool when the product was updated previously. The release index of some products instructs the **install** tool to preserve certain user-modifiable files, rather than simply overwrite them, by adding a date extension to their original names. Such objects are not deinstalled since their names differ from those in the product's released file set. If an object is is not deinstalled, the directory containing that object is also not deinstalled. **install** displays a warning message whenever it cannot deinstall an object.
- **install** deinstalls a product even when another product installed on the target depends on the product being deinstalled.
- If you want to remove a patch because the patch does not function correctly, we do *not* recommend that you deinstall it, since this removes possibly critical objects from a product, without replacing them. This is especially true of Domain/OS patches, whose deinstallation may make the node unusable. Its better to reinstall the entire product (recognizing that reinstallation overwrites *all* patches to the product, not just the one that you want to remove). Or wait for another patch that corrects the original patch and then install the new patch on top of the old.

2 Check the installation transcript for errors and warnings.

When the **install** program completes execution, check the installation transcript (the series of messages displayed by **install**) for error messages and warning messages. Errors messages are prefixed with the label `ERROR:`; warnings messages with `WARNING:`. Chapter 12 explains some common error and warning messages.

If the transcript contains errors, correct any problems that exist. Then, if necessary, rerun **install**. Errors often result from transient network problems during the execution of **install**. You can usually correct such problems by running **install** again.

Example

To deinstall version 4.1 of the product **dpak**, contained in the Authorized Area **//server/aa**, from the node **//bsa**:

```
% //server/aa/install/tools/install -D dpak 4.1 -s //server/aa //bsa
RAI Install Tool V2.T11 12 Dec 90

Deleting Product ri.apollo.dpak.v.4.1
Checking status of baseline file entry ri.apollo.os.v.10.3
Checking status of baseline file entry ri.apollo.dwb.v.2.0
Checking status of baseline file entry ri.apol-
lo.dpss_mail.v.2.3.1
Checking status of baseline file entry ri.apollo.kb_read-
er.v.1.1
Checking status of baseline file entry ri.apollo.dpak.v.4.1
RAI install has successfully completed
%
```

Part 2

Advanced Installation Tasks

This part of the manual provides several *advanced* procedures: procedures intended primarily for system administrators who wish to exercise greater control over the configuration of products loaded and installed, and who want to make the most efficient and effective use of their Authorized Area(s). Single-node users, however, especially those who are concerned about saving disk space, may find many of these procedures useful as well. Users who are not responsible for managing an Authorized Area can ignore these procedures entirely.

All of these procedures are optional. None are required to load a product from distribution media into an Authorized Area, or to install an operational configuration of a product on a node.

Chapter 6 describes how to use the **cfgsa** tool to control and restrict the configurations of products users can install on a node.

Chapter 7 describes how to use the **cfgsa** and **distaa** tools to define and load a custom product subset from distribution media into an Authorized Area.

Chapter 8 describes how to use the **mrgri** tool to merge three kinds of products in an Authorized Area: patches with the products they patch, Product Support Kits (PSKs) with Domain/OS, and the a88k and m68k versions of a product.

Chapter 9 describes how to manipulate Authorized Areas using standard UNIX and Aegis commands. It tells you how to copy, move, remove, and distribute an Authorized Area with links; how to remove products from an Authorized Area; and how to load the installation tools into an Authorized Area.

Restricting Product Configurations

How to control and restrict the configurations of products users can install, using the **cfgsa** tool

Restricting Product Configurations

The following procedure describes how to control and restrict the configurations of products that users can install. For example, you can prevent users from installing product subcomponents not needed in your environment, or force users to install certain product subcomponents as links to an administrative node rather than as local copies. These measures save disk space and simplify the installation process for users.

You constrain product configurations with the interactive **cfgsa** tool. For each configuration question associated with a product, **cfgsa** enables you to limit the possible answers a user may supply or preselect an answer. When a user runs the **config** or **install + +** tool to configure the product, the user is presented with a modified set of configuration questions that reflect the constraints you define. If you preselect an answer, the question is not presented to the user at all. If the user installs the product with a configuration file that was created before you created the constraints, or with a default configuration file, the constraints you define take precedence over any configuration selections not allowed by the constraints.

Restricting product configurations is entirely optional. If you do not restrict a product configuration with **cfgsa**, the user is presented with the full range of configuration questions and options defined by HP for the product or product subset.

For a comprehensive description of **cfgsa**, see "cfgsa" in Chapter 10.

1 Invoke the **cfgsa** tool.

Invoke **cfgsa** by entering the command

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

where *AA* is the pathname of the Authorized Area containing the product(s) whose configuration(s) you want to constrain.

cfgsa displays a list of products available in the Authorized Area and a CFGSA> prompt. You can redisplay this list later during the **cfgsa** session by entering **available** (or **av**) at the CFGSA> prompt.

2 Select the product.

Select the product you want to constrain by entering the command

CFGSA> **select** *product_number*

where *product_number* is the number of the product in the list of available products. Or enter

CFGSA> **select** *product_name version*

where *product_name* and *version* are the name and version number of the product as they appear in the list of available products; for example **pas 8.7.m**. You can omit *version* if there is only one version of the product in the Authorized Area.

3 Define the configuration constraints.

Enter the command

CFGSA> **constrain**

This begins the process of defining configuration constraints for the selected product.

cfgsa displays the configuration questions that are shown to a user when a user runs **config** or **install + +** to configure the product, and the possible answers that a user can supply. If no active override file for the product exists, **cfgsa** presents the full set of questions defined by HP for the product. If an active **override** file exists, the questions presented reflect the constraints defined by the **override** file. (See "Override Files," Chapter 11, for more information about override files.)

For each question, enter **answer**, **limit**, or **user** at the YOUR CHOICE prompt.

answer Enter **answer** to answer the question for the user. **cfgsa** then prompts you to enter one of the possible answers. The configuration option is unconditionally set to the answer you supply; when a user configures the product, the question does not appear at all. If the question asks if a particular product subcomponent is to be installed as a link (versus a local copy or not at all) and you preanswer the question as **link**, **cfgsa** prompts you for the destination of the link.

limit Enter **limit** to limit the answers a user can supply to a subset of the possible answers. **cfgsa** then prompts you to enter the answers you want presented to the user. When a user configures the product, the user is presented with the question and the reduced answer set. If the text of the question refers or alludes to answers that you exclude, the question may become slightly confusing to the user.

user Enter **user** (or just press <RETURN>) to impose no constraints. When a user configures the product, the user is presented with the question and the full set of answers. This is the default response.

In response to each configuration question, you can also enter **help** to obtain information about the constraint subcommands; **refresh** to redisplay the question and answers; or **abort** to exit the constraint session. **abort** saves any constraints already defined and returns to the CFGSA> prompt.

When you finish responding to all the configuration questions, the CFGSA> prompt reappears.

4 Optionally display and change the constraints.

You can enter **show** at the CFGSA> prompt to display all the questions for the product and the constraints you've applied. To change the constraints, enter **revert**. **revert** removes *all* constraints that you've applied to the selected product during the current **cfgsa** session. You can then redefine constraints with the **constrain** command.

5 Save the constraints.

When you've finished defining the constraints for the selected product, enter the command

```
CFGSA> save
```

This saves the constraints in an overrides file for the product. The file is named **ri.apollo.***product_name***.v.***version*. It is placed in the directory *AA*/**install/overrides**, where *AA* is the name of the Authorized Area that you specified when you invoked **cfgsa**.

Placing the override file in the **install/overrides** directory makes the file *active*, meaning the constraints are in effect. If an override file for the product already exists in the **install/overrides** directory, it it overwritten with the new one.

Also note:

- **cfgsa** provides a **generate** subcommand, which you can use instead of **save**. **generate** creates both a selection file and a corresponding, but non–active, override file. You can use this selection file to load a subset of the product from distribution media. This process is described in Chapter 7.
- To subsequently remove the constraints, you can delete, move, or rename the active override file, using standard Aegis or UNIX commands. This restores the full set of configuration questions and options for the product, as defined by the product's release index. But do this only if the entire product (not a subset) resides in the Authorized Area. Or, if a product subset resides in the Authorized Area, remove the constraints only if you replace the active override file with another override file that corresponds to the product subset (or that corresponds to a set of product components that is smaller than the subset). Otherwise, a user can potentially define a product configuration that includes product components that do not physically reside in the Authorized Area. This causes numerous errors at installation time.

6 Exit **cfgsa**.

When you finish creating override files for as many products as you want, enter the command

```
CFGSA> exit
```

This returns you to the shell.

Example

In this example, we restrict the configuration of the product **c++** in the Authorized Area **//server/aa**. The restrictions force users to install the BSD4.3 c++ man pages as a link or not at all; users cannot install the man pages as a local copy. We then change this restriction so users can install the man pages only as a link to the node **//bsa**.

```
% //server/aa/install/tools/cfgsa //server/aa
RAI System Administrator Override Tool V1.01 3 Apr 89
Scanning Authorized Area in //server/aa ...

    Products Available in //server/aa
     1) ada          3.0
     2) c++          2.0.0.m
     3) dsee         3.3.2
     4) hpvue        1.0
     5) os           10.4
     6) pas          8.8.m
     7) pas          8.8.mpx


CFGSA> sel 2
Loading release index for c++ 2.0.0.m ...

CFGSA> constrain

QUESTION: Do you want a local copy of Domain/C++ executable
          components and libraries or a link to another node?
ANSWERS:  Up to 1 of [ copy(D) link ]
YOUR CHOICE [ Answer Limit User(D) Help Refresh Abort ]: user

...

QUESTION: /BSD4.3/USR/MAN
          The /usr/man directory contains man pages for the
          bsd4.3 environment. If this directory exists and is
          NOT a link on the target, you may want to install
          the man pages for Domain/C++. If /usr/man IS a link
          from the target to another node, then you must
          install Domain/C++ on that node if you want the
```

```
          latest version of these files to be available on
          the target. The C++ man pages consume about 0.18 MB
          of disk space.
          Do you want a local copy of BSD4.3 C++ man pages, 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

...

All queries for c++ 2.0.0.m have been processed

CFGSA> show

For product: c++ 2.0.0.m

...

For: Do you want a local copy of Domain/C++ executable
     components and libraries or a link to another node?
===> User chooses answer

...

For: /BSD4.3/USR/MAN
     The /usr/man directory contains man pages for the bsd4.3
     environment.  If this directory exists and is NOT a link
     on the target, you may want to install the man pages for
     Domain/C++. If /usr/man IS a link from the target to
     another node, then you must install Domain/C++ on that
     node if you want the latest version of these files to be
     available on the target. The C++ man pages consume about
     0.18 MB of disk space.
     Do you want a local copy of BSD4.3 C++ man pages, a link
     to another node or neither?
===> Answers limited to: "link" "none"

...
```

```
CFGSA> revert

CFGSA> constrain

...

QUESTION: /BSD4.3/USR/MAN
          The /usr/man directory contains man pages for the
          bsd4.3 environment. If this directory exists and is
          NOT a link on the target, you may want to install
          the man pages for Domain/C++. If /usr/man IS a link
          from the target to another node, then you must
          install Domain/C++ on that node if you want the
          latest version of these files to be available on
          the target. The C++ man pages consume about 0.18 MB
          of disk space.
          Do you want a local copy of BSD4.3 C++ man pages, 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 ]: ans
Pick 1 of [ copy(D) link none ]: link
LINK QUESTION: Enter the name of the node which has the
               BSD4.3 C++ man pages installed:
Enter link text: //bsa


QUESTION: /SYS5.3/USR/CATMAN
          The /usr/catman directory contains man pages for
          the sys5.3 environment. If this directory exists
          and is NOT a link on the target, you may want to
          install the man pages for Domain/C++. If
          /usr/catman IS a link from the target to another
          node, then you must install Domain/C++ on that node
          if you want the latest version of these files to be
          available on the target. The C++ man pages consume
          about 0.18 MB of disk space.
          Do you want a local copy of SYS5.3 C++ man pages, 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 ]: abort

CFGSA> show

...

For: /BSD4.3/USR/MAN
     The /usr/man directory contains man pages for the bsd4.3
     environment.  If this directory exists and is NOT a link
     on the target, you may want to install the man pages for
     Domain/C++. If /usr/man IS a link from the target to
     another node, then you must install Domain/C++ on that
     node if you want the latest version of these files to be
     available on the target. The C++ man pages consume about
     0.18 MB of disk space.
     Do you want a local copy of BSD4.3 C++ man pages, a link
     to another node or neither?
===> Answer forced to be: "link" and linking to: "//bsa"

...

CFGSA> save

CFGSA> exit

% ls //server/aa/install/overrides
ri.apollo.c++.v.2.0.0.m
%
```

Defining and Loading a Custom Product Subset

How to define your own product subset and load it from media into an Authorized Area, using the **cfgsa** and **distaa** tools

Domain/OS and some Domain optional products ship with a set of predefined selection files (described in the product's release notes). The selection files enable you to load a subset of the product, rather than the entire product, from distribution media into an Authorized Area. This allows you to save space on the disk containing the Authorized Area by not loading components of a product not needed in your environment.

With the **cfgsa** tool you can create *your own* selection files that define product subsets of your choosing. You can then use the selection files with **distaa** to load the product subsets into your Authorized Area. **cfgsa** also creates a corresponding override file, which limits the full range of configuration questions and options to a set consistent with the product subset. This chapter provides a five–step procedure for doing this. The procedure is entirely optional.

Summary of Procedure

Here's an overview of the procedure:

Step 1 If the product does not already reside in an Authorized Area, you use the **rbak** command to load the product's release index and other administrative objects from distribution media into an Authorized Area. **cfgsa** requires that at least the product's release index reside in an Authorized Area.

Step 2 You use the interactive **cfgsa** tool to create a selection file and a corresponding override file that define a product subset.

Step 3 You remove the entire version of the product from the Authorized Area, if it resided there before you began the procedure.

Step 4 You use the selection file you created in Step 2 with the **distaa** tool to load the product subset from media into the Authorized Area.

Step 5 You activate the override file you created in Step 2, so the configuration constraints take effect. The configuration questions presented to the user, when the user configures the product with **config** or **install++**, will then be consistent with the reduced set of product components.

Some Notes of Caution

cfgsa prohibits you, as it should, from creating a product subset that does not include a product subcomponent upon which another subcomponent is dependent. However, it is possible to exclude a subcomponent upon which *another product* is dependent. If you do this, when you attempt to configure the other product, **config** displays an appropriate warning message.

Also, because this procedure is somewhat cumbersome, you might be tempted to simply manually remove unwanted subcomponents of a product from an Authorized Area to save disk space. Do not do this. You might remove subcomponents upon which other subcomponents depend. Also, you'll have no way of ensuring that the configuration questions presented to a user by **config** or **install++** are consistent with the reduced set of product components.

Step 1. Load the Product's Release Index

If the version of the product for which you want to create a subset already resides in an Authorized Area, skip this step. If the product does not already reside in an Authorized Area, you must use the following procedure to load (with **rbak**) the product's release index and other administrative files (such as the product's release documentation) from the distribution media into the Authorized Area.

1 Change the current directory to the Authorized Area into which you want to load the product subset.

2 Insert the first product tape or floppy in the drive, or mount the magnetic tape.

3 Load the product's administrative objects from the distribution media.

If the product is Domain/OS, enter the command

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

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

If the product is an optional product, enter the command

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

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

These commands load the distribution media's Table of Contents (**install/toc**), the product release documentation (**install/doc**), and the predefined selection and override files (**install/templates**) for all products on the media into the Authorized Area. The command for Domain/OS also loads the installation tools.

4 Identify which physical file on the media contains the product's release index file.

If the product is Domain/OS or an optional product that has no other products on the distribution media, skip this step. (In these cases, the product release index is always in file 2 on the media.) If the product is an optional product and you're not sure if other products are on the distribution media, read the TOC you just loaded to determine which file on the media contains the product's release index.

The pathname of the TOC is *AA*/**install/toc/toc.apollo.***volume_id.dev*, where

AA is the pathname of the Authorized Area.

volume_id is the volume ID of the distribution media as displayed by the previous **rbak** command.

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

The TOC lists each product on the distribution media in the form

`Prod ri.apollo.`*product_name.version* *x y z*

Repeatedly search for the string `Prod` until you find the desired product (*product_name.version*). The second number (*y*) is the number of the file containing that product's release index. Make a note of this number.

5 Load the product's release index.

Load the product's release index from the distribution media into the Authorized Area by entering the command

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

where

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

n is the number of the file on the distribution media containing the product's release index. If the product is Domain/OS or an optional product that has no other products on the distribution media, $n = 2$.

Example

To load the administrative objects and release index for version 10.4 of Domain/OS (SR10.4) from cartridge tape into the Authorized Area **//aa/sr10.4**,

```
% cd //aa/sr10.4
% install/tools/rbak_sr10 -dev ct -ms -sacl -pdt -force -du -f 1 -all
Label:
   Volume ID:      ST0168
   Owner ID:       apollo
   File number:    1
   File section:   1
   File ID:        force
   File written:   1991/09/18 11:03:03 EDT

Starting restore:
Restore complete.

% install/tools/rbak_sr10 -dev ct -ms -sacl -pdt -force -du -f 2 -all
Label:
   Volume ID:      ST0168
   Owner ID:       apollo
   File number:    2
   File section:   1
   File ID:        ri.apollo.os.v.10
   File written:   1991/09/18 11:08:52 EDT

Starting restore:
Restore complete.
```

Step 2. Define a Product Subset with cfgsa

Once the product's release index is in an Authorized Area, you can define a subset of the product using the interactive **cfgsa** tool. You define the subset with **cfgsa**'s **constrain** subcommand. You then issue **cfgsa's generate** command, which creates a selection file and a corresponding override file in the current directory that reflect the specified subset. For a comprehensive description of **cfgsa** and its subcommands, see "cfgsa" in Chapter 10.

1 Change the current directory.

Change the current directory to the directory where you want to place the custom selection file and override file that you create with **cfgsa**. (**cfgsa**'s **generate** subcommand creates these files in the current directory.) You may want to put these files in the directory that contains the product's HP-supplied selection and override files: *AA*/**install/templates/apollo**/*product_name*.**v**.*version*.

2 Invoke the **cfgsa** tool.

Invoke **cfgsa** by entering the command

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

where *AA* is the pathname of the Authorized Area containing the product (or product release index).

cfgsa displays a list of products available in the Authorized Area and a `CFGSA>` prompt. You can redisplay this list later by entering **available** (or **av**) at the `CFGSA>` prompt.

3 Select the product.

Select the product by entering the command

`CFGSA>` **select** *product_number*

where *product_number* is the number of the product in the list of available products. Or enter

`CFGSA>` **select** *product_name version*

where *product_name* and *version* are the name and version number of the product as they appear in the list of available products; for example **pas 8.7.m**. You can omit *version* if there is only one version of the product in the Authorized Area.

4 Define the product subset.

Enter the command

CFGSA> **constrain**

This begin the *constraint session* -- the process of defining the product subset.

cfgsa displays the configuration questions for the product (the questions that are shown to a user when a user runs **config** or **install + +** to configure the product) and the possible answers that a user can supply. If no active override file for the product exists, **cfgsa** presents the full set of questions defined by product's release index. If an active override file exists, the questions presented reflect the constraints defined by the override file.

cfgsa pauses after each configuration question and prompts you for a response. Keep in mind that you are simultaneously defining constraints for both the loading of the product from media into an Authorized Area and for the configuration of the product (with **config** or **install + +**) prior to its installation. Depending on the particular configuration question and your responses, the results may effect only the configuration process prior to installation and not what gets loaded into the Authorized Area.

For each question, enter **answer**, **limit**, or **user**:

answer Enter **answer** to restrict the product configuration to *one* of the possible answers. **cfgsa** then prompts you to enter the answer of your choice. If the question concerns the installation of a product subcomponent and one of the possible answers is "none"or "no", entering **answer** and then **none** or **no** causes the subcomponent not to be loaded from media. Similarly, if the question asks which of a number of related objects are to be installed (such as the question about which **sau** directories to install for Domain/OS), entering **answer** and a then one of the objects causes only that object to be loaded from media. When a user configures the product, the question does not appear at all.

limit Enter **limit** to limit the product configuration to a subset of the possible answers. **cfgsa** then prompts you to enter one or more of the possible answers. If the question asks which of a number of related objects are to be installed, entering **limit** and a then a subset of the full range of choices causes only that subset to be loaded from media. When a user configures the product, the user is presented with the question and the reduced answer set.

user Enter **user** (or just press <RETURN>) to impose no constraints. If the question concerns the installation of a subcomponent or group of related subcomponents, all subcomponents are loaded from media. When a user configures the product, the user is presented with the question and the full set of possible answers.

In response to a question, you can also enter **help** to obtain information about the constraint subcommands; **refresh** to redisplay the question and answers; or **abort** to exit the constraint session. **abort** saves any constraints already defined and returns you to the CFGSA> prompt.

When you finish responding to all the configuration questions, the CFGSA> prompt reappears.

5 Optionally display and change the constraints.

You can enter **show** at the CFGSA> prompt to display all the questions for the product and the constraints you've applied. To change the constraints, enter **revert. revert** removes *all* constraints that you've applied to the selected product during the current **cfgsa** session. You can then redefine constraints with the **constrain** command.

6 Create a selection file and a corresponding override file.

When you finish defining the configuration constraints, create a selection file and a corresponding override file that reflect the constraints. To do this, enter the command

CFGSA> **generate** *name*

where *name* is an identifier of your choice for both the selection and override file. **cfgsa** creates a selection file with the name **aa.***name* and an override file with the name **ov.***name*. Both files are created in the current directory.

If the current directory is the directory that contains the HP-supplied selection and override files for the product, do not set *name* to *product_name*, since this might overwrite one of the HP–supplied files.

7 Exit **cfgsa**.

Select, constrain, and generate selection/override file pairs for as many products as you want. When you finish, enter the command

CFGSA> **exit**

to return to the shell prompt.

Example

In this example, we create a custom subset of SR10.4, which resides in the Authorized Area **//aa/sr10.4**. We define a subset that includes the sys5.3 environment only (it excludes Aegis and BSD4.3); that includes only the **sau** directories and hardware diagnostics for the Series 3500 (**sau7**) and Series 2500 (**sau9**) machines; and that excludes optional fonts.

```
% cd //aa/sr10.4/install/templates/apollo/os.v.10.4
% //aa/sr10.4/install/tools/cfgsa //aa/sr10.4
RAI System Administrator Override Tool V1.01 3 Apr 89
Scanning Authorized Area in //aa/sr10.4 ...

    Products Available in //aa/sr10.4
     1) c++           2.1.0.m
     2) dpcc          3.5
     3) dpcc          3.4
     4) hpvue         1.0
     5) omniback      1.2
     6) os            10.4


CFGSA> select 6
Loading release index for os 10.4 ...

CFGSA> constrain

QUESTION: You may install any combination of the available
          environments:
          aegis   bsd4.3   sys5.3
          Please specify the environments to be installed, or
          'all' for all three.
ANSWERS:  Up to 3 of [ all(D) aegis bsd4.3 sys5.3 ]
```

```
YOUR CHOICE [ Answer Limit User(D) Help Refresh Abort ]:
          answer
Pick 3 of [ all(D) aegis bsd4.3 sys5.3 ]: sys5.3


QUESTION: MACHINE-SPECIFIC SUPPORT
          The sauN directories provide support for Apollo
          machine types:
          sau7  - dn3500, dsp3500, dn4000, dsp4000, dn4500,
                  dsp4500
          sau8  - dn3000, dsp3000
          sau9  - dn2500
          sau10 - dn10000, dsp10000
          sau11 - 9000/425s, 9000/425t
          sau12 - 9000/400s, 9000/400t
          sau14 - dn5500
          You must select the sau(s) for all machine types
          using this installation configuration, including
          the sau(s) for any machines that will boot diskless
          from the target(s).
          Note: Unless this is a cmpexe release, only ISP
          compatible saus are available.
          Please select the set of saus you want (´all´ for
          all saus).
ANSWERS:  Up to 8 of [ all(D) sau7 sau8 sau9 sau10 sau11
          sau12 sau14 ]
YOUR CHOICE [ Answer Limit User(D) Help Refresh Abort ]: limit
Pick 8 of [ all(D) sau7 sau8 sau9 sau10 sau11 sau12 sau14 ]:
          sau7 sau9


QUESTION: HARDWARE DIAGNOSTICS
          Offline hardware diagnostics (in /sauN) are used
          for troubleshooting hardware problems. They do not
          need to be local to every node, but they should be
          available somewhere on the network. Disk space
          requirements differ for different saus.
          Note: Unless this is a cmpexe release, diagnostics
          are only available for ISP compatible saus.
          Would you like offline hardware diagnostics for the
```

```
          selected machine types?
ANSWERS:  Up to 2 of [ yes(D) no ]
YOUR CHOICE [ Answer Limit User(D) Help Refresh Abort ]:
            answer
Pick 2 of [ yes(D) no ]: yes


...


QUESTION: OPTIONAL DM FONTS
          Optional DM fonts include 8-bit (European-based),
          16-bit (Asian-based) fonts, and some additional
          size and typeface versions of existing fonts. The
          optional DM fonts require on the order of 2.5 MB of
          disk space.
          Would you like the set of optional DM fonts?
ANSWERS:  Up to 1 of [ yes(D) no ]
YOUR CHOICE [ Answer Limit User(D) Help Refresh Abort ]:
            answer
Pick 1 of [ yes(D) no ]: no

...


CFGSA> show

For product: os 10.4

...

For: You may install any combination of the available
     environments:
     aegis   bsd4.3   sys5.3
     Please specify the environments to be installed, or
     'all' for all three.
===> Answer forced to be: "sys5.3"


For: You have selected to install more than one environment.
```

```
     You must now select a  PRIMARY ENVIRONMENT  for use on
     the node.
     Please select the primary environment for this
     installation.
===> Answer forced to be: "sys5.3"

...


For: MACHINE-SPECIFIC SUPPORT
     The sauN directories provide support for Apollo machine
     types:
     sau7  - dn3500, dsp3500, dn4000, dsp4000, dn4500,
             dsp4500
     sau8  - dn3000, dsp3000
     sau9  - dn2500
     sau10 - dn10000, dsp10000
     sau11 - 9000/425s, 9000/425t
     sau12 - 9000/400s, 9000/400t
     sau14 - dn5500
     You must select the sau(s) for all machine types using
     this installation configuration, including the sau(s)
     for any machines that will boot diskless from the
     target(s).
     Note: Unless this is a cmpexe release, only ISP
     compatible saus are available.
     Please select the set of saus you want ('all' for all
     saus).
===> Answers limited to: "sau7" "sau9"


For: HARDWARE DIAGNOSTICS
     Offline hardware diagnostics (in /sauN) are used for
     troubleshooting hardware problems. They do not need to
     be local to every node, but they should be available
     somewhere on the network. Disk space requirements differ
     for different saus.
     Note: Unless this is a cmpexe release, diagnostics
     are only available for ISP compatible saus.
     Would you like offline hardware diagnostics for the
```

```
     selected machine types?
===> Answer forced to be: "yes"

...

For: OPTIONAL DM FONTS
     Optional DM fonts include 8-bit (European-based), 16-bit
     (Asian-based) fonts, and some additional size and
     typeface versions of existing fonts. The optional DM
     fonts require on the order of 2.5 MB of disk space.
     Would you like the set of optional DM fonts?
===> Answer forced to be: "no"

...
```

CFGSA> **generate sys5.3_custom**

CFGSA> **exit**

% ls -x

```
aa.aegis_bsd4.3_large    aa.aegis_bsd4.3_medium
aa.aegis_large           aa.aegis_medium
aa.aegis_small           aa.aegis_small_prog
aa.aegis_sys5.3_large    aa.aegis_sys5.3_medium
aa.aegis_bsd4.3_large    aa.bsd4.3_large
aa.bsd4.3_medium         aa.large
aa.sys5.3_custom         aa.sys5.3_large
aa.sys5.3_medium         cf.os
ov.aegis_bsd4.3_large    ov.aegis_bsd4.3_medium
ov.aegis_large           ov.aegis_medium
ov.aegis_small           ov.aegis_small_prog
ov.aegis_sys5.3_large    ov.aegis_sys5.3_medium
ov.bsd4.3_large          ov.bsd4.3_medium
ov.large                 ov.sys5.3_custom
ov.sys5.3_large          ov.sys5.3_medium
```

Step 3. Remove the Product from the Authorized Area

If the version of the product for which you defined a subset was not in the Authorized Area before you began this procedure (meaning you loaded just the product's release index in Step 1), skip this step. If the product was in the Authorized Area before you began this procedure (meaning you skipped Step 1), you must now remove the product from the Authorized Area.

You must be **root** to remove some products, depending on how the access rights for the product's objects are set.

- Remove the product from the Authorized Area, using one of the following commands:

  UNIX environment:

  rm -rf *AA*/**install/ri.apollo.***product_name*.**v.***version*

  Aegis environment:

  dlt -f -du *AA*/**install/ri.apollo.***product_name*.**v.***version*

  where

  AA is the pathname of the Authorized Area.

  product_name is the name of the product.

  version is the version number of the product.

  These commands remove the product directory, which contains the product's release index and source files. The product's release documentation and predefined selection and override files, which reside in different directories, are not removed.

Examples

To remove SR10.4 from the Authorized Area **//aa/sr10.4**,

Using UNIX,

 # rm -rf //aa/sr10.4/install/ri.apollo.os.v.10.4

Using Aegis,

 # dlt -f -du //aa/sr10.4/install/ri.apollo.os.v.10.4

Step 4. Load the Product Subset into the Authorized Area

You can now load the product subset from distribution media into the Authorized Area. You do this by supplying the name of the selection file you created with **cfgsa** (in Step 2) as a command-line argument to the **distaa** tool.

1 Insert the first product tape or floppy in the drive, or mount the first magnetic tape, if you have not already done so.

2 Load the product subset from distribution media into the Authorized Area.

Load the product subset by entering the command

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

where

AA is the pathname of the Authorized Area.

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

selection_file is the pathname of the selection file you created with the **cfgsa** tool in Step 2.

For more information about **distaa**, see "distaa" in Chapter 10.

Example

To load the custom subset that we defined for SR10.4 in the Step 2 example,

```
% //aa/sr10.4/install/tools/distaa -m c //aa/sr10.4
  //aa/sr10.4/install/templates/apollo/os.v.10.4/aa.sys5.3_custom

.
.
.

Restore complete.
```

Step 5. Activate the Override File

You must now *activate* the override file that you created with **cfgsa** in Step 2. This causes the configuration options for the product to be consistent with the reduced set of product components. For example, if you excluded a subcomponent from the product, the configuration question (displayed by the **config** tool) that asks if the user wants to install this component does not appear. You activate the override file by copying it to the **install/overrides** directory in the Authorized Area, and giving it the same name as the product's release index.

- Activate the override file, using one of the following commands:

  UNIX environment:

  cp *override_file AA*/**install/overrides/ri.apollo.***product_name***.v.***version*

  Aegis environment:

  cpf *override_file AA*/**install/overrides/ri.apollo.***product_name***.v.***version*

  where

  override_file is the pathname of the override file that you created with the **cfgsa** tool in Step 2.

  product_name is the name of the product.

  version is the version number of the product.

Example

To activate the override file that we created for SR10.4 in the Step 2 example,

Using UNIX,

```
% cp //aa/sr10.4/install/templates/apollo/os.v.10.4/aa.sys5.3_custom
  //aa/sr10.4/install/overrides/ri.apollo.os.v.10.4
```

Using Aegis,

```
$ cpf //aa/sr10.4/install/templates/apollo/os.v.10.4/aa.sys5.3_custom
  //aa/sr10.4/install/overrides/ri.apollo.os.v.10.4
```

Merging Products in an Authorized Area

How to use the **mrgri** tool to merge patches with products, PSKs with Domain/OS, and the a88k and m68k versions of a product

With the **mrgri** tool, you can combine, or merge, products in a Authorized Area. That is, you can merge two product directories in a Authorized Area into a single product directory with a single release index. You can then configure and install the merged product as you would any other product.

You can use **mrgri** to merge three types of products:

- Patches (bug fixes) with the products they patch.
- Product Support Kits (PSKs) with Domain/OS.
- The a88k and m68k versions of a product, creating a **compound product**.

In all cases, merging products is an entirely optional process.

Why Merge?

The primary advantage of merging products is that you only have one product in the Authorized Area to manage and install, rather than two. For example, rather than installing a product and then a patch for that product on a number of nodes, you can merge the patch with the product once in an Authorized Area, and then install the patched product. Compound products provide additional advantages, as discussed in the "Merging a88k and m68k Products" section later in this chapter.

Merge Only Designated Products

You cannot indiscriminately merge any two products. For two products to merge successfully, or for a merged product to install and run successfully, the products must be designed to be merged. Before you merge a product, check the product release notes to determine whether the product can be merged, which products and which versions it can be merged with, and what restrictions you must follow, if any,

when installing and using the merged product. The **mrgri** tool does *not* check that products have been designed to be merged.

Creating a New Product versus Overwriting a Product

When you merge two products in a disk Authorized Area, you can overwrite objects in one of the products with objects from the other. Or you can create the merged product as a separate, third product (in the same or another Authorized Area), retaining both of the constituents products in the Authorized Area. Overwriting one of the products saves disk space and takes less time. However, if you want to install the overwritten product after the merge, you must reload it from media — you cannot unmerge a product to recover its constituent products.

In the following sections, we provide some specific recommendations about whether to create a third product or overwrite one of the constituent products. Ultimately, however, the particular needs of your site should determine precisely how you merge any two products.

Merging a Patch with a Product

In the Domain installation model, each patch is a separate product with its own release index and product directory. After you load a patch from distribution media into an Authorized Area, you can configure and install the patch on a target node, just as you would an optional product.

Alternately, you can use the following procedure to merge the patch and the product it patches in an Authorized Area, and then install the resulting patched product. This 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 and product is that you only have one product in the Authorized Area to manage and install, rather than two.

This procedure is entirely optional. It assumes that both the patch and the product it patches already reside in a Authorized Area. Also, the patch and product must reside in the same Authorized Area.

1 Merge the patch with the product.

Merge the patch with the product by entering the command

AA/**install/tools/mrgri** **–merge** **–s** *AA* **ri.apollo.***product_name***.v.***version* **ri.apollo.***patch_name***.v.***version*

where

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

product_name is the name of the product to be patched.

version is the version number of the product and the patch, respectively. The patch version number is most always **1.0**.

patch_name is the name of the patch.

The installation chapter (usually Chapter 2) in the patch release notes provides the names and version numbers of the patches. This is a suggested command-line only. For a complete listing of **mrgri**'s command-line options, see "mrgri" in Chapter 10.

This command does not create the patched product as a separate, third product. Rather, it copies the files comprising the patch to the product directory of the unpatched product and overwrites the original versions of these files. The resulting patched product has the same name and version number as the unpatched product.

It is possible, using the **-p**, **-t**, or **-v** option, to create the patched product as a new, third product, retaining both the patch and the unpatched product in the Authorized Area. However, this uses more disk space and takes considerably longer. For example, if you merge a patch with Domain/OS and create the patched product as a third product, the merge can take hours, since **mrgri** must copy the entire OS. If you overwrite Domain/OS with the patch, the merge takes only a few minutes. If for some reason you need the unpatched product after the merge, you can always reload it from distribution media.

2 Optionally remove the patch from the Authorized Area.

The patch remains in the Authorized Area. Once you've successfully merged the patch with the product, you may want to remove the patch from the Authorized Area to save disk space and unclutter the Authorized Area.

To remove the patch, use one of the following commands:

UNIX environment:

rm -rf *AA*/**install/ri.apollo.**_patch_name_**.v.**_version_

Aegis environment:

dlt -f *AA*/**install/ri.apollo.**_patch_name_**.v.**_version_

The variables *AA*, *patch_name*, and *version* have the same meaning as in Step 1. These commands remove the entire product directory, which contains the patch and the release index for the patch.

Example

In the following example we merge patch **patch_m0146**, which contains five **gmrlib** files, with version 2.7 of the product **gmr3d** in the Authorized Area **//aa**. We then remove the patch from the Authorized Area. We use the optional **-i** switch to display informational messages during the merge.

```
% //aa/install/tools/mrgri -merge -s //aa -i ri.apollo.gmr3d.v.2.7 ri.apollo.patch_m0146.v.1.0

RAI Merge Tool V0.39 06 July 90
Loading primary product:
     //aa/install/ri.apollo.gmr3d.v.2.7/ri.apollo.gmr3d.v.2.7
Loading secondary product:
     //aa/install/ri.apollo.patch_m0146.v.1.0/ri.apollo.patch_m0146.v.1.0
Merging secondary product:
     //aa/install/ri.apollo.patch_m0146.v.1.0/ri.apollo.patch_m0146.v.1.0
Copying from //aa/install/ri.apollo.patch_m0146.v.1.0/lib/gmr3dlib
        to   //aa/install/ri.apollo.gmr3d.v.2.7/lib/gmr3dlib
Copying from //aa/install/ri.apollo.patch_m0146.v.1.0/lib/gmr3dlib.460
        to   //aa/install/ri.apollo.gmr3d.v.2.7/lib/gmr3dlib.460
Copying from //aa/install/ri.apollo.patch_m0146.v.1.0/lib/gmr3dlib.881
        to   //aa/install/ri.apollo.gmr3d.v.2.7/lib/gmr3dlib.881
Copying from //aa/install/ri.apollo.patch_m0146.v.1.0/lib/gmr3dlib.fpa
        to   //aa/install/ri.apollo.gmr3d.v.2.7/lib/gmr3dlib.fpa
Copying from //aa/install/ri.apollo.patch_m0146.v.1.0/lib/gmr3dlib.peb
        to   //aa/install/ri.apollo.gmr3d.v.2.7/lib/gmr3dlib.peb
mrgri completed successfully.

% rm -rf //aa/ri.apollo.patch_m0146.v.1.0
```

Merging a PSK with Domain/OS

A PSK (Product Support Kit) is a subset of Domain/OS files that provides support for a new hardware platform or peripheral device, or that provides new Domain/OS functionality. As with patches, each PSK is released and handled as an individual product. After you load a PSK from tape into an Authorized Area, you can configure and install the PSK on a node running the appropriate version of Domain/OS, as you would an optional product.

Alternately, you can use the following procedure to merge the PSK with the appropriate version of Domain/OS in an Authorized Area, and then install the merged product. This has the same result as would installing Domain/OS first and then installing the PSK on top of it. The advantage of merging the PSK with Domain/OS is that you only have one product in the Authorized Area to manage and install, rather than two.

This procedure is entirely optional. It assumes that both the PSK and Domain/OS already reside in a Authorized Area. Also, the PSK and Domain/OS must reside in the same Authorized Area.

1 Merge the PSK with Domain/OS.

Merge the PSK with Domain/OS, by entering the command

AA/**install/tools/mrgri -merge -s** *AA* **ri.apollo.os.v.***version*
ri.apollo.*psk_name***.v.***version*

where

AA is the pathname of the Authorized Area containing the product and the PSK.

version is the version number of Domain/OS and the PSK, respectively. The PSK version number is usually the same as the Domain/OS version number.

psk_name is the product name of the PSK.

The installation chapter in the PSK release notes (usually Chapter 2) identifies the PSK product name and version number. Make sure you put the Domain/OS product before the PSK in the command line; the order *is* significant. This is a suggested command-line only. For a complete listing of **mrgri**'s command-line options, see "mrgri" in Chapter 10.

This command does not create the merged product as a separate, third product. Rather, it copies the PSK files to the Domain/OS product directory and overwrites the appropriate Domain/OS files with the PSK files. The resulting product has the same name and version number as the original Domain/OS product.

It is possible, using the **-p**, **-t**, or **-v** options, to create the merged product as a new, third product, retaining both the PSK and the original version of Domain/OS in the Authorized Area. However, this uses more disk space and takes considerably longer. If for some reason you need the original version of Domain/OS after the merge, you can reload it from distribution media into the Authorized Area.

2 Optionally remove the PSK from the Authorized Area.

The PSK remains in the Authorized Area. Once you've successfully merged the PSK with Domain/OS, you may want to remove the PSK from the Authorized Area to save disk space and unclutter the Authorized Area.

To remove the PSK, use one of the following commands:

UNIX environment:

rm -rf *AA*/**install/ri.apollo.**_psk_name_**.v.**_version_

Aegis environment:

dlt -f *AA*/**install/ri.apollo.**_psk_name_**.v.**_version_

The variables *AA*, *psk_name*, and *version* have the same meaning as in Step 1. These commands remove the entire product directory containing the PSK and the PSK's release index. If for some reason you need the PSK after you remove it, you can reload it from distribution media into the Authorized Area.

Example

In the following example, we merge **pskq2_91** with version 10.3 of Domain/OS in the Authorized Area **//aa**. We then remove the PSK from the Authorized Area.

```
% //aa/install/tools/mrgri -merge -s //aa ri.apollo.os.v.10.3
  ri.apollo.pskq2_91.v.10.3

RAI Merge Tool V0.39 06 July 90
mrgri completed successfully.

% rm -rf //aa/ri.apollo.pskq2_91.v.10.3
```

Merging a88k and m68k Products

Most Domain products are released in two versions: a version that runs on workstations based on Motorola's 68000 series of microprocessors, and a version that runs on the Series 10000 workstation. Two versions are required because the Series 10000 workstation has a different ISP (Instruction Set Processor) type, called PRISM (Parallel Reduced Instruction Set Multiprocessor). We refer to the Series 10000 ISP type as a88k, and the Motorola 68000 ISP type as m68k.

The following procedure describes how to use the **mrgri** tool to merge the a88k and m68k versions of a product into a single product in a Authorized Area. The resulting product, called a **compound product**, can be installed and run on nodes of either ISP type. Also, if you create and install a compound version of Domain/OS, nodes of either ISP type can boot diskless off the node running the compound OS. And you can link Domain/OS components on nodes of either ISP type to the compound node.

This procedure is entirely optional. It assumes that both products already reside in a Authorized Area. Also, both products must reside in the same Authorized Area.

1 Check that the products are mergeable.

Check the product release notes to determine whether the products can be merged, and what restrictions, if any, apply to installing and using the merged product. The **mrgri** tool does *not* check to ensure that the products are designed to be merged. If they aren't, the results of running **mrgri** and installing and running the merged products are unpredictable.

2 Check available disk space.

The size of a compound product is about 70 percent of the sum of the two constituent products. Use the Aegis **lvolfs** or the UNIX **df** command to make sure you have enough space on the Authorized Area node for the compound product.

When assessing disk space, note that you can use the **mrgri -t** option to create the compound product in an Authorized Area on another node, rather than in the one containing the constituent products. Also, to save disk space, you can overwrite one of the products in the Authorized Area to create the compound version, rather than retain both the a88k and m68k versions in the Authorized Area. However, if you subsequently need to install the overwritten product, you must reload it from distribution media. The command line in the next step retains the original a88k and m68k products.

3 Merge the a88k and m68k products.

To merge the products, enter the command

AA/**install/tools/mrgri -cmpexe -s** *AA* **-v** *a88k_version*.**cmpexe**
ri.apollo.*product*.**v.***a88k_version* **ri.apollo.***product*.**v.***m68k_version*

where

AA is the pathname of the Authorized Area containing the a88k and m68k versions of the product.

a88k_version is the version number of the a88k version of the product. a88k products are identified by the extension **.p** on the version field of the product's release index name. For example, the release index name for the a88k version of SR10.4 is **ri.apollo.os.v.10.4.p**; the m68k version is **ri.apollo.os.v.10.4**.

product is the name of the product.

m68k_version is the version number of the m68k version of the product.

This is a suggested command-line only. For a complete listing of **mrgri**'s command-line options, see "mrgri" in Chapter 10.

This command uses the **-v** switch to create the compound product as a third product; the original a88k and m68k versions are retained in the Authorized Area. The name of the compound product is the same as the a88k and m68k products. Its version number is the a88k version number with the extension **.cmpexe** added to it. We explain why we recommend this naming scheme in the "mrgri" section in Chapter 10.

Merging the a88k and m68k versions of a large, complex product, such as Domain/OS, can take a very long time. We've measured times of about 12 hours for merging two complete Domain/OS products.

Example

In the following example, we merge the m68k and a88k version of SR10.4 in the Authorized Area //**aa**. We create the compound product as a separate, third product in the same Authorized Area and give it the release index name **ri.apollo.os.v.10.4.p.cmpexe**.

```
% //aa/install/tools/mrgri -cmpexe -s //aa -v 10.4.p.cmpexe
   ri.apollo.os.v.10.4.p ri.apollo.os.v.10.4

RAI Merge Tool V0.39 06 July 90
mrgri completed successfully.
```

Manipulating Authorized Areas

How to copy, move, distribute, remove, remove products from, and load tools into an Authorized Area

This chapter describes how to do some basic things with Authorized Areas and their components. Specifically, it describes how to

- Copy or move an Authorized Area.
- Distribute an Authorized Area among more than one disk using links.
- Remove an Authorized Area.
- Remove a product from an Authorized Area.
- Load installation tools from distribution media into an Authorized Area.

These tasks enable you to use and manage Authorized Areas more effectively and, in many cases, save disk space. All of the tasks are optional. Also see Chapter 5, which describes how to load products from distribution media into an Authorized Area, and Chapter 11, which describes the components and structure of an Authorized Area.

Just a Directory

The key concept in learning to manipulate Authorized Areas and their contents effectively is that an Authorized Area is not a special construct, but just a directory like any other. Its only distinguishing feature is that it contains an **install** subdirectory, which in turn contains other subdirectories. Accordingly, all of the tasks in this chapter are performed with standard Aegis and UNIX commands, such as **cpt**, **dlt**, **cp**, and **rm**.

A Complicating Factor

The only case where you must exercise special caution is when an Authorized Area is located at a node entry directory (*//node_name*/**install**/...). *Every* node contains an **install** directory at the node entry level (*//node_name*/**install**). This directory contains objects that are not part of an Authorized Area, such as a node's baseline files (in the **/install/baseline** directory), the **not_installed** file, and the **preserve.list** file (see "install," Chapter 10). These objects are created or used by the **install** program when it installs products on the node, and are unique to each node. The baseline files are required for proper installation of products on the node.

When an Authorized Area is located at a node entry directory, the node's **/install** directory doubles as the **install** directory of the Authorized Area. Therefore, when you perform certain tasks with a node-entry Authorized Area, such as copy or remove it, you must take special steps to ensure that the non–Authorized Area components in the node's **/install** directory are preserved.

Preserving ACLs

Also be sure to preserve Access Control Lists (ACLs) when you copy or move an Authorized Area or the installation tools, by using the appropriate command-line options. The ACLs of the installation tools set the owner to **root** and turn the **setuid** bit on. This causes the effective user ID of the processes to be **root**. If you don't preserve these ACLs, the copied tools may not have sufficient access privileges to perform some tasks.

Creation of Authorized Areas

Notice that we don't have a task in this chapter for creating an Authorized Area. This is because an Authorized Area is created automatically when you load and install Domain/OS on an invol'ed disk from media (Chapter 1). This Authorized Area contains the installation tools and their help files; the Domain/OS product; and the release documentation, selection files, and override files associated with Domain/OS.

You can create other Authorized Areas in the network by simply copying this original Authorized Area or parts of it, using the copy procedure in this chapter. You can also create a new Authorized Area when you load a Domain/OS update or PSK from media onto an existing, non-initialized node, using the load procedure in Chapter 5.

Copying or Moving an Authorized Area

This section describes how to copy or move an Authorized Area from one location (directory) to another, using standard UNIX and Aegis commands. You can copy or move the Authorized Area to a directory on the same disk or another node's disk. The destination directory can be another existing Authorized Area or a directory that is not an Authorized Area. If the destination is another Authorized Area, the contents of the two Authorized Areas are combined.

The only complicating factor is if you copy or move an Authorized Area that is at a node entry directory (*//node_name*/**install**/...), since *every* node has a (*//node_name*/**install**) directory that contains non–Authorized Area objects. Accordingly, we provide three procedures:

- Copying an Authorized Area that is not at a node entry directory to a directory that also is not a node entry directory (the simplest case).
- Copying an Authorized Area from one node entry directory to another.
- Copying an Authorized Area from a node entry directory to a directory that is not a node entry directory.

You can easily adapt these procedures to copy or move subdirectories of an Authorized Area, such as the **install/tools** directory.

When you copy an Authorized Area, be sure to preserve the Access Control Lists (ACLs) of the Authorized Area objects. (The command lines in this section provide the appropriate options for doing this.) The ACLs of the installation tools specify the owner as **root** and turn the **setuid** bit on, which causes the effective user ID of the processes to be **root**. If you don't preserve these ACLs, the copied tools may not have sufficient access privileges to perform some tasks.

If you use these procedures to set up a new Authorized Area, we recommend that you do *not* locate the new Authorized Area at a node entry directory and that you create a new directory to be used exclusively for the Authorized Area (*//node_name*/**aa_m68k**, for example).

These procedures describe how to *copy* an Authorized Area. To *move* an Authorized Area, use the appropriate copy procedure in this section. Then remove the source Authorized Area, using the "Removing an Authorized Area" procedure later in this chapter.

- To copy an Authorized Area that is *not* at a node entry directory to another directory or Authorized Area that is not a node entry directory (for example, from **//node_1/aa** to **//node_2/another_aa**), execute the following command:

  UNIX environment:

  cp -rpsoP *source_AA*/**install** *target_AA*

  Aegis environment:

  cpt -pdt -sacl -md *source_AA*/**install** *target_AA*/**install**

  where

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

  target_AA is the pathname of the directory you want to copy the Authorized Area to. This can be the pathname of an existing Authorized Area or a directory that is not an Authorized Area. If *target_AA* is an Authorized Area, the contents of the two Authorized Areas are combined. If *target_AA* is not an Authorized Area, *target_AA* must already exist, but the **install** subdirectory doesn't have to. (If *target_AA* doesn't exist, the UNIX **cp** command creates the directory and copies objects without error, but does not create the **install** subdirectory. The Aegis **cpt** command returns an error.)

- To copy an Authorized Area at a node entry directory (*//node_name*) to another node entry directory (whether or not an Authorized Area already exists there), execute the following commands:

  UNIX environment:

  mv *//target*/**install/baseline** *//target*/**install/baseline.tmp**
  mv *//target*/**install/not_installed** *//target*/**install/not_installed.tmp**
  mv *//target*/**install/preserve.list** *//target*/**install/preserve.list.tmp**
  cp -rpsoP *//source*/**install** *//target*
  rm -rf *//target*/**install/baseline**
  mv *//target*/**install/baseline.tmp** *//target*/**install/baseline**
  mv *//target*/**install/not_installed.tmp** *//target*/**install/not_installed**
  mv *//target*/**install/preserve.list.tmp** *//target*/**install/preserve.list**

Aegis environment:

chn *//target*/**install/baseline baseline.tmp**
cpt -pdt -sacl -md *//source*/**install** *//target*/**install**
dlt -f *//target*/**install/baseline**
chn *//target*/**install/baseline.tmp baseline**

where

target is the name of the node you are copying the Authorized Area to.

source is the name of the node you are copying the Authorized Area from.

These commands preserve the following non-Authorized Area objects in the target node's **/install** directory: the baseline files (in the **/install/baseline** directory), the **preserve.list** file, and the **not_installed** file. (For more information about these objects, see "install," Chapter 10.) To preserve the objects, the commands rename the objects before the Authorized Area is copied and then restore the objects' original names after the copy.

The UNIX commands we provide assume the **preserve.list** file (optionally created by the user) and the **not_installed** file (created by the **install** tool only when it fails to install objects on the node) exist on both the source and the target. If one of these files exists only on the target, or exists on neither the source or the target, you don't have to execute the move (**mv**) commands for the file, but doing so does no harm. If, however, one of the files exists on the source but not the target, you should remove (**rm**) the file from the target's **/install** directory after you copy the Authorized Area.

The Aegis commands work fine whether or not the **preserve.list** and **not_installed** files exist, with one exception. If one of the files exists on the source but not on the target, you should delete the file (using **dlf**) from the target's **/install** directory after you copy the Authorized Area.

- To copy an Authorized Area at a node entry directory (*//node_name*) to a directory or Authorized Area that is not a node entry directory (for example, from **//node_1** to **//node_2/aa**), execute the following commands:

  UNIX environment:

  cp -rpsoP *//source*/**install** *target_AA*
  rm -rf *target_AA*/**install/baseline**
  rm *target_AA*/**install/not_installed**
  rm *target_AA*/**install/preserve.list**

Aegis environment:

cpt -pdt -sacl -md *//source*/**install** *target_AA*/**install**
dlt -f *target_AA*/**install/baseline**
dlf *target_AA*/**install/not_installed**
dlf *target_AA*/**install/preserve.list**

where

source is the name of the node you are copying the Authorized Area from.

target_AA is the pathname of the directory you want to copy the Authorized Area to. This can be the pathname of an existing Authorized Area or a directory that is not an Authorized Area. If *target_AA* is an Authorized Area, the contents of the two Authorized Areas are combined. If *target_AA* is not an Authorized Area, *target_AA* must already exist, but the **install** subdirectory doesn't have to. (If *target_AA* doesn't exist, the UNIX **cp** command creates the directory and copies objects without error, but does not create the **install** subdirectory. The Aegis **cpt** command returns an error.)

When you copy the Authorized Area, non-Authorized Area components — the **/install/baseline** directory, the **/install/preserve.list** file, and the **/install/not_installed** file — are copied along with it. (For more information about these objects, see "install," Chapter 10.) The commands we provide remove these objects from the target Authorized Area, since the objects are superfluous there.

The **preserve.list** file (optionally created by the user) and the **not_installed** file (created by the **install** tool only when it fails to install objects on the node) do not necessarily exist in the source node's **/install** directory. But executing the remove commands if they don't does no harm.

Examples

To copy an Authorized Area at **//node_1/aa** to **//node_2/aa**, assuming the **aa** directory on **//node_2** does not exist initially,

Using UNIX,

```
% cd //node_2
% mkdir aa
% cp -rpsoP //node1/aa/install //node_2/aa
```

Using Aegis,

```
$ wd //node_2
$ crd aa
$ cpt -pdt -sacl -md //node_1/aa/install //node_2/aa/install
```

To copy an Authorized Area at the node entry directory of **//node_1** to the node entry directory of **//node_2**, assuming the files **//node_2/install/preserve.list** and **//node_2/install/not_installed** exist,

Using UNIX,

```
% mv //node_2/install/baseline //node_2/install/baseline.tmp
% mv //node_2/install/not_installed //node_2/install/not_installed.tmp
% mv //node_2/install/preserve.list //node_2/install/preserve.list.tmp
% cp -rpsoP //node_1/install //node_2
% rm -rf //node_2/install/baseline
% mv //node_2/install/baseline.tmp //node_2/install/baseline
% mv //node_2/install/not_installed.tmp //node_2/install/not_installed
% mv //node_2/install/preserve.list.tmp //node_2/install/preserve.list
```

Using Aegis,

```
$ chn //node_2/install/baseline baseline.tmp
$ cpt -pdt -sacl -md //node_1/install //node_2/install
$ dlt -f //node_2/install/baseline
$ chn //node_2/install/baseline.tmp baseline
```

To copy an Authorized Area at the node entry directory of **//node_1** to the directory **//node_2/aa**, assuming the directory **aa** already exists,

Using UNIX,

```
% cp -rpsoP //node_1/install //node_2/aa
% rm -rf //node_2/aa/install/baseline
% rm //node_2/aa/install/not_installed
% rm //node_2/aa/install/preserve.list
```

Using Aegis,

```
$ cpt -pdt -sacl -md //node_1/install //node_2/aa/install
$ dlt -f //node_2/aa/install/baseline
$ dlf //node_2/aa/install/not_installed
$ dlf //node_2/aa/install/preserve.list
```

Distributing an Authorized Area with Links

The objects in an Authorized Area do not have to physically reside on the disk associated with that Authorized Area. Using standard UNIX and Aegis commands, you can replace objects in an Authorized Area with soft (symbolic) links to another directory or Authorized Area in the network. Replacing objects with links allows you to build an Authorized Area that is virtually larger than the space you want to commit from a single disk, and to build multiple Authorized Areas without duplicating objects in them.

The procedure in this section describes how to replace a product directory in an Authorized Area with a link. A product directory contains a product's release index and the objects that actually make up the product. Because of their relatively large space requirements, product directories are the most likely candidates for linking. You can easily adapt this procedure to distribute other Authorized Area components (the **install/doc** directory, for example) with links.

The only objects you should *not* distribute with links are subcomponents of a product. If you want to distribute product subcomponents, you must do it when you load the product from distribution media (see "Loading Products from Media into an Authorized Area" in Chapter 5).

You may have to be **root** to perform this procedure, depending on how the product's objects are protected.

1 Copy the product directory, if necessary.

Copy the product directory that you want to replace with a link to the desired location, if the same product directory does not already reside there. Use one of the following commands:

UNIX environment:

cp –rpsoP *source_AA*/**install/ri.apollo.***product_name***.v.***version* *target_directory*

Aegis environment:

cpt –pdt –sacl *source_AA*/**install/ri.apollo.***product_name***.v.***version*
target_directory/**ri.apollo.***product_name***.v.***version*

where

source_AA is the pathname of the Authorized Area containing the product directory that you want to replace with a link.

product_name is the name of the product.

version is the version number of the product.

target_directory is the pathname of the directory where you want the product directory to reside. If you want to use the target directory simply as extended storage space for the source Authorized Area and not as another location from which to install the product, the target directory can be any directory (*//node_name* or *//node_name*/**aa_extender**, for example). If you want to be able to install the product from its new location — to use the new location as another source Authorized Area for the product — the target directory must be an **install** subdirectory. This can be the **install** subdirectory of another, existing Authorized Area (enter *authorized_area_name*/**install** as the *target_directory*) or an **install** subdirectory that you create expressly for the purpose of holding the product. In all cases, *target_directory* must exist before you copy the product directory.

2 Remove the product directory from the source Authorized Area.

UNIX environment:

rm -rf *source_AA*/**install/ri.apollo.***product_name***.v.***version*

Aegis environment:

dlt -f -du *source_AA*/**install/ri.apollo.***product_name***.v.***version*

The variables in these commands have the same meaning as in Step 1 and throughout the rest of this procedure.

3 Create a link from the source Authorized Area to the product directory at the target.

UNIX environment:

ln -s *target_directory*/**ri.apollo.***product_name***.v.***version*
source_AA/**install/ri.apollo.***product_name***.v.***version*

Aegis environment:

crl *source_AA*/**install/ri.apollo.***product_name***.v.***version*
target_directory/**ri.apollo.***product_name***.v.***version*

You can replace a product directory with a soft link even if the product is installed on the Authorized Area node with hard links to the Authorized Area. However, the space saved is minimal.

If you are replacing a product directory that contains a product subset (an **aegis_small** configuration of Domain/OS, for example) with a link, you can link to a product directory that contains the same product subset or a superset (an **aegis_large** configuration, for example). However, do not link to a product directory that contains a smaller configuration than the source subset.

Examples

In the following example, we replace the SR10.4 product directory in the Authorized Area **//node_1/aa** with a link to an SR10.4 product directory in the node entry directory of **//node_2**. We cannot install SR10.4 from **//node_2**, since the SR10.4 product directory does not reside in an **install** subdirectory.

Using UNIX,

```
% cp -rpsoP //node_1/aa/install/ri.apollo.os.v.10.4 //node_2
% rm -rf //node_1/aa/install/ri.apollo.os.v.10.4
% ln -s //node_2/ri.apollo.os.v.10.4 //node_1/aa/install/ri.apollo.os.v.10.4
```

Using Aegis,

```
$ cpt -pdt -sacl //node_1/aa/install/ri.apollo.os.v.10.4
  //node_2/ri.apollo.os.v.10.4
$ dlt -f -du //node_1/aa/install/ri.apollo.os.v.10.4
$ crl //node_1/aa/install/ri.apollo.os.v.10.4 //node_2/ri.apollo.os.v.10.4
```

In the following example, we replace the SR10.4 product directory in the Authorized Area **//node_1/aa** with a link to the product directory in the existing Authorized Area **//node_2/aa_2**.

Using UNIX,

```
% cp -rpsoP //node_1/aa/install/ri.apollo.os.v.10.4 //node_2/aa_2/install
% rm -rf //node_1/aa/install/ri.apollo.os.v.10.4
% ln -s //node_2/aa_2/install/ri.apollo.os.v.10.4
  //node_1/aa/install/ri.apollo.os.v.10.4
```

Using Aegis,

```
$ cpt -pdt -sacl //node_1/aa/install/ri.apollo.os.v.10.4
  //node_2/aa_2/install/ri.apollo.os.v.10.4
$ dlt -f -du //node_1/aa/install/ri.apollo.os.v.10.4
$ crl //node_1/aa/install/ri.apollo.os.v.10.4
  //node_2/aa_2/install/ri.apollo.os.v.10.4
```

Removing an Authorized Area

This section describes how to remove an Authorized Area entirely. This is a radical step that you seldom need to do. However, you may want to remove a corrupted Authorized Area or a duplicate Authorized Area that is no longer needed. Also, when you load and install Domain/OS on an invol'ed node from media (Chapter 1), an Authorized Area containing Domain/OS and the installation tools is always created on the node. If you already have an Authorized Area in your network, the one created on the invol'ed node may not be needed.

This section provides two procedures for removing an Authorized Area: one for removing an Authorized Area that is not at a node entry directory (*//node_name*/**aa/install/**..., for example), and one for removing an Authorized Area that is at a node entry directory (*//node_name*/**install/**...). The latter case requires special steps to ensure that the non–Authorized Area components in the node's **/install** directory are preserved. (For more information about these components, see "install," in Chapter 10.)

If Domain/OS or optional products on the Authorized Area node are installed with hard links to the Authorized Area, you can still remove the the Authorized Area. However, the space saved is minimal.

You may have to be **root** to remove an Authorized Area, depending on the contents of the Authorized Area and how they are protected.

- To remove an Authorized Area that is not at a node entry directory, execute one of the following commands:

  UNIX environment:

  rm –rf *AA*/**install**

  Aegis environment:

  dlt –f –du *AA*/**install**

  where

  AA is the pathname of the Authorized Area.

These commands remove the **install** subdirectory that holds all the objects comprising an Authorized Area, but they do not remove the Authorized Area directory (*AA*) itself. If *AA* contains only the **install** subdirectory, you may want to remove *AA* as well. You can do this by specifying *AA* on the command line instead of *AA*/**install**.

- To remove an Authorized Area that is at a node entry directory (*//node_name*/**install** ...), execute the following commands:

  UNIX environment:

  mkdir *//node_name*/**install.tmp**
  mv *//node_name*/**install/baseline** *//node_name*/**install.tmp/baseline**
  mv *//node_name*/**install/doc** *//node_name*/**install.tmp/doc**
  mv *//node_name*/**install/not_installed** *//node_name*/**install.tmp**
  mv *//node_name*/**install/preserve.list** *//node_name*/**install.tmp**
  rm -rf *//node_name*/**install**
  mv *//node_name*/**install.tmp** *//node_name*/**install**

  Aegis environment:

  crd *//node_name*/**install.tmp**
  mvf *//node_name*/**install/baseline** *//node_name*/**install.tmp**
  mvf *//node_name*/**install/doc** *//node_name*/**install.tmp**
  mvf *//node_name*/**install/not_installed** *//node_name*/**install.tmp**
  mvf *//node_name*/**install/preserve.list** *//node_name*/**install.tmp**
  dlt -f -du *//node_name*/**install**
  chn *//node_name*/**install.tmp** *//node_name*/**install**

  These commands preserve the node's **/install** directory and the non-Authorized Area objects in it: the baseline files (in the **/install/baseline** directory), the **preserve.list** file, and the **not_installed** file. The **preserve.list** file (optionally created by the user) and the **not_installed** file (created by the **install** tool when it fails to install objects on the node) do not necessarily exist. But executing the commands if they don't does no harm.

The commands also preserve the node's **/install/doc** directory. Since this directory contains release documentation for all products in the Authorized Area, as well as release documentation for every product installed on the node, you may not want to preserve it. If you don't, you can check the node's latest baseline file to determine which products are installed, and then manually copy the release documentation for just those products from another Authorized Area to the node.

For more information about the non–Authorized Area objects in a node's **/install** directory, see "install" in Chapter 10.

Examples

To remove an Authorized Area named **//server/aa**, where **aa** contains only Authorized Area objects:

Using UNIX,

```
rm -rf //server/aa
```

Using Aegis,

```
dlt -f -du //server/aa
```

To remove an Authorized Area at the node entry directory of **//server**, assuming the files **//server/install/preserve.list** and **//server/install/not_installed** do not exist:

Using UNIX,

```
mkdir //server/install.tmp
mv //server/install/baseline //server/install.tmp/baseline
mv //server/install/doc //server/install.tmp/doc
rm -rf //server/install
mv //server/install.tmp //server/install
```

Using Aegis,

```
crd //server/install.tmp
mvf //server/install/baseline //server/install.tmp
mvf //server/install/doc //server/install.tmp
dlt -f -du //server/install
chn //server/install.tmp //server/install
```

Removing Products from an Authorized Area

This section describes how to remove a product and its associated components from an Authorized Area, using standard UNIX and Aegis commands. Removing old, unused products saves disk space, unclutters your Authorized Area, and makes it easier for users to select the appropriate versions of products when they create product configurations (with **config** or **install + +**). However, only remove a product from an Authorized Area when no one in your network is using, or anticipates using, the particular version of the product. See the "Notes" later in this section for more information about when *not* to remove products.

You may have to be **root** to remove some products, depending on how the product's objects are protected.

- To remove a product and its associated components from an Authorized Area, execute the following commands:

  UNIX environment:

  rm -rf *AA*/**install/ri.apollo.***product_name*.**v.***version*
  rm -rf *AA*/**install/templates/apollo/***product_name*.**v.***version*
  rm *AA*/**install/doc/apollo/***product_name*.**v.***version**
  rm *AA*/**install/overrides/ri.apollo.***product_name*.**v.***version*

  Aegis environment:

  dlt -f -du *AA*/**install/ri.apollo.***product_name*.**v.***version*
  dlt -f -du *AA*/**install/templates/apollo/***product_name*.**v.***version*
  dlf -f -du *AA*/**install/doc/apollo/***product_name*.**v.***version**
  dlf -f -du *AA*/**install/overrides/ri.apollo.***product_name*.**v.***version*

  where

  AA is the pathname of the Authorized Area containing the product.

  product_name is the name of the product (for example, **pas**).

  version is the version of the product (for example, **8.8.m**).

These commands remove the product directory, which contains the product's release index and source files; the product's **templates** directory, which contains the product's predefined selection files, override files, and configuration file; the product's release documentation; and the product's active override file, if one exists.

Note:

- When you load a newer version of a product into an Authorized Area, keep at least the release index of the earlier version in the Authorized Area. This is because when you install a later version of a product on a node that has an earlier version installed, the **install** program uses the release index of the earlier version to properly change local copies of directories to links (and vice versa) and to correctly handle product customization.
- To deinstall a product or product subcomponents, at least the release index of *all* products installed on the target node (the node containing the product or product subcomponents to be deinstalled) must reside in the Authorized Area. So, if you want to use the deinstall functionality of the **install** program, don't remove any products from the Authorized Area that are installed on any nodes.
- If a product on the Authorized Area node is installed with hard links to the Authorized Area, you can still remove its product directory from the Authorized Area. However, the space saved is minimal.

Example

To remove version 10.1 of Domain/OS (SR10.1) from the Authorized Area **//server/aa**, assuming no active override file for this product exists,

Using UNIX,

```
rm -rf //server/aa/install/ri.apollo.os.v.10.1
rm -rf //server/aa/install/templates/apollo/os.v.10.1
rm //server/aa/install/doc/apollo/os.v.10.1*
```

Using Aegis,

```
dlt -f -du //server/aa/install/ri.apollo.os.v.10.1
dlt -f -du //server/aa/install/templates/apollo/os.v.10.1
dlf -f -du //server/aa/install/doc/apollo/os.v.10.1*
```

Loading the Installation Tools into an Authorized Area

This section describes how to load (with **rbak**) the installation tools and their help files from cartridge or magnetic tape into a disk Authorized Area. You can load the tools and help files into an existing Authorized Area or create a skeletal Authorized Area containing just these objects.

You should not need to do this in the normal course of events. The installation tools and help files are released on file 1 of the first Domain/OS product tape. They get loaded into an Authorized Area when you load and install Domain/OS on an invol'ed disk from media (Chapter 1), and when you load a Domain/OS update into an Authorized Area (Chapter 5).

After the initial load of the tools, you can copy or move the tools to other Authorized Areas as needed, rather than reload them from media. So generally you need to reload the tools from media only if your current tools have become corrupted or if you inadvertently overwrite them by loading earlier versions from distribution media.

You must be logged in as **root** to reload tools into an existing Authorized Area.

1 Insert the first Domain/OS product tape into the cartridge-tape drive, or mount the first magnetic tape.

The cartridge tape is labeled **CRTG_STD_SFW_1**.

Do *not* use an optional-product tape. Some older optional products contain the installation tools on their distribution media. But these tools may be earlier versions than those released with your current version of Domain/OS.

2 Load the installation tools and help files.

To load the tools and help files, enter the command

rbak -dev *dev* **-ms -sacl -pdt -force -du -f 1 install/tools -as** *AA*/**install/tools install/help -as** *AA*/**install/help**

where

dev is **ct** for cartridge tape and **m** for magnetic tape.

AA is the pathname of an existing Authorized Area or an Authorized Area that you want to create. In the latter case, **rbak** creates both the *AA* directory and the other subdirectories (*AA*/**install/tools** and *AA*/**install/help**).

If you are creating an Authorized Area, you may want to add the clause **install/overrides -as** *AA*/**install/overrides** to the command line. This creates an empty **install/overrides** directory in the Authorized Area, which is required by the **cfgsa** tool to execute. Alternately, you can create the **overrides** directory later with the **mkdir** or **crd** command.

Example

In the following example, we load the installation tools and help files from cartridge tape into the Authorized Area **//server/aa**. We've added the **-ld** option to the **rbak** command line to list the directories restored.

```
# rbak -dev ct -ms -sacl -pdt -force -du -f 1 -ld\<RETURN>
> install/tools -as //server/aa/install/tools\<RETURN>
> install/help -as //server/aa/install/help

Label:
   Volume ID:      ST0147
   Owner ID:       apollo
   File number:    1
   File section:   1
   File ID:        force
   File written:   1990/04/27 06:59:13 EST

Starting restore:
(dir)  "//server_node/aa/install/help" restored.
(dir)  "//server_node/aa/install/tools" restored.
Restore complete.
```

Part 3

Installation Reference

This part of the manual (Part 3) describes the bits and pieces of the Domain installation model in detail. Unlike Parts 1 and 2, which organize information in terms of specific tasks and procedures, this part describes each component of the model in a more traditional, expository fashion.

You can use this part simply as a reference for more detailed information when you perform the tasks in Parts 1 and 2. (Parts 1 and 2 do *not* describe all the command line options for the installation tools.) You can also read this part as a self-contained description of the Domain installation model. We strongly encourage any one who is responsible for managing an Authorized Area or installs products frequently to read this part in its entirety.

Chapter 10 provides a detailed description of each installation tool and its command line options in a format similar to UNIX man pages.

Chapter 11 provides a detailed description of the components in an Authorized Area and how these components relate to distribution media.

Chapter 12 describes some common error and warning messages displayed by the **install** program.

10

Installation Tool Reference

A comprehensive description of each installation tool and its command line options

This chapter provides a detailed, comprehensive description of each installation tool and its command line options. The tools described are:

- cfgsa
- config
- distaa
- install
- install++
- minst
- mrgri

The approach is structural, not task-oriented. The format used is similar, but not identical, to the format used for UNIX man pages.

cfgsa

cfgsa *AA*

The **cfgsa** tool (*AA*/**install/tools/cfgsa**) is an interactive tool used to create selection and override files for a product in an Authorized Area.

The parameters are as follows:

AA — Specify the pathname of the Authorized Area that contains the product(s) for which you want to create selection and/or override files.

Example

To begin an interactive **cfgsa** session for the products in the Authorized Area **//server/aa**,

```
//server/aa/install/tools/cfgsa //server/aa
```

With **cfgsa**, you can create a selection file and a corresponding override file for a product, or create just an override file. In both cases, the product or at least the product's release index must reside in the specified Authorized Area. The process is entirely optional. For some products, most notably Domain/OS, HP supplies several predefined selection and override files (described in the product's release notes). You may find it easier and sufficient to use predefined files rather than create your own.

Creating a Selection-Override File Pair

With **cfgsa**'s **generate** command, you can create a selection file and a corresponding override file in the current directory. The selection file defines a subset of a product of your choosing. The override file defines a set of configuration options that are consistent with the reduced set of product components.

After you remove the original product (or the product's release index) from the Authorized Area, you can supply the pathname of the selection file as a command-line argument to the **distaa** tool to load the product subset from

distribution media into the Authorized Area. This enables you to save disk space on the Authorized Area node by not loading components of a product that you don't need in your environment. You then make the corresponding override file active, by copying it or moving it to the Authorized Area's **install/overrides** directory and naming it **ri.apollo.***product_name***.v.***version*. This procedure is fully described in Chapter 7.

Creating an Override File Only

With **cfgsa**'s **save** command, you can create just an override file for a product. The override file restricts the product's configuration choices presented to users by the **config** (or **install++**) tool. Also, if a user installs a product with a configuration file created prior to the restrictions, the restrictions defined by the override file take precedence over any configuration choices specified in the configuration file but not allowed by the override file.

You can use override files to prevent users from installing certain product subcomponents or to force users to install subcomponents in a certain way — as a link to another node, for example, rather than a local copy. This enables you to precisely control how products are installed and to make the most efficient use of disk space throughout the network.

The **save** command creates override files with pathnames of the form *AA*/**install/overrides/ri.apollo.***product_name***.v.***version*. Placing an override file in the **install/overrides** directory and naming it this way makes it *active*, meaning the restrictions immediately take effect.

cfgsa Commands

When you invoke **cfgsa**, an interactive configuration session is started. A CFGSA> prompt appears, at which you enter **cfgsa** commands. A description of each of these commands follows. Although there are several commands, the basic process is one where you select a single product (using **select**), define the configuration constraints for the selected product (using **constrain**), and then save these constraints (using **save** to create an active override file, or **generate** to create a selection file-override file pair). You can then repeat this process for other products in the Authorized Area.

The commands are shown here in lowercase, but you can enter them in upper, lower, or mixed case. You can also abbreviate them to the point of uniqueness. For example, you can enter **a** instead of **available**.

available Display a list of the products available in the Authorized Area specified on the command line. The name and version number of each product is listed. **cfgsa** automatically executes an **available** command when it starts up.

constrain Begin a constraint session for the currently selected product. **cfgsa** displays the configuration questions that are shown to a user when a user runs **config** or **install++** to configure the product, and the possible answers that a user can supply. The questions shown are those defined by the product's release index, as constrained by any active override file for the product (this can be a predefined override file or one you created previously with **cfgsa**).

cfgsa pauses after each configuration question and prompts you to enter one of three types of constraints: **answer**, **limit**, or **user**. Keep in mind that the meaning of each of these responses depends in part on whether you subsequently use the **save** command (create an override file only) or the **generate** command (create a selection file and corresponding override file). If you use **save**, the constraints you define apply only to the configuration questions displayed to a user. If you use **generate**, the constraints also define a product subset that you can load from media with **distaa**. Depending on the particular configuration question and your responses, the results may effect only the configuration process prior to installation and not what gets loaded into the Authorized Area.

> **answer** Answer the question for the user; **cfgsa** prompts you to enter the answer of your choice. (If the question asks if a particular product subcomponent is to be installed as a link, versus a local copy or not at all, and you preanswer the question as **link**, **cfgsa** also prompts you for the name of the link node.) The answer you supply becomes the answer to the configuration question. When a user configures the product, the user does not see the question at all. If you preanswer a question concerning the installation of a product subcomponent as **none** or **no**, and you subsequently create and use a selection file that reflects the constraints, the subcomponent is not loaded from media. Similarly, if you specify that just one of a number of related objects is to be installed, use of the resultant selection file causes only that object to be loaded from media.

limit Limit the answers a user can supply to a subset of the possible answers; **cfgsa** prompts you to enter the answers you want presented to the user at configuration time. (Note that if the text of the question refers or alludes to answers that you exclude, the question may become slightly confusing to the user.) If you limit the answer set for a question that asks which of a number of related objects are to be installed (such as the question about which **sau** directories to install for Domain/OS), and you subsequently create and use a selection file that reflects the constraints, only the specified subset is loaded from media.

user Impose no constraints. When a user configures the product, present the user with the question and the full set of possible answers. If the question concerns the installation of a subcomponent or group of related subcomponents, and you subsequently create and use a selection file that reflects the constraints, all subcomponents are loaded from media. This is the default response, which you can also select it by entering <RETURN>.

In response to each configuration question, you can also enter

help to obtain information about the constraint responses described here.

refresh to redisplay the question and answers.

abort to exit the constraint session, saving any constraints already defined, and return to the CFGSA> prompt.

When you finish responding to all of the configuration questions for the product, the CFGSA> prompt returns. To save the constraints, you must then use the **save** or **generate** command. If you select and constrain another products before executing **save** or **generate**, you lose the constraints defined for the first product.

exit Exit the **cfgsa** tool. If you constrain a product (with **constrain**) and exit before you save the constraints (with **save** or **generate**), **cfgsa** does not save the constraints.

generate *name* Create a selection file and a corresponding, non-active override file for the product that reflect the constraints currently defined (via **constrain**). The selection file is named **aa**.*name* and the override file **ov**.*name*, where *name* is a character string of your choice. The files are placed in the current working directory.

help Display a summary of the **cfgsa** commands accepted at the CFGSA> prompt.

revert Remove all constraints that you've imposed on the selected product during the current **cfgsa** session.

save Create an active override file for the product that reflects the constraints currently defined (via **constrain**). The file is named *AA*/**install/overrides/ri.apollo.***product_name***.v.***version***, where *product_name* and *version* are the release index name and version number, respectively, of the constrained product. 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_number* | *product_name version*} Select a product from the list of available products. You must select a product before you can constrain it with the **constrain** command. You can select a product by entering the number displayed to the left of the product name in the list of available products (*product_number*). Or you can enter the product's name and version number (*product_name version*) as displayed in the list of available products. If you select a product and then select another product before you constrain the first product, the first product selection is ignored. Similarly, if you select and constrain a product and then select another product before you save the constraints (using **save** or **generate**), the constraints for the first product are lost.

show List all of the configuration questions, and any constraints you've applied during the current **cfgsa** session, for the selected product. The questions displayed are those defined in the product's release index, as constrained by any active override file.

See Also

Chapter 6, Restricting Product Configurations.
Chapter 7, Defining and Loading a Custom Product Subset.
"distaa," this chapter.
"Selection Files" and "Override Files," Chapter 11.
AA/**install/help/cfgsa.hlp**, the online help file for **cfgsa**.

config

config –s *AA* –c *configuration_file*

The **config** tool (*AA*/**install/tools/config**) is an interactive tool used to create and modify product configuration files.

The parameters are as follows:

-s *AA*

Specify the pathname of the Authorized Area (*AA*) that contains the products to be configured. **config** uses the release indexes for the products in the specified Authorized Area as sources for the configuration questions it asks. If an active override file for a product exists (in the *AA*/**install/overrides** directory), the configuration questions presented for the product reflect the configuration constraints defined by the override file.

-c *configuration_file*

Specify the pathname of the configuration file (*configuration_file*) that you want **config** to create or modify. The product configuration you define with **config** is saved in the specified file.

Example

To configure one or more products in the Authorized Area **//server/aa** and save the configuration in a file named **//my_node/config_file**,

//server/aa/install/tools/config –s //server/aa –c //my_node/config_file

Most Domain software products are not installed on an all-or-none basis. Rather, most products ship with a set of configuration options, defined in the product's release index file, that enable you to specify precisely how the product is installed on a node. The configuration options enable you to specify which optional subcomponents of a product you want to install, whether to install the subcomponents as local copies or as links to other nodes (on which the subcomponents are or will be installed), and other operational aspects of the product.

config is used to define a configuration of one or more products resident in an Authorized Area and to save the configuration in a file specified on the command line. To actually install the product configuration, you subsequently supply the pathname of the configuration file as a command-line argument to the **install** (or **install++**) tool.

config Commands

When you invoke **config**, an interactive configuration session is started. A `CONFIG>` prompt appears, at which you enter **config** commands. Although there are several commands, the basic process is one where you select the products you want to include in the configuration (using the **select** command), answer a series of configuration questions for each of the selected products (using the **configure** or **reconfigure** command), and then **exit** from **config** to save the configuration you defined in the specified file. Most of the commands relate to either the selection or configuration of products.

The commands are shown here in lowercase, but you can enter them in upper, lower, or mixed case. You can also abbreviate them to the point of uniqueness. For example, you can enter **s a** instead of **show available**. Several commands take *product_name* and *version_number* as arguments. *product_name* and *version_number* are the name and version number of a particular product, as displayed by **config**'s **show available** or **show selections** command. If you omit *version_number*, **config** selects the latest version of the product available in the Authorized Area or in the list of selected products, whichever applies.

Note that when both the m68k and a88k versions of a product reside in an Authorized Area, **config** considers the a88k version *later than* the m68k version when it must select the latest version of a product. (An a88k version is indicated by a **.p** extension in the product's version number.) For example, if you use the **select all** command, and the products **ri.apollo.os.v.10.4** and **ri.apollo.os.v.10.4.p** reside in the Authorized Area, only **ri.apollo.os.v.10.4.p** (the a88k version) is selected.

abort Exit the configuration session without saving any input from the current session. If you are modifying an existing configuration file, the modifications are not written to the file. If you are creating a new configuration file, **config** creates an empty file with the specified name. To remove the empty file, you must use the UNIX **rm** or Aegis **dlf** command.

configure *product_name* [*version_number*] Configure the specified product. **config** displays all unanswered configuration questions for the product. (To unconditionally display all questions, use **reconfigure**.) The questions displayed are those defined in the product's release index, as constrained by any active override file for the product. Some products have no configuration questions. In this case, **configure** displays the message "All queries answered for the product." Before you can configure a product, you must select it, using the **select** command or one of the related select commands. See "Answering Configuration Questions" later in this section for information about configuration questions and how to answer them.

deselect *product_name* [*version_number*] Remove the specified product from the list of currently selected products and, if you are modifying an existing configuration file, remove the product from the configuration file upon exit from **config**. If you omit *version_number*, **config** deselects the latest version of the product in the list of selected products. If you deselect a product that is required for the operation of a currently selected product, you are warned when you try to install the configuration, not at this time.

exit Exit the **config** tool and save the results of the **config** session in the configuration file specified on the command line. Any unanswered configuration questions for a product are set to their default values.

help Display a summary of the commands accepted at the `CONFIG>` prompt.

install checking *product_name [version_number] check_type* Set the type of object checking to perform (to *check_type*) when the specified product is reinstalled or installed as an update on a target. The possible values for *check_type* are:

none Install all constituent objects of the product, regardless of whether any of the objects already exist on the target or have been manually changed by the user in any way; a *force* install. This overrides the default action of the **install** tool, which is to respect product customization; that is, to *not* reinstall objects that have been manually deleted, manually changed from a local copy to a link, or manually changed from a link to a copy.

exist Install all objects as called for by the configuration, except do *not* reinstall objects that were manually deleted, manually changed from a local copy to a link, or manually changed from a link to a copy on the target node. Unlike **version** checking, an object that already exists on the target node is reinstalled, even if it the same version as the object in the product being installed.

version If an object already exists on the target, install it only if it is a different version (has a different date and time stamp). Additionally, respect product customization (that is, do not reinstall objects that have been manually deleted, manually changed from a local copy to a link, or manually changed from a link to a copy). If you use the **-m** switch with the **install** tool (do not respect product customization), **-m** overrides the respect-product-customization aspect of **version** checking. **version** checking is the default and the type of object checking we recommend.

reanswer *product_name [version_number] query_name* Display the configuration question named *query_name* for the specified product and enable you to reanswer it (rather than force you to reconfigure the entire product to change the answer to the question). The product must be selected first and, if you are creating a new configuration file, the question must have been previously answered during the current configuration session. If there are other configuration questions conditionally related to the one you reanswer, **config** displays and asks you to reanswer these as well. See "Answering Configuration Questions" later in this section for information about configuration questions and how to answer them.

To identify the *query_name* for a configuration question, use the **show queries** command. Also note that when you configure a product (with **configure** or **reconfigure**), the *query_name* for each configuration question appears before the actual text of the query as `** Name: query_name **`.

In some cases, when you try to reanswer a question, **config** displays the error message "*query_name* can't be reanswered." This means that the question is no longer applicable because of the way another related configuration question was previously answered. For example, if you answer "red" to the configuration question "Do you want blue or red?" and then try to reanswer the question "Do you want sky blue or slate blue?" (a question that presupposes an answer of "blue" to the red-or-blue question), the error message appears because the shade-of-blue question is no longer applicable.

reconfigure *product_name* [*version_number*] Reconfigure the specified product. **reconfigure** presents *all* the configuration questions defined in the product's release index, as constrained by any active override file. (This differs from **configure**, which displays only unanswered questions.) Upon invocation, **reconfigure** resets all configuration questions to their default values; any answers already defined in the configuration file or specified during the current configuration session are thrown out. Before you can reconfigure a product, you must select it. See "Answering Configuration Questions" later in this section for information about configuration questions and how to answer them.

select *product_name* [*version_number*] Add the specified product to the list of currently selected products. You must select a product before you can configure it (using **configure**) or reconfigure it (using **reconfigure**). If you omit *version_number*, **config** selects the latest version of the product available in the Authorized Area. If you select a product that is incompatible with a currently selected product, you are warned when you try to install the configuration, not at this time (because **config** is target-machine independent). If you exit **config** without configuring a selected product, the product is added to the configuration with all configuration questions set to their default values. If you are modifying an existing configuration file, the products in the existing configuration are automatically selected when **config** starts up.

select all Select the latest version of all products in the Authorized Area. These products are added to the list of products already selected (if any). If an earlier version of one of the products is already selected, this version is *not* removed from the list of selected products.

set linkprompt *text* Specify a pathname (*text*) to be used as the default link destination for any configuration questions that prompt you for the destination of a link. When you configure products, you are still prompted for the link destination and can override the linkprompt *text* at that time; setting the link destination just keeps you from having to enter it repeatedly. The default link destination does not change until you issue another **set linkprompt** command. It lasts for the duration of the configuration session, and is not saved in the configuration file. To remove a default link prompt that you set previously, enter **set linkprompt** without any *text*. If you don't explicitly set a linkprompt, the last link destination you enter (if any) becomes the default link destination for all products during the current configuration session.

show available Display a list of products available in the Authorized Area specified on the command line. The name and version number of each product is listed. **config** automatically executes a **show available** command when it starts up.

show linkprompt Display the pathname (set by **set linkprompt**) that is used as the default link destination for any configuration questions that prompt you for the destination of a link.

show queries *product_name* [*version_number*] Display all configuration questions and their current answers for the specified product. The product must be selected first.

show selections Display the products currently selected for installation.

update *product_name* Replace all versions of the specified product, in the list of selected products, with the latest version of the product available in the Authorized Area. At least one version of the the specified product must already be selected. Answers to configuration questions for the earlier version of the product are retained, if the latest version has the same questions.

update all Replace all products currently selected with the latest versions of these products available in the Authorized Area. Answers to configuration questions for any updated products are retained, if the latest versions of the products have the same questions.

validate *product_name* [*version_number*] Determine whether installation of the specified product will succeed using the current configuration and current Authorized Area. **validate** checks that all the product subcomponents selected in the configuration reside in the specified Authorized Area. If **validate** does not return any messages, the product configuration is valid.

Answering Configuration Questions

When you enter the **configure**, **reconfigure**, or **reanswer** command at the CONFIG> prompt, **config** presents configuration questions for the product. The questions are presented in the following format:

```
** Name: query_name **
query_text
: [ possible_answers ]
==>
```

where:

query_name is the internal name by which **config** identifies the question, and the name you use to specify a question when you use the **reanswer** command.

query_text is the actual text of the query.

possible_answers is a list of the possible answers you can supply to the question. The default answer, which you can select by pressing <RETURN>, is marked with a (D).

==> is the prompt at which you can enter one of several responses to the question. You can enter one or more of the *possible_answers*. The answer can be abbreviated to the shortest unique string. You can also enter **stop** or **help** at the ==> prompt:

stop stop the query answering process and return to the CONFIG> prompt.You can resume the answering process where you left off with the **configure** command. If you exit the configuration session without answering all of the questions, **config** sets the unanswered questions to their default values. Entering an interrupt signal (usually ^Q in Aegis and ^C in UNIX) or an end-of-file signal (usually ^Z in Aegis and ^D in UNIX) at the ==> prompt is equivalent to entering **stop**.

help display a list of the possible responses you can enter at the ==> prompt.

Note that when one of the possible answers to a question is **stop** or **help**, **config** interprets the response as the **stop** or **help** command, and not the answer to the question. To have a response of **stop** or **help** interpreted as an answer rather than a command, prefix the answer with a dash (-).

Some configuration queries ask if you want to install a product subcomponent as a link to another node. If you request that the subcomponent be installed as a link, this prompt appears:

```
Link Text>>>>
```

At this prompt you enter the target of the link (typically the name of another node, *//node_name*, on which the product is or will be installed). The **stop** and **help** commands are not recognized at this prompt. Also, the interrupt signal has no effect: entering it results in a warning message and a reprompt for the link text.

config Versus install + +

Instead of invoking **config** directly, you can use the **install + +** tool. Upon startup, **install + +** invokes the **config** tool. Once you complete the configuration process, **install + +** immediately invokes the **install** program to install the product configuration. **install + +** enables you to define a product configuration that is used only for that particular installation and not saved in a file. A primary advantage to invoking **config** directly is that you can reuse the configuration file for subsequent installs. Also, **config** allows you to separate in time the configuration and installation processes, which many administrators find desirable.

See Also

"Configuring Products," Chapter 5.
Chapter 6, Restricting Product Configurations.
"install" and "install + +," this chapter.
"Override Files" and "Configuration Files," Chapter 11.
AA/**install/help/config.hlp**, the online help file for **config**.

distaa

distaa {**-a** | *selection_file*} [**-f**] [**-v**] [**-e** *max_errors*] [**-m** *media_type*] *AA*

The **distaa** tool (*AA*/**install/tools/distaa**) loads products non-interactively from distribution media (cartridge tape, magnetic tape, floppy disk) into an Authorized Area.

The parameters are as follows:

-a
selection_file

You must specify either the **-a** (all) option or the *selection_file* argument.

-a The **-a** option causes **distaa** to load all products in their entirety from the distribution media into the Authorized Area.

selection_file The *selection_file* argument is the pathname of a selection file. The selection file causes **distaa** to load only a single product or a subset of a product's components, rather than the entire product. Also, you can edit a selection file to distribute the components of a product on more than one node. *selection_file* can either be the pathname of a selection file you created previously with the **cfgsa** tool, or the pathname of an HP-supplied selection file. HP-supplied selection files have pathnames of the form

AA/**install/templates/apollo/***product_name*.**v.***version*/**aa.***name*

You can specify only one selection file pathname on the command line. If the distribution media contains more than one product, and you want to load more than one (but not all) of the products, you must invoke **distaa** separately with the appropriate selection file pathname for each product.

If you use a selection file to load a product subset, you should *activate* the corresponding override file (the file that restricts the configuration options to a set that is consistent with the reduced set of product components) after you load the subset into the Authorized Area. See "Override Files" in Chapter 11 for details.

See "Selection Files" in Chapter 11 for a comprehensive discussion of selection files.

-f Force **distaa** to load every object into the Authorized Area, even if the object already exists in the Authorized Area. Without **-f, distaa** loads only those objects which are not already present in the Authorized Area.

-v List each object as it is loaded. Without **-v, distaa** displays an abbreviated listing.

-e *max_errors* Set the maximum number of load errors that can occur before **distaa** aborts to *max_errors*. The *max_errors* argument must be a decimal integer. Without **-e, distaa** does not abort when load errors occur.

-m *media_type* Specify the type of distribution media that **distaa** is to load from. The argument *media_type* can be one of the following:

c	Cartridge tape
f	Floppy disk
m	Open-reel magnetic tape

Without **-m, distaa** assumes the media type is cartridge tape.

AA Specifies the pathname of the Authorized Area in which to load products. The directory *AA* must exist. However, it does not have to be an Authorized Area. If it isn't, **distaa** creates the requisite **install** subdirectory and places the product directory in *AA*/**install**, as it does with an existing Authorized Area.

Examples

To load all products on a cartridge tape into the Authorized Area **//aa**,

```
//aa/install/tools/distaa -a //aa
```

To load a small Aegis subset of SR10.4 (version 10.4 of the Domain/OS product) from cartridge tape into the Authorized Area **//aa**,

```
//aa/install/tools/distaa //aa/install/templates/apollo/os.v.10.4/aa.aegis_small //aa
```

To load patches **pd91_m0223** and **pd91_m0224** from a patch tape into the Authorized Area **//aa**,

```
//aa/install/tools/distaa
  //aa/install/templates/apollo/pd91_m0223.v.1.0/aa.pd91_m0223 //aa

//aa/install/tools/distaa
  //aa/install/templates/apollo/pd91_m0224.v.1.0/aa.pd91_m0224 //aa
```

distaa loads products and their associated release indexes from distribution media into an Authorized Area (*AA*). It places each product and product release index in a product directory named *AA*/**install**/*product_name*.**v**.*version*. If you load a product or set of products that reside on more than one piece of media, **distaa** prompts you during the load process to remove each piece and replace it with the next one.

What distaa Does Not Do

distaa loads only products and their release indexes from media. It does *not* load the administrative objects located in the first file on the media. These objects includes the product release documentation, HP-supplied selection and override files, the media TOC file, and the installation tools and their help files (released on the distribution media for Domain/OS and some older optional products). You must load these objects using the **rbak** command before you execute **distaa**, as described in the the "Loading Products from Media into an Authorized Area" section in Chapter 5.

An Alternative to distaa

Instead of using **distaa**, you can use the **minst** tool. **minst** provides an interactive interface to the **distaa** and **install**++ tools. It enables you to quit after you execute the load (**distaa**) phase and before you install any products. Unlike **distaa**, **minst** automatically loads the administrative objects from the media. Also, if you load a subset of a product, it automatically activates the appropriate override file.

See Also

"Loading Products from Media into an Authorized Area," Chapter 5.
Chapter 7, Defining and Loading a Custom Product Subset.
"cfgsa" and "minst," this chapter.
"Selection Files," "Override Files," and "Relation of Authorized Areas to Distribution Media" Chapter 11.
AA/**install/help/distaa.hlp**, the online help file for **distaa**.

install

install [**-dehilmoprtvx**] **-s** *AA* {**-c** *configuration_file* | **-a** | **-u**}
{*target* |**-n** *target_list*}

install -A [**-hoprtvx**] **-s** *AA* {**-c** *configuration_file* | **-u**}
{*target* |**-n** *target_list*}

install -D *product_name version* [**-oprtvx**] **-s** *AA*
{*target* |**-n** *target_list*}

The **install** tool (*AA*/**install/tools/install**) installs a product configuration defined in one or more configuration files on one or more target nodes. **install** can also reset the Access Control Lists (ACLs) of products, deinstall an entire product, and deinstall product subcomponents. The first syntax line above is for installing products and deinstalling product subcomponents, the second for resetting ACLs, and the third for deinstalling an entire product.

While **install** has many command line parameters and options, only three are required for installing products: the **-s** switch to specify an Authorized Area containing the products to be installed; either the **-a**, **-c**, or **-u** option to specify a configuration of those products; and specification of at least one target node, either explicitly (*target*) or with the **-n** *target_list* option. We also recommend using the **-x** (continue on error) and **-v** (verbose) options.

The parameters are as follows:

-s *AA*

Specify the pathname of the Authorized Area (*AA*) containing the products to be installed. The Authorized Area must contain the products defined in the configuration specified on the command line.

-c *configuration_file*
-a
-u

You must use either the **-c** *configuration_file*, **-a**, or **-u** option.

-c *configuration_file* installs the product configuration defined in the specified configuration file. *configuration_file* can be either the pathname of a configuration file that you created previously with the **config** tool or the pathname of a product's default configuration file. Default configuration files have the pathname *AA*/**install/templates/apollo/***product_name*.v.*version*/**cf.***product_name*. You can use the **-c** option multiple times on the same command line to specify multiple configuration files (**-c** *file_name* **-c** *file_name* ...), which together define a configuration. The constraints defined by any active override file for a product in the configuration take precedence over the configuration choices defined in the specified configuration file(s).

-a installs the latest version of all products available in the Authorized Area. The products are installed according to the default configuration settings defined in the products' release indexes, as constrained by any active override files. (The default values are the first answers given in the list of possible answers for each configuration question displayed by the **config** tool.) If the Authorized Area contains an m68k and a88k (Series 10000) version of a product, the m68k version is installed on m68k target nodes and the a88k version on a88k target nodes.

-u updates every product already installed on the specified target(s) with the latest versions available in the Authorized Area. The configurations installed are the same as those resident on the target(s). However, if a configuration question has been added for a later version of a product, the default answer for this question is used. To determine which products to install and the configuration of those products, **install** reads the latest baseline file on each target You can also use **-u** with the **-A** option (reset ACLs) to reset the ACLs of *all* products installed on the target node(s) to their original settings.

target

Specify the pathname of the directory in which to install the products. Although *target* can be the pathname of any directory, you typically specify the name of a node's entry directory (*//node_name*) or the pathname of the mount point of a mounted volume. You can specify more that one *target* on the same command line, separating each with a space. You can also specify a list of targets using the **-n** *target_list* option instead of, or in addition to, specifying individual targets. **install** requires at least one target, specified by either of these means.

-n *target_list*

Specify the pathname of a user-created file (*target_list*) that contains a list of target directories (one per line). In the *target_list* file, you can specify the pathname of any directory as an installation target. But you typically specify the name of a node's entry directory (*//node_name*) or the pathname of the mount point of a mounted volume. You can use **-n** *target_list* instead of, or in addition to, specifying *target*s explicitly on the command line.

-A

Reset the Access Control Lists (ACLs) of the designated products to their original settings, as defined by each product's release index; or, if a product had a configuration question that specifies how the product's ACLs are to be set, set the ACLs according to setting in the specified configuration file. **-A** modifies the ACLs of *installed* products only; it does not effect the ACLs of products in an Authorized Area. Also, when you use **-A**, no objects are installed or updated; only ACLs are effected.

-A has two primary uses: to reset the ACLs of an installed product whose ACLs have been manually modified or corrupted in some way; and to change the ACL settings of an installed, pre-10.4 version of Domain/OS from "open" to "closed" or vice-versa. Pre-10.4 versions of Domain/OS have a configuration question that asks if you want to install Domain/OS objects with "open" or "closed" ACLs. Open ACLs means the world has **p** rights — any user can change all other access rights for the object.

You can use either the **-c** *configuration_file* or **-u** option to specify which product's ACLs are to be reset. If you use **-u**, the ACLs of *all* products installed on the target node(s) are reset to their initial settings, as defined by the product release indexes. If you use **-c**, the ACLs of only those products specified in the configuration file are reset. The ACLs are changed to their initial settings as defined by each product's release index or, if the product is a pre-10.4 version of Domain/OS, to "open" or "closed" as specified in the configuration file. (The ACL settings of the product in the Authorized Area are immaterial.) The configuration file must be the configuration file that was originally used to install the product(s) on the designated target(s). If you do not have the original configuration file, you must reinstall the products to reset their ACLs.

-A changes only those ACLs that are explicitly set in the product's release index. **-A** does not change the ACLs of any objects whose ACLs are set by inheritance or of any objects that did not exist in the original set of product files.

-d

Deinstall (remove) product subcomponents from the target(s). You can use **-d** with the sole purpose of removing a specified subcomponent of an installed product. To do this, you reconfigure the appropriate product(s) and answer **none** or its equivalent to the configuration question that asks if you want to install the particular subcomponent. You then run **install** with **-d** and supply the name of the new configuration file on the command line. **install** removes the files and directories that correspond to the nulled answers in the configuration file. Other product subcomponents are installed as usual as called for by the configuration file. You can also use **-d** routinely whenever you reinstall or update a product to ensure installed subcomponents that are not requested by the new configuration are removed.

To use **-d** successfully, at least the release index of *all* products currently installed on the target node must reside in the specified Authorized Area. Also note that **install** does *not* deinstall an object (a file, directory, or link) in a product subcomponent in two cases: first; if the object is part of another product installed on the target; second; if the object (judging by its name), is not an object that was originally released with the product. Such objects can include user-created files and files renamed by the **install** tool when the product was updated previously. The release index of some products instructs the **install** tool to rename, and thereby preserve, certain user-modifiable files, rather than simply overwrite them, by adding a date extension to their original names. Renamed objects are not deinstalled since their names differ from those in the product's released file set. If an object is is not deinstalled, the directory containing that object is also not deinstalled. **install** displays a warning message whenever it cannot deinstall an object.

-D
product_name
version

Deinstall (remove) the specified product (*product_name version*) from the target(s). *product_name* and *version* are the name and version number of the product as used in the name of the product's release index. You can only specify one product on the command line. At least the release index of *all* products currently installed on the target node must reside in the specified Authorized Area.

install does *not* deinstall an object (a file, directory, or link) in the specified product in two cases: first; if the object is part of another product installed on the target; second; if the object (judging by its name), is not an object that was originally released with the product. Such objects can include user-created files and files renamed by the **install** tool when the product was updated previously. The release index of some products instructs the **install** tool to rename, and thereby preserve, certain user-modifiable files, rather

than simply overwrite them, by adding a date extension to their original names. Renamed objects are not deinstalled since their names differ from those in the product's released file set. If an object is is not deinstalled, the directory containing that object is also not deinstalled. **install** displays a warning message whenever it cannot deinstall an object.

install deinstalls a product even when another product installed on the target depends on the product being deinstalled.

Do not use **-D** to deinstall Domain/OS. To remove Domain/OS, you must initialize the disk using the **invol** utility. Also, deinstallation of patches is not recommended, since this removes possibly critical objects from a product, without replacing them.

-e Check whether all objects required for installing the product configuration 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, since the objects required on each target may differ when you are updating products.

-h Ignore hardware compatibility checking: force **install** to install all products in the product configuration on all specified targets, regardless of the ISP type of the products and the target nodes. Without **-h**, **install** does *not* install a product with an ISP type of a88k (a product released for Series 10000 nodes) on a node with an ISP type of m68k (a non-Series 10000 node), or vice versa, because the product may not run correctly or at all and, in some cases, may prevent the target from booting. Instead, **install** prints an error message. A situation where you might want to use **-h** would be if you are using an a88k node as a mothernode from which to boot m68k nodes diskless. You would want to install products, such as compilers, of both ISP types on the a88k mothernode, so the m68k nodes could run the m68k products when booted diskless.

-i Ignore product configuration size checking. Without **-i**, **install** checks that the product configuration is smaller than the amount of free disk space on the target(s). If it isn't, **install** displays a warning message, halts, and asks if you want to proceed. **-i** is most useful when you are installing product updates unattended (overnight perhaps) and don't want to have the installation halt unnecessarily. The space calculations are not totally accurate, so sometimes you are warned that there is insufficient space when this isn't the case, and sometimes you are *not* warned when there is not sufficient space.

-l Install the product configuration by creating hard links from the target to the objects in the product directories of the Authorized Area, rather than by copying the objects from the Authorized Area. The target directory must be on the same node as the Authorized Area. Using **-l** saves disk space when you install products on the Authorized Area node by not duplicating objects.

A hard link is simply an additional reference to the same physical object. Therefore, if you modify (write to) an installed object that is hard linked to an Authorized Area — a user-modifiable Domain/OS file, perhaps — those modifications apply to the object "in" the Authorized Area as well. When the product is subsequently installed from this Authorized Area on other nodes, the modified object — not the original version — gets installed. So use caution when modifying hard-linked objects. If you delete an installed object that is hard linked to the Authorized Area, the "corresponding" object in the Authorized Area is not deleted (and vice versa) — the deletion just removes the name of the installed object and not the object itself.

-m Do not respect product customization. If you are reinstalling or updating a product (rather than installing a product that has not been installed previously on the target node), replace any objects in the product that have been manually deleted, manually changed from a local copy to a link, or manually changed from a link to a local copy, as called for by the product configuration.

-m is the easiest way to get a clean configuration on a node whose software you suspect may have been inadvertently or erroneously modified or not quite properly installed. Also, use **-m** whenever you update a beta version of a product (to a later beta version or the fcs version) to ensure that any new objects in the later version are installed.

Even when **-m** is used, **install** does *not* overwrite a link with a directory or a directory with a link under certain conditions. These conditions are described in "Changing Copies to Links and Links to Copies" later in this section.

-o Install to each target as many times as the target is specified on the command line or in a *target_list* file. Without **-o**, **install** installs only once to each target, even if the target is repeated on the command line or in a *target_list*. **-o** is primarily useful during long, unattended installations. By using **-o** and naming each target twice on the command line or in a *target_list*, you can improve the chances of a successful installation on each target. You can also use **-o** when you deinstall a product (**-D**), deinstall product subcomponents (**-d**), and reset ACLs (**-A**) to repeat the deinstallation or ACL-resetting attempt as many times as the target is listed.

-p Purge all previously created baseline files on the target(s). When you update a product, the earlier version of the product is not listed in the newly created baseline file. Therefore, do not use **-p** if you want to retain a comprehensive history of all products formerly installed on the node.

-r Execute the installation processes remotely on each target node, rather than locally on the node on which **install** is invoked. Without **-r**, **install** runs only on the node it is invoked from and installs (or deinstalls) the configuration on each target in succession, one after the other. When you install (or deinstall) products on more than one target, **-r** distributes the workload and is faster. The Server Process Manager (**/sys/spm/spm**) must be running on a target for **install** to execute remotely on that target. If **spm** is not running on a target, **install** executes locally just for that target; it still executes remotely on targets with **spm** running. The **-r** option is incompatible with the **-a** and **-u** options; you must specify a configuration file using the **-c** option (unless you are using **-D** to deinstall an entire product).

For each remote target, **install** creates a log file in the source Authorized Area (*AA*). Each log file has the pathname *AA/node_name.x*, where *node_name* is the name of the remote target node and *x* is an integer value. The log file records any any error or warning messages from the target and other pertinent installation information.

-t Test whether the configuration can be successfully installed from the Authorized Area to the target(s), without actually installing the configuration. If you are deinstalling a product (**-D**) or product subcomponents (**-d**), **-t** lists all objects that will be deleted and those that cannot.

-v Report more information while **install** runs. Without **-v**, **install** displays an abbreviated listing. We recommend using the **-v** option to assist you in troubleshooting any installation errors and understanding any warnings.

-x Continue if an error is encountered. Without **-x**, **install** aborts the installation when it encounters an error. You can often correct an installation error without having to rerun **install**.

Examples

In all of the following examples, the name of the Authorized Area is **//server/aa**.

To install the product configuration defined in the configuration file **//my_node/cf.compilers** to the node **//my_node**,

//server/aa/install/tools/install –vx –s //server/aa –c //my_node/cf.compilers //my_node

To remotely install the product configuration defined in the configuration file **//server/cf.os_update** to the nodes listed in the file **//server/target_list**,

//server/aa/install/tools/install –irvx –s //server/aa –c //server/cf.os_update –n //server/target_list

To deinstall version 4.0 of the product **lisp** from the nodes **//bsa** and **//bsa2**,

//server/aa/install/tools/install –vx –s //server/aa –D lisp 4.0 //bsa //bsa2

The following sections describe additional aspects of the **install** tool: objects **install** uses and creates to monitor, record, and control the installation process; disk space requirements; how **install** handles the changing of local copies to links and links to copies; and things to do after **install** completes execution.

The Installation Log File

Whenever you invoke the **install** tool, the invocation is logged in a log file in the Authorized Area. The pathname of the log file is *AA*/**install/log_file**, where *AA* is the name of source Authorized Area specified on the command line with **–s**. For each invocation, the following information is logged:

- The name of the user who invoked **install**, and the date and time.
- The **install** command line used.
- The name of the node on which **install** was invoked and the version of Domain/OS running on that node.

If the **log_file** is open when **install** attempts to write to it (because another installation is being logged), **install** attempts to write to the log file up to five times. If all five attempts are unsuccessful, the installation is not logged and the installation continues.

The **excludes.list** File

You can specify files that you do not want installed by creating a file named **excludes.list** in the Authorized Area's **install/overrides** directory (*AA*/**install/overrides/excludes.list**). In the file, list the absolute pathnames (pathnames beginning with the node entry directory, /) of the files you don't want installed. Use the pathnames that the files would have if installed, not the pathnames of the files as they reside in the product directory in the Authorized Area. Whenever you install products from this Authorized Area, **install** checks the file and does not install the files listed on any of the specified targets.

The Target's /install Directory

Every node contains an **/install** directory that is created by the **install** tool the first time you install Domain/OS on the node. The directory contains a number of objects that are created and/or used by the **install** tool whenever products are installed on the node. If a node contains an Authorized Area at the node entry directory, the node's **/install** directory doubles as the **install** directory of the Authorized Area.

The objects in, or potentially in, each node's **/install** directory are:

/install/baseline files Whenever you install products on a node, **install** creates a file in the target node's **/install/baseline** directory, called **baseline.***n*, where *n* is a number. The **baseline** file lists all products installed on the node and the settings of the configuration options for each product. **baselines** files are cumulative: each time you run **install**, **install** creates a new **baseline** file by appending the products installed during that execution to the list of products in the previous **baseline** file, and increments the *n* file-name extension by 1. (However, when you install an update of a product, the earlier version is removed from the newly created **baseline** file.) Whenever you deinstall a product, **install** creates a new **baseline** file with the deinstalled product removed from the list. **install** also creates a new baseline file when you reset a product's ACLs using the **-A** option.

The **install** tool uses a target's latest **baseline** file in several ways. For example, it reads the latest **baseline** to avoid installing products that are already installed and files that are already current; to determine the current software configuration on a node when the **-u** option (update all installed products) is used; to avoid deinstalling objects that are used by another product installed on the node; and to determine when an object has been manually removed from a product and when a local copy has been manually changed to a link

(or vice versa). Therefore, you should never remove or modify a node's latest **baseline** file. If the **/install/baseline** directory is empty or does not exist, **install** assumes no products are installed on the node.

/install/not_installed file If for some reason **install** cannot install an object on a target node, it logs the pathname of the object in a file named **not_installed** in the target's **/install** directory (**/install/not_installed**). When you subsequently install products on this node, **install** attempts to install any objects listed in the **not_installed** file that are part of the products being installed. Once the objects are successfully installed, **install** removes their names from the **not_installed** file.

/install/rai_acl_temp files When **install** replaces an existing file on a target node, but cannot set the ACLs (permissions) on the new file to those of the existing file (due to a system call failure), it renames the existing file as a means of preserving a record of the original ACLs and installs the new file. **install** renames the original file **/install/rai_acl_temp***number*, where *number* is an arbitrary value. **install** flags the event with an error message that identifies the original pathname of the file. After the installation, you can look at the ACLs of the **rai_acl_temp** file to determine the original ACLs of the file, and then manually change the installed file's ACLs to the original settings with the **chmod** or **edacl** command. Alternately, you can rerun **install** with the **-A** option.

/install/preserve.list file You can specify files on a target node that you do not want overwritten when products are installed. To do this, you create a file named **preserve.list** in the target's **/install** directory (**/install/preserve.list**). In the file, you list the absolute pathnames (pathnames beginning with the node entry directory, /) of the files to be preserved. Whenever you install products on the node, **install** checks the contents of the file and does not overwrite any of the files named in it. **install** issues warnings to that effect and continues with the installation.

/install/doc/apollo **install** unconditionally installs the online release documentation for each product it installs. It places the documentation in the directory **/install/doc/apollo** on each target node.

Disk Space Requirements of install Execution

If you are installing Domain/OS or a large optional product, and the target node is the node on which you invoke the **install** tool, make sure there is at least approximately 12 MB of free space in addition to the actual size of the product(s) you are installing. This additional space is required by the installation processes as they execute.

Changing Copies to Links and Links to Copies

When you reinstall a product or install a product update, the **install** program changes local copies of files and directories to links (to soft or symbolic links), and changes files and directories that are links to local copies, when called for by the product configuration. Pre–SR10.4 versions of **install** did this for files but not for directories. For example, suppose you define a configuration for SR10.4 that requests the online man pages to be installed as a link. If you then install this configuration on a node that already has a local copy of the man pages installed, **install** deletes the resident man pages and installs the new ones as a link.

install also overwrites an existing link with another link, if the configuration being installed specifies a different link destination for the particular product subcomponent.

However, **install** does *not* change a local copy of an object to a link (or vice versa) under the following four conditions. The first three apply to directories only; the last to directories and files.

Product Intersections **install** does *not* change a local copy of a directory to a link (or vice versa) when the resident directory, or any file or subdirectory in that directory, is part of another product installed on the target node. (**install** reads the target node's latest baseline file to determine which products are installed.) When this happens, **install** prints an appropriate warning message. Note that a Product Support Kit (PSK) is treated by the installation tools as a product different from Domain/OS (because PSKs are not named **os**). Therefore, when you install Domain/OS on a node with a PSK installed, any directories in common between the PSK and the new version of Domain/OS are not changed to links (or vice versa) when called for the product configuration.

Non-Product Files **install** does *not* change a local copy of a directory to a link (or vice versa) when the resident directory tree contains any files that were not originally released with the product. These files can be user-created files or files renamed by the **install** tool during a previous product update. (The release index of some products instructs the **install** tool to rename and thereby preserve certain user-modifiable files, rather than simply overwrite them, by adding a date extension to their original names. This enables users to reuse these files, if desired, after they reinstall or update a product.) When this happens, **install** displays an appropriate warning message. To change the directory to a link, delete the offending files at your discretion (or temporarily move them to some non-product directory) and then reinstall.

Missing Release Index **install** does *not* change a local copy of a directory to a link (or vice versa) when the release index of the version of the product being updated does not reside in the Authorized Area specified on the **install** command line. In this case, *no* directories are changed to links (or vice versa) when called for by the product configuration.

Product Customization If a user has manually changed a local copy of a directory or a file on the target node to a link, or manually changed a link to a local copy, **install** does not change the link to a copy or the copy to a link, unless the **install -m** option (override product customization) is used.

After install Completes Execution

As **install** executes, it displays various informational messages. When the **install** program completes execution, check these messages for errors and warnings. Errors messages are prefixed with the label `ERROR:`; warnings messages with `WARNING:`. If the transcript contains errors, correct any problems that exist. Then, if necessary, rerun **install**. Errors often result from transient network problems during the installation. You can usually correct such problems by running the installation again.

Also, once you've successfully installed the product, reboot the node if the installation transcript displays a message instructing you to do so. Rebooting is required when you install Domain/OS or any software that changes the system libraries.

See Also

"Installing Products from an Authorized Area," Chapter 5.
"Deinstalling Products," Chapter 5.
Chapter 12, Installation Errors and Warnings.
AA/**install/help/install.hlp**, the online help file for **install**.

install+ +

install+ + [**-adehijklmoprtuvx**] **-s** *AA* [**-c** *configuration_file*]
[**-C** *config_file_list*] {*target* |**-n** *target_list*}

install+ + **-A** [**-hoprtvx**] **-s** *AA*
{**-c** *configuration_file* | **-C** *config_file_list* | **-u**}
{*target* |**-n** *target_list*}

install+ + **-D** *product_name version* [**-oprtvx**] **-s** *AA*
{*target* |**-n** *target_list*}

The **install+ +** tool (*AA*/**install/tools/install+ +**) installs a configuration of one or more products on one or more target nodes. **install+ +** optionally invokes the interactive **config** program, enabling you to interactively define or modify a product configuration (the configuration phase). **install+ +** then invokes the **install** program to install the product configuration on one or more nodes non-interactively (the installation phase). You can also use **install+ +** to reset the Access Control Lists (ACLs) of products, deinstall an entire product, and deinstall product subcomponents.

The first syntax line above is for configuring and installing products and deinstalling product subcomponents, the second for resetting ACLs, and the third for deinstalling an entire product.

The parameters are as follows:

-s *AA*

Specify the pathname of the Authorized Area (*AA*) containing the products to be configured and installed. If you specify one or more configuration files on the command line (using the **-c** or **-C** option), the Authorized Area must contain the products defined in the specified configuration(s).

target

Specify the pathname of a directory in which to install the products. Although *target* can be the pathname of any directory, you typically specify the name of a node's entry directory (*//node_name*) or the pathname of the mount point of

a mounted volume. You can specify more that one *target* on the same command line, separating each with a space. You can specify a list of targets using the **-n** *target_list* option instead of, or in addition to, specifying individual targets. **install + +** requires at least one target pathname, specified by either of these means.

-n *target_list*

Specify the pathname of a user-created file (*target_list*) that contains a list of target directories (one per line). In the *target_list* file, you can specify the pathname of any directory as an installation target. But you typically specify the name of a node's entry directory (*//node_name*) or the pathname of the mount point of a mounted volume. You can use **-n** *target_list* instead of, or in addition to, specifying *target*s explicitly on the command line.

-c *configuration_file*

Create or optionally modify the specified configuration file (*configuration_file*) and then install this product configuration. *configuration_file* can be the pathname of a non-existing configuration file (a configuration file you want to create), a configuration file that you created previously with the **config** or **install + +** tool, or a product's default configuration file. Default configuration files have the pathname *AA*/**install/templates/apollo**/*product_name*.**v**.*version*/**cf**.*product_name*.

If you specify a non-existing configuration file, **install + +** invokes the **config** tool and saves the product configuration you define in the specified file. If you specify an existing configuration file, the behavior of **install + +** depends on whether you use the **-j** or **-k** options. With neither of these options, **install + +** invokes the **config** tool and enables you to modify the existing configuration before installing it. If you immediately exit the **config** phase without modifying the configuration, the existing configuration is installed and any unanswered configuration questions are set to their default values (as defined in the product's release index). With **-j**, **install + +** skips the **config** phase and installs the existing configuration; any unanswered configuration questions are set to their default values. **-k** works the same as **-j**, except it forces you to answer any unanswered configuration questions (and only those questions) before installing the configuration.

You can use the **-c** option multiple times on the same command line to specify multiple configuration files (**-c** *file_name* **-c** *file_name* ...), which together define a configuration. However, any changes you make to the aggregate configuration during the **config** phase are *not* saved in any of the existing configuration files.

-c is incompatible with the **-a** and **-u** options.

-C *config_file_list*

Optionally modify and then install the product configuration defined by the configuration files listed in the file at *config_file_list*. *config_file_list* is a file you can create that contains the pathnames of existing configuration files (one pathname per line). With **-C**, the behavior of **install + +** depends on whether you use the **-j** or **-k** options.

With neither **-j** or **-k**, **install + +** invokes the **config** tool and enables you to modify the product configuration defined in aggregate by the set of files. Changes you make to the configuration are *not* saved in any of the existing configuration files. If you immediately exit the **config** phase without modifying the configuration, the aggregate configuration is installed and any unanswered configuration questions are set to their default values. With **-j**, **install + +** skips the **config** phase and installs the existing, aggregate configuration; any unanswered configuration questions are set to their default values. **-k** works the same as **-j**, except it forces you to answer any unanswered configuration questions (and only those questions) before installing the configuration.

You can use **-C** multiple times on the same command line to specify multiple configuration file lists, which together define a configuration. **-C** is incompatible with the **-a** and **-u** options.

-a

Install the latest version of all products available in the Authorized Area. The products are installed according to the default configurations settings defined in the products' release indexes, as constrained by any active override files. (The default values are the first answers given in the list of possible answers for each configuration question displayed by the **config** tool.) **install + +** skips the configuration phase. If the Authorized Area contains an m68k and a88k (Series 10000) version of a product, the m68k version is installed on m68k target nodes and the a88k version on a88k target nodes. **-a** is incompatible with the **-c**, **-C**, **-d**, **-k**, and **-u** options.

-A

Reset the Access Control Lists (ACLs) of the designated products to their original settings, as defined by each product's release index; or, if a product had a configuration question that specifies how the product's ACLs are to be set, set the ACLs according to setting in the specified configuration file. **-A** modifies the ACLs of *installed* products only; it does not effect the ACLs of products in an Authorized Area. Also, when you use **-A**, no objects are installed or updated; only ACLs are effected.

-A has two primary uses: to reset the ACLs of an installed product whose ACLs have been manually modified or corrupted in some way; and to change the ACL settings of an installed, pre-10.4 version of Domain/OS from "open" to "closed" or vice-versa. Pre-10.4 versions of Domain/OS have a configuration question that asks if you want to install Domain/OS objects with "open" or "closed" ACLs. Open ACLs means the world has **p** rights — any user can change all other access rights for the object.

You can use either the **-u**, **-c**, or **-C** option to specify which product's ACLs are to be reset. If you use **-u**, the ACLs of *all* products installed on the target node(s) are reset to their initial settings, as defined by the product release indexes. (**install + +** does not enter a configuration phase.) If you use **-c** or **-C**, the ACLs of only those products specified in the configuration file(s) are reset. The ACLs are changed to their initial settings as defined by each product's release index or, if the product is a pre-10.4 version of Domain/OS, to "open" or "closed" as specified in the configuration file. (The ACL settings of the product in the Authorized Area are immaterial.) The configuration file must be the configuration file that was originally used to install the product(s) on the designated target(s). If you do not have the original configuration file, you must reinstall the products to reset their ACLs.

When you use **-c** or **-C**, if none of the products have a configuration question related to ACLs, immediately exit the **config** phase without modifying the product configuration. If one of the products does have an ACL question (SR10.3, for example), optionally change the answer to the ACL question (but no other), and then exit **config**.

-A changes only those ACLs that are explicitly set in the product's release index. -A does not change the ACLs of any objects whose ACLs are set by inheritance or of any objects that did not exist in the original set of product files.

-d

Deinstall (remove) product subcomponents from the target(s). You can use **-d** with the sole purpose of removing a specified subcomponent of an installed product. To do this, you reconfigure the appropriate product(s) and answer **none** or its equivalent to the configuration question that asks if you want to install the particular subcomponent. During the installation phase, **install + +** then removes the files and directories that correspond to the nulled answers in the configuration file. Other product subcomponents are installed as usual as called for by the configuration file. You can also use **-d** routinely whenever you reinstall or update a product to ensure installed subcomponents that are not requested by the new configuration are removed.

To use **-d** successfully, at least the release index of *all* products currently installed on the target node must reside in the specified Authorized Area. Also note that **install + +** does *not* deinstall an object (a file, directory, or link) in a product subcomponent in two cases: first; if the object is part of another product installed on the target; second; if the object (judging by its name), is not an object that was originally released with the product. Such objects can include user-created files and files renamed by **install** when the product was updated previously. The release index of some products instructs **install** to rename, and thereby preserve, certain user-modifiable files, rather than simply overwrite them, by adding a date extension to their original names. Renamed objects are not deinstalled since their names differ from those in the product's released file set. If an object is is not deinstalled, the directory containing that object is also not deinstalled. **install + +** displays a warning message whenever it cannot deinstall an object.

-D
product_name
version

Deinstall (remove) the specified product (*product_name version*) from the target(s). *product_name* and *version* are the name and version number of the product as used in the name of the product's release index. You can only specify one product on the command line. At least the release index of *all* products currently installed on the target node must reside in the specified Authorized Area.

install + + does *not* deinstall an object (a file, directory, or link) in the specified product in two cases: first; if the object is part of another product installed on the target; second; if the object (judging by its name), is not an object that was originally released with the product. Such objects can include user-created files and files renamed by **install** when the product was updated previously. The release index of some products instructs **install** to rename, and thereby preserve, certain user-modifiable files, rather than simply overwrite them, by adding a date extension to their original names. Renamed objects are not deinstalled since their names differ from those in the product's released file set. If an object is is not deinstalled, the directory containing that object is also not deinstalled. **install + +** displays a warning message whenever it cannot deinstall an object.

install + + deinstalls a product even when another product installed on the target depends on the product being deinstalled.

Do not use **-D** to deinstall Domain/OS. To remove Domain/OS, you must initialize the disk using the **invol** utility. Also, deinstallation of patches is not recommended, since this removes possibly critical objects from a product, without replacing them.

-e Check whether all objects required for installing the product configuration 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, since the objects required on each target may differ when you are updating products.

-h Ignore hardware compatibility checking: force **install + +** to install all products in the product configuration on all specified targets, regardless of the ISP type of the products and the target nodes. Without **-h**, **install + +** does *not* install a product with an ISP type of a88k (a product released for Series 10000 nodes) to a node with an ISP type of m68k (a non-Series 10000 node), or vice versa, because the product may not run correctly or at all and, in some cases, may prevent the target from booting. Instead, **install + +** prints an error message. A situation where you might want to use **-h** would be if you are using an a88k node as a mothernode from which to boot m68k nodes diskless. You would want to install products, such as compilers, of both ISP types on the a88k mothernode, so the m68k nodes could run the m68k products when booted diskless.

-i Ignore product configuration size checking. Without **-i**, **install + +** checks that the product configuration is smaller than the amount of free disk space on the target(s). If it isn't, **install + +** displays a warning message, halts, and asks if you want to proceed. **-i** is most useful when you are installing product updates unattended (overnight perhaps) and don't want to have the installation halt unnecessarily. The space calculations are not totally accurate, so sometimes you are warned that there is insufficient space when this isn't the case, and sometimes you are *not* warned when there is not sufficient space.

-j Skip the configuration phase and directly install the existing product configuration specified with the **-c** or **-C** option. Answer any unanswered configuration questions with their default values (as defined in the respective product's release index). **-j** requires that you specify at least one existing configuration file on the command line with the **-c** or **-C** option. **-j** is incompatible with the **-a**, **-k**, and **-u** options.

-k Skip the configuration phase and directly install the existing product configuration specified with the **-c** or **-C** option, unless any of the configuration questions in the specified configuration are unanswered. In this case, **install + +** invokes **config** and forces you to answer these unanswered questions interactively. You can answer only the unanswered questions. You cannot

modify any other aspects of the specified product configuration. If you specify more than one configuration file on the command line and answer unanswered questions, the answers are *not* saved in any of the existing configuration files.

-k requires that you specify at least one existing configuration file on the command line with the **-c** or **-C** option. **-k** is incompatible with the **-a**, **-d**, and **-u** options.

-l

Install the product configuration by creating hard links from the target to the objects in the product directories of the Authorized Area, rather than by copying the objects from the Authorized Area. The target directory must be on the same node as the Authorized Area. Using **-l** saves disk space when you install products on the Authorized Area node by not duplicating objects.

A hard link is simply an additional reference to the same physical object. Therefore, if you modify (write to) an installed object that is hard linked to an Authorized Area — a user-modifiable Domain/OS file, perhaps — those modifications apply to the object "in" the Authorized Area as well. When the product is subsequently installed from this Authorized Area on other nodes, the modified object — not the original version — gets installed. So use caution when modifying hard-linked objects. If you delete an installed object that is hard linked to the Authorized Area, the "corresponding" object in the Authorized Area is not deleted (and vice versa) — the deletion just removes the name of the installed object and not the object itself.

-m

Do not respect product customization. If you are reinstalling or updating a product (rather than installing a product that has not been installed previously on the target node), replace any objects in the product that have been manually deleted, manually changed from a local copy to a link, or manually changed from a link to a local copy, as called for by the product configuration.

-m is the easiest way to get a clean configuration on a node whose software you suspect may have been inadvertently or erroneously modified or not quite properly installed. Also, use **-m** whenever you update a beta version of a product (to a later beta version or the fcs version) to ensure that any new objects in the later version are installed.

Even when **-m** is used, **install + +** does *not* overwrite a link with a directory or a directory with a link under certain conditions. These conditions are described in "Changing Copies to Links and Links to Copies" in the description of the **install** tool in this chapter.

-o Install to each target as many times as the target is specified on the command line or in a *target_list* file. Without **-o**, **install + +** installs only once to each target, even if the target is repeated on the command line or in a *target_list*. **-o** is primarily useful during long, unattended installations. By using **-o** and naming each target twice on the command line or in a *target_list*, you can improve the chances of a successful installation on each target. You can also use **-o** when you deinstall a product (**-D**), deinstall product subcomponents (**-d**), and reset ACLs (**-A**) to repeat the deinstallation or ACL-resetting attempt as many times as the target is listed.

-p Purge all previously created baseline files on the target(s). When you update a product, the earlier version of the product is not listed in the newly created baseline file. Therefore, do not use **-p** if you want to retain a comprehensive history of all products formerly installed on the node.

-r Execute the installation processes remotely on each target node, rather than locally on the node on which **install + +** is invoked. Without **-r**, **install + +** runs only on the node it is invoked from and installs (or deinstalls) the configuration on each target in succession, one after the other. When you install (or deinstall) products on more than one target, **-r** distributes the workload and is faster. The Server Process Manager (**/sys/spm/spm**) must be running on a target for the installation to execute remotely on that target. If **spm** is not running on a target, the installation executes locally just for that target; it still executes remotely on targets with **spm** running. The **-r** option is incompatible with the **-a** and **-u** options. Also, to use **-r** you must specify a new or existing configuration file on the command line with the **-c** or **-C** option; you cannot use **-r** and create a configuration that is saved only for the duration of the installation session.

For each remote target, **install + +** creates a log file in the source Authorized Area (*AA*). Each log file has the pathname *AA/node_namex*, where *node_name* is the name of the remote target node and *x* is an integer value. The log file records any any error or warning messages from the target and other pertinent installation information.

-t Test whether the configuration can be successfully installed from the Authorized Area to the target(s), without actually installing the configuration. If you are deinstalling a product (**-D**) or product subcomponents (**-d**), **-t** lists all objects that will be deleted and those that cannot.

-u Update every product already installed on the specified target(s) with the latest versions available in the Authorized Area. The configurations installed are the same as those resident on the target(s). However, if a configuration question has been added for a later version of a product, the default answer for this question is used. To determine which products to install and the configuration of those products, **install++** reads the latest baseline file on each target. You can also use **-u** with the **-A** option (reset ACLs) to reset the ACLs of *all* products installed on the target node(s) to their initial settings. **-u** is incompatible with the **-a**, **-c**, **-C**, **-j**, and **-k** options.

-v Report more information while **install++** runs. Without **-v**, **install++** displays an abbreviated listing. We recommend using the **-v** option to assist you in troubleshooting any installation errors and understanding any warnings.

-x Continue if an error is encountered. Without **-x**, **install++** aborts the installation when it encounters an error. You can often correct an installation error without having to rerun **install++** (or **install**).

Examples

In all of the following examples, the name of the Authorized Area is **//server/aa**.

To interactively configure and install one or more products on the node **//my_node**, without saving the defined product configuration,

```
//server/aa/install/tools/install++ -vxs //server/aa //my_node
```

To modify the product configuration defined in the configuration file **//server/cf.compilers**, and then install the modified configuration on the nodes listed in **//server/node_list**,

```
//server/aa/install/tools/install++ -vxs //server/aa -c //server/cf.compilers
  -n //server/node_list
```

To deinstall version 4.0 of the product **lisp** from the nodes **//bsa** and **//bsa2**,

```
//server/aa/install/tools/install ++ -vxs //server/aa -D lisp 4.0 //bsa //bsa2
```

The following sections provide additional information about **install + +** that is not directly related to the individual command-line options. Also see "install" in this chapter for information about the following topics related to the installation phase of **install + +**: objects created and used during the installation phase to monitor and control the installation process; disk space required by the install processes; the changing of local copies to links and links to copies; and things to do after the installation finishes.

Finding Your Way Around the Maze of Options

While **install + +** has many possible command line arguments, keep in mind that only two are required for installing products: the **-s** option to specify an Authorized Area containing the products to be configured and installed; and at least one target pathname, specified either directly on the command line (*target*) or with the **-n** *target_list* option.

A primary consideration when using **install + +** is specifying a configuration of products to install. You have five basic choices. You can

- Specify no configuration on the command line (by omitting the **-a**, **-c**, **-C**, and **-u**) options. **install + +** invokes the **config** program, enabling you to interactively define a configuration of products, and then installs the configuration you define. The configuration is not saved in a configuration file for later use.
- Specify the pathname of a non-existing configuration file using the **-c** *configuration_file* option. This is the same as specifying no configuration on the command line, except the configuration you define during the interactive configuration phase is saved in the specified file.
- Specify one or more existing configuration files using the **-c** or **-C** options. You can modify these configurations during the configuration phase or install them as is, depending on whether you use the **-j** or **-k** options.
- Use the **-a** (all) option to install the latest version of all products in the Authorized Area, using the default product configurations.
- Use the **-u** (update) option to update all products currently installed on the target node(s) with the latest versions available in the Authorized Area, using the same configurations as those currently installed.

Most of the other **install++** options are special purpose options. We suggest you familiarize yourself with them and use them at your own discretion. The only options we recommend that you use on a regular basis are the **-v** (verbose) and **-x** (continue on error) options. Using **-v** is particularly helpful when you need to analyze the installation transcript to troubleshoot problems.

Using install++ Versus config and install

install++ is a *wrapper* of the **config** and **install** tools: it combines the **config** and **install** tools, both of which you can execute independently, into a single process. **install++** has three command line options that **install** does not: **-C**, **-j**, and **-k**. Otherwise, using **install++** is nearly functionally equivalent to using **config** and **install** separately.

Which tools you use is largely a matter of personal preference. You may find **install++** easier and more expedient for one-shot installations of a product or a group of products, when you don't want to save the product configuration in a file for subsequent use. On the other hand, many users (particularly system administrators) prefer to use **config** and **install** so they can separate the configuration and installation processes. Also, **install++** is generally not suitable for unattended or remote installations on multiple targets, because of its interactive phase.

See Also

"Configuring Products," Chapter 5.
"config" and "install," this chapter.
"Configuration Files," Chapter 11.
Chapter 12, Installation Errors and Warnings.
AA/**install/help/install++.hlp**, the online help file for **install++**.

minst

minst [*authorized_area*]

The **minst** tool (*AA*/**install/tools/minst**) loads software products from distribution media into an Authorized Area and then optionally installs the products from the Authorized Area to one or more target nodes.

The parameters are as follows:

authorized_area

Specify the pathname of the Authorized Area into which you want to load the products. The Authorized Area directory must exist and must reside on the node to which the tape drive or floppy drive is connected. Specifying an Authorized Area on the command line is optional. If you omit the argument, **minst** prompts you for the Authorized Area pathname.

The directory you specify (either on the command line or in response to the **minst** prompt) can be *any* existing directory — it does not have to be one that already contains Authorized Area components. If the directory is not an existing Authorized Area, **minst** creates an Authorized Area at that directory. That is, **minst** creates the requisite **install** subdirectory (*authorized_area_directory*/**install**) and restores the products and associated administrative objects from media into the appropriate subdirectories. If you use **minst** to load an optional product, we recommend that you specify an existing Authorized Area. Optional product media either contains no installation tools or pre-SR10.4 installation tools. So if you specify a non-Authorized Area directory, the resulting Authorized Area will either contain no tools or outdated versions of the tools.

Examples

To start **minst** (located in the Authorized Area **//server/aa**) and to specify the target Authorized Area interactively,

//server/aa/install/tools/minst

To use **minst** to load products into the Authorized Area named **//server/aa**,

//server/aa/install/tools/minst //server/aa

minst combines the functionality of the **distaa** and **install++** tools into a single process. It transparently invokes these tools and adds an interactive, textual interface that asks you several questions, rather than have you specify installation options as command-line arguments.

Because **minst** is a self–guiding program, we do not describe the individual **minst** questions here. Rather, the following sections give you an overview of the program and highlight some of its most important aspects.

Why use minst?

You don't have to use **minst** to load and install products. You can do everything **minst** does by invoking other commands directly: **rbak** and **distaa** to load administrative objects and products, respectively, into an Authorized Area; and **config** and **install** (or **install++**) to configure and install products from an Authorized Area to target nodes. (This is the approach we take in Chapter 5 for loading and installing optional products, Domain/OS updates, PSKs, and patches.)

The primary difference between **minst** and using the tools directly is that **minst** leads you through the entire process step-by-step, with detailed instructions and numerous prompts along the way. It takes less prior knowledge to use **minst** and it performs many tasks automatically for you. Therefore, **minst** is well-suited for new and infrequent user of the installation tools, and for the first time install of Domain/OS from media, a fairly complex task.

However, **minst** is less flexible than the individual tools and does not offer as many options; many users prefer to separate the loading and installation processes; and due to the greater complexity and scope of the **minst** program, any problems you encounter using **minst** may be more difficult to troubleshoot.

Starting minst

minst starts automatically whenever you log in after booting a node from distribution media. This is to simplify the first-time installation of Domain/OS from media on an initialized node. If you choose not to use **minst** for the first time installation of Domain/OS, you can quit **minst** after it automatically starts up, create a shell with the **cp** command, and then use the other installation tools.

To start **minst** yourself (for the installation of optional product, patches, and Domain/OS updates), enter the command *AA*/**install/tools/minst** (where *AA* is the pathname of an Authorized Area) at a shell prompt. Optionally follow the command with the *authorized_area* argument.

Two Phases: Load and Install

minst executes in two phases: a load phase, and then an installation phase. In the load phase, **minst** restores products and associated administrative objects from the distribution media into an Authorized Area. In the installation phase, **minst** installs an operational configuration of the products on one or more target nodes. You cannot skip the load phase. However, you can quit after the load phase and skip the installation phase.

Load Phase

In the load phase, **minst** first invokes the **rbak** command to restore administrative objects from the distribution media into the Authorized Area. It then invokes the **distaa** installation tool to restore the actual products and their release indexes into the Authorized Area.

The administrative objects restored by **rbak** include: the installation tools (in the **install/tools** directory), if any, and their help files (**install/help**); the media TOC (Table of Contents) file (**install/toc**); the product release documentation (**install/doc/apollo**); and the selection and override files for the products (**install/templates**). **minst** prompts you to read the online product release notes after it **rbak**s the administrative objects and before it loads products.

Also note:

- The installation tools and their help files are currently released with Domain/OS and PSKs only. Formerly (approximately prior to the release of version 10.3 of Domain/OS), the tools were released on the distribution media of all products.

- If a set of installation tools already resides in the Authorized Area, **minst** loads the tools (if present) from media only if they are a later version than the resident tools.
- If the media contains more than one product, **minst** unconditionally loads the release documentation and the selection and override files for *all* products. If you load only some of the products, you can delete the documentation and selection and override files for the products you did not load to save space on the Authorized Area node. Do this with standard UNIX and Aegis commands after **minst** completes execution.
- If the set of distribution media contains more than one tape volume, **minst** prompts you in sequence to insert each tape into the drive. However, if you select a product or product subset that does *not* contain any objects on one of the tapes, **minst** skips the prompt for that particular tape volume.

Installation Phase

When **minst** finishes loading products into the Authorized Area, it gives you the option of installing the products from the Authorized Area to specified target nodes. To install the products, **minst** invokes the **install++** installation tool. **install++** in turn invokes the **config** and **install** installation tools to configure the products for installation and to transfer the products from the Authorized Area to the target nodes, respectively. Prior to installing the products, **minst** prompts you to read the online product release notes.

Novice and Expert Mode

You can run **minst** in either novice mode or expert mode. **minst** prompts you to select a mode early in its execution. The modes differ in the amount of information that **minst** asks you to provide. Novice mode asks fewer questions and transparently provides default answers for most of the questions asked in expert mode. This makes novice mode easier to use but less flexible. Conversely, expert mode is more complex and requires more knowledge, but gives you greater control over the loading and installation process. Specifically,

- If the media contains more than one product, novice mode unconditionally loads *all* products from media into the Authorized Area; expert mode allows you to select which products to load.

- Novice mode enables you to select which products to install from among the products loaded; expert mode enables you to select from among all products in the Authorized Area, including those just loaded. (If you load Domain/OS, however, both modes allow you to install Domain/OS only.)
- Novice mode unconditionally installs all selected products with default configurations; expert mode allows you to install the default configurations *or* to interactively configure the products. (If you choose to interactively configure products, **minst** invokes the **config** program, which presents you with a series of HP-supplied configuration questions for each product. For information about using **config**, see the "config" section in this chapter.)
- Novice mode allows you to install products only on the node connected to the tape or floppy drive; expert mode allows you to specify multiple targets.
- Novice mode unconditionally installs products on the Authorized Area node with hard links to the products in the Authorized Area (to conserve disk space); expert mode allows you to install products on the Authorized Area node with hard links *or* as local copies.
- In novice mode, when you reinstall a product or install a product update (rather than install a product that has not been previously installed on the node), any customization to the resident product is overwritten. Files that have been manually deleted are replaced, and objects that have been manually changed from links to local copies (or vice-versa) are overwritten. Expert mode allows you to overwrite product customization *or* to leave it intact.
- When you install Domain/OS, novice mode installs only the **/sau** directory for the machine type of the target mode. This means other machine types cannot boot diskless from the target node. Expert mode allows you to choose which **/sau** directories to install.
- Expert mode enables you to save your answers to the **minst** prompts in a file so you can optionally reuse the answers on subsequent runs of **minst**; novice mode does not.

See Also

"Step 4. Load and Install Domain/OS," Chapter 1.
"Loading Products from Media into an Authorized Area," Chapter 5.
"Configuring Products," Chapter 5.
"config," "distaa," "install+ +," this Chapter.
AA/**install/help/minst.hlp**, the online help file for **minst**.

mrgri

mrgri {**-merge** | **-cmpexe**} [**-i**] **-s** *source_AA*
[**-t** *target_AA*] [**-p** *new_name*] [**-v** *new_version*]
primary_product secondary_product

The **mrgri** tool (*AA*/**install/tools/mrgri**) merges two products in an Authorized Area into a single product with a single release index. **mrgri** stands for "merge release index."

The parameters are as follows:

-merge
-cmpexe

You must use either the **-merge** or the **-cmpexe** switch.

Use **-merge** to merge two products that have the same ISP type (m68k, a88k, or cmpexe) or where the *primary_product* is a compound (cmpexe'd) product. In practice, **-merge** is almost always used to merge a patch with a product or a Product Support Kit (PSK) with Domain/OS.

Use **-cmpexe** to create a compound product (also called a cmpexe'd product); that is, to merge two products with the same name but with different ISP types. If the two products do not have different ISP types, **mrgri** returns an error.

-i

Display more informational messages. List the source and destination pathname of each object copied and additional execution milestones.

-s *source_AA*

Specify the pathname of the Authorized Area (*source_AA*) containing the products to be merged (containing *primary_product* and *secondary_product*). The products to be merged must reside in the same Authorized Area.

-t *target_AA*

Specify the pathname of an Authorized Area in which to place the resulting merged product. **mrgri** creates a new product directory containing the merged product and the merged release index in the specified target Authorized Area. The constituent products (*primary_product* and *secondary_product*) are retained in the *source_AA* in their original state. Without **-t**, **mrgri** creates the merged product in the *source_AA*.

target_AA does not need to exist. If it doesn't, **mrgri** creates it and an **install** subdirectory in which to place the product directory for the merged product. If *target_AA* exists but does not have an **install** subdirectory, **mrgri** creates the **install** subdirectory.

-t is useful when space on the disk containing the *source_AA* is limited. We also recommend using **-t** as a means of placing all compound products in their own Authorized Area (separate from a88k and m68k products). **-t** is not generally recommended when you are merging a patch with a product, since a patched product rarely takes up much more space than the unpatched product. Also, **-t** significantly increases the time for merging, especially when you are patching Domain/OS.

-p *new_name*

Specify a new product name (*new_name*) to be applied to the merged product. The merged product is created as a third, separate product with the specified name and the version number of the *primary_product*. A new product directory containing the merged product and the merged release index are created. The constituent products (*primary_product* and *secondary_product*) are retained in the *source_AA* in their original state.

There is rarely a need for this option, and its use is not recommended. It may break product dependencies specified in the products' release indexes. For example, suppose you merge a patch with version 10.4 of Domain/OS (**os.v.10.4**), use **-p** to change the product name from **os** to something else, and then install the newly-named OS on a node running SR10.3. An attempt to install an optional product that depends on SR10.4 on that node will fail, since the node's baseline file will *not* indicate that **os.v.10.4** is installed.

-v *new_version*

Specify a new version number (*new_version*) to be applied to the merged product. The merged product is created as a third, separate product with the specified version number and the name of the *primary_product*. A new product directory containing the merged product and the merged release index are created. The constituent products (*primary_product* and *secondary_product*) are retained in the *source_AA* in their original state.

This is the recommended way to distinguish a merged product from its constituent products and to retain the ability to install each of the constituent products from the Authorized Area. **-v**, however, is generally not necessary nor recommended when you merge a patch with a product. When you create a compound product and use **-v**, we recommend that you specify a version number that consists of the extension **cmpexe** added to the version number of

the a88k product (for example, **os.v.10.4.p.cmpexe**). We explain why in "Version Numbering Scheme" later in this section.

primary_product
secondary_product

Specify the release index names (**ri.apollo.***product_name***.v.***version*) of the two products to be merged. Without the **-p**, **-t**, or **-v** option, **mrgri** overwrites the *primary_product* with the *secondary_product* to create the merged or compound product; the *secondary_product* is retained in its original state. The product name and version number of the *primary_product* become the name and version number of the merged product. Use **-p**, **-v**, or **-t** if you want to retain the ability to install the *primary_product* from the source Authorized Area.

When you merge a patch with a product, make the product to be patched the *primary_product* and the patch the *secondary_product*. When you merge a PSK with Domain/OS, make Domain/OS the *primary_product* and the PSK the *secondary_product*. In both cases, you may optionally remove the patch or PSK from the Authorized Area after the merge. When you create a compound product, make the larger product the *primary_product* (if you don't use **-p**, **-v**, or **-t**) to reduce the merge time.

Examples

To apply the patch named **pd91_m0205** to version 3.1 of the product **gmr3d** in the Authorized Area at **//server/aa**,

```
//server/aa/install/tools/mrgri -merge -s //server/aa ri.apollo.gmr3d.v.3.1
  ri.apollo.pd91_m0205.v.1.0
```

To create a compound (cmpexe'd) version of version 10.4 of Domain/OS, located in the Authorized Area **//server/aa**,

```
//server/aa/install/tools/mrgri -cmpexe -s //server/aa -t //server/aa_cmpexe
  -v 10.4.p.cmpexe ri.apollo.os.v.10.4.p ri.apollo.os.v.10.4
```

The resulting compound product is created in the Authorized Area at **//server/aa_cmpexe**. Its product directory and release index have the name **ri.apollo.os.v.10.4.p.cmpexe**.

mrgri has three main uses: merging patches (bug fixes) with products, merging Product Support Kits (PSKs) with Domain/OS, and creating compound products. It is an optional, non-interactive tool intended primarily for system administrators. It is not required for loading, configuring, or installing any products.

Merging Patches and PSKs

In the Domain installation model, each patch is released and handled as an individual product, with its own product name, version number, release index, and product directory. A PSK is a subset of Domain/OS that provides support for a new hardware platform or peripheral device, or that provides new Domain/OS functionality. Like patches, PSKs are released and handled as individual products.

You can use **mrgri** to merge a patch with the product it patches in an Authorized Area and to merge a PSK with the appropriate version of Domain/OS. In both cases (patches and PSKs), installing the merged product has the same result as installing the constituent products (the patch and the broken product, or the PSK and Domain/OS) separately. The advantage of merging patches and merging PSKs is that you have only one product in an Authorized Area to manage and install instead of two.

Creating Compound Products

a88k and m68k ISP Types With the advent of the Series 10000 workstations and servers, Apollo releases two versions of most products: a version that runs on Series 10000 machines and a version that runs on machines based on Motorola's 68000 series of microprocessors. Two versions are required because the Series 10000 machine has a different architecture, called PRISM (Parallel Reduced Instruction Set Multiprocessor). We encapsulate these two different CPU architectures with the concept of an ISP (Instruction Set Processor) type. Series 10000 nodes belong to the a88k ISP type; all other nodes belong to the m68k ISP type.

We distinguish the two different ISP types in product releases by adding the extension **.p** to the version field of the release index name for a88k products. For example, **ri.apollo.os.v.10.4.p** is the name of the release index for version 10.4 of Domain/OS that runs on Series 10000 nodes, whereas **ri.apollo.os.v.10.4** is the name for the version that runs on m68k nodes.

Advantages of Compound Products You can sometimes use **mrgri** to combine two products that differ only in ISP type into a single product, called a compound product. The file type of executable objects in a compound product is called **cmpexe** — short for "compound executable." You can install and run compound products on nodes of either ISP type. Additionally, if you create and install a compound version of Domain/OS, nodes of either ISP type can boot diskless off the node running the compound OS. And you can link Domain/OS components on nodes of either ISP type to the compound Domain/OS node.

How mrgri Creates Compound Products When **mrgri** encounters two executable objects in the constituent products with the same name but different ISP types, it combines the two objects to create a compound executable (an object with a file type of **cmpexe**). The cmpexe object is an executable containing the code for both ISP types. When **mrgri** encounters two non–executable objects with the same name, it uses the object that has the later date and time stamp or the object in the primary product, if the date and time stamps are the same.

Size of Compound Products Since each **cmpexe** object in a compound product contains executable code for both ISP types, a compound product is larger than either one of its constituent products. 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 constituent products. A good rule of thumb is that a compound product requires about 70 percent of the disk space occupied by the sum of the two constituent products.

Version Numbering Scheme When you create a compound product as a third product with a new version number (by using **–v**), we recommend you specify a version number consisting of the extension **cmpexe** added to the version number of the a88k product (**10.4.p.cmpexe**, for example). This ensures that the **install** tool recognizes the version number of the compound product as lexically "greater than" both the m68k and a88k versions. Then, when you install products on a node with the **–u** option (update every product on the node with the latest version available in an Authorized Area) or the **–a** option (install the latest version of all products available in an Authorized Area), the compound product gets installed rather than the a88k version or m68k version, whenever a potential conflict exists. This version-number scheme also ensures that you can successfully install other products dependent on the a88k or m68k version, or dependent on a "greater-than" version of the a88k or m68k version.

Segregating Compound Products Often, it is a good idea to keep compound products in an Authorized Area that does not contain m68k and a88k products. You can do this by using **mrgri**'s **-t** option. When a compound product resides in the same Authorized Area as its a88k and m68k counterparts, the **install -a** and **-u** options treat the compound product as "greater than" the a88k and m68k versions (provided you use our recommended version numbering scheme). However, this may not always be what you want. By segregating compound products in another Authorized Area, you can exercise greater control over their installation relative to the a88k and m68k versions.

Merge Only Designated Products

Theoretically you can merge any two products. However, for two products to merge successfully, or for a merged product to function correctly, the products must be designed to be merged. Therefore, do not indiscriminately merge any two products. Unless otherwise stated in the product release notes, you can assume that a patch can be successfully merged with the product it patches, a PSK with the version of Domain/OS that it augments, and an a88k version of Domain/OS with the corresponding m68k version. However, for other products, check the product release notes to determine whether the product can be merged, which versions can be merged, and what restrictions you must follow, if any, when installing and using the merged products.

mrgri checks only that the ISP types of the two products conform to the rules described for the **-merge** and **-cmpexe** switches. It does not check that the products or product release indexes have been designed to be merged. Therefore, even when **mrgri** completes execution successfully, this does not necessarily mean that the merged product will function properly.

Cannot Unmerge Products

You cannot "unmerge" a product created by **mrgri** to recover its constituent products. If you create a merged product by allowing **mrgri** to overwrite one of the individual products (by not using **-p**, **-t**, or **-v**), or if you delete one or both of the constituent product release directories after merging them, you cannot recover that product from the merged product. You must reload it from the distribution media or copy it from another Authorized Area.

See Also

Chapter 8, Merging Products in an Authorized Area.
"Product Release Directories," Chapter 11.
AA/**install/help/mrgri.hlp**, the online help file for **mrgri**.

Components and Structure of an Authorized Area

A detailed description of the components in an Authorized Area, and how these components relate to distribution media

Components and Structure of an Authorized Area

This chapter provides a detailed description of each of the components in an Authorized Area. It then describes how Authorized Areas and their components are organized and formatted on distribution media (cartridge tape, magnetic tape, and floppy disk).

This information is not required for performing the procedures in Part 1 of this manual. It is intended primarily for users who are responsible for managing Authorized Areas and who perform product installations frequently. Before you read this chapter, make sure you have read "The Authorized Area: A Key Concept," in the introductory chapter of Part 1. We also suggest you consult the "install" section in Chapter 10 for a description of the **/install** directory present on every node, which contains non-Authorized objects created and used by the installation tools.

Table 11-1 lists the Authorized Area components described in this chapter and their location within the Authorized Area directory tree.

Table 11-1 **Authorized Area Components and Their Location**

Component	Location
media TOC files	**install/toc** directory
install tool directories	**install/tools** **install/minst**
help files (for install tools)	**install/help/***tool_name***.hlp**
product directories	**install/** **ri.apollo.***product***.v.***version*
selection files	HP-supplied selection files: **install/templates/apollo/** *product***.v.***version***/aa.***name*
override files	HP-supplied override files: **install/templates/apollo/** *product***.v.***version***/ov.***name* active override files: **install/overrides/** **ri.apollo.***product***.v.***version*
configuration files	default configuration files: **install/templates/apollo/** *product***.v.***version***/cf.***name*
product release documentation	**install/doc/apollo** directory
installation log file	**install/log_file**
excludes.list file	**install/overrides/excludes.list**

Media TOC Files

When you load products from distribution media into an Authorized Area (using the procedures in this manual), a media TOC (Table of Contents) file is loaded from file 1 on the media to the directory **install/toc** in the Authorized Area. Each TOC file is an ASCII file that identifies the particular piece of media and lists all products and product subcomponents on the media. Together, the TOC files in the **install/toc** directory provide a record of the media source of all products loaded into the Authorized Area.

TOC files have names of the form

toc.apollo.*volume_id*.*media_type*

where

volume_id is a volume identifier for the piece of media. Volume identifiers for Domain products consist of two uppercase alphabetic characters followed by two hexadecimal and two decimal numeric characters.

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

For example, a TOC file for a cartridge tape containing a Domain product might be named **toc.apollo.ST0104.c**.

See Also

"Loading Products from Media into an Authorized Area," Chapter 5.
"Relation of Authorized Areas to Distribution Media," this chapter.

Installation Tool Directories

The executable versions of the installation tools and the data files to support them are contained in two Authorized Area subdirectories: **install/tools** and **install/minst.**

install/tools Directory

The **install/tools** directory contains the executable installation tools for nodes running an SR10.*x* version of the Domain/OS operating system. The tools include **cfgsa, config, distaa, install, install++, mrgri,** and **minst.** The directory also contains other programs and data files required by the installation tools, which you generally need not concern yourself with.

install/minst Directory

The **install/minst** directory contains data files required by the **minst** installation tool. You need not concern yourself with this directory. Do not alter the directory or the objects in it.

See Also

"Loading Products from Media into an Authorized Area," Chapter 5.
"Loading the Installation Tools into an Authorized Area," Chapter 9.
Chapter 10, Installation Tool Reference.
"Relation of Authorized Areas to Distribution Media," this chapter.

Help Files

An online help file for each installation tool is contained in the Authorized Area directory **install/help**. The name of each file is the name of the installation tool with the extension **.hlp** added to it; for example, **distaa.hlp**. Each help file provides a detailed description of the respective tool and its command line options, using the same format as UNIX man pages. Chapter 10 of this manual also provides a comprehensive description of each tool, using a somewhat different and more detailed format.

See Also

"Loading Products from Media into an Authorized Area," Chapter 5.
"Loading the Installation Tools into an Authorized Area," Chapter 9.
Chapter 10, Installation Tool Reference.
"Relation of Authorized Areas to Distribution Media," this chapter.

Product Release Directories

The product release directories form the core of an Authorized Area. There is one product release directory for each product that you load into an Authorized Area. If the Authorized Area contains more than one version of a product, there is a directory for each version. Each directory contains the product's release index and the objects that actually comprise the product.

Names of Product Release Directories

Product release directories have pathnames of the form

install/ri.apollo.*product_name***.v.***version*

where

product_name is the name of the software product recognized by the installation tools. Types of products include Domain/OS, optional products, individual patches, and Product Support Kits (PSKs).

version is the version number of the product. Products designed to run on Series 10000 (a88k) machines have the extension **.p** added to the version field.

Product Objects

A product directory contains the files and directories that actually make up the product. When you install a product, the **install** program copies these objects to the target node and builds an operational configuration of the product. The objects in the Authorized Area do *not* comprise an operational configuration of the product.

Product Release Indexes

In addition to a product's objects, each product directory contains a release index file for the respective product. Each release index file has the same name as the product's release directory: **ri.apollo.***product_name***.v.***version*. A release index is, in a sense, a master blueprint of the product. It is used by installation tools to build and install a viable, operational configuration of the product on a node. Specifically, a release index defines

- The configuration questions displayed by the **config** and **cfgsa** tools, and the default answers for those questions.
- The subcomponents of the product.
- The initial settings of the ACLs (Access Control Lists) for the installed product's objects. (The ACLs of a product's constituent objects in an Authorized Area have no effect on the ACLs of the installed product.)
- Any dependencies that the product has on other products. For example, the release index might specify that the product can only be installed on nodes running version 10.4 or greater of Domain/OS. Most dependencies are dependencies on a particular version of Domain/OS.
- How to install the product's subcomponents in a way consistent with the answers supplied to the configuration questions.

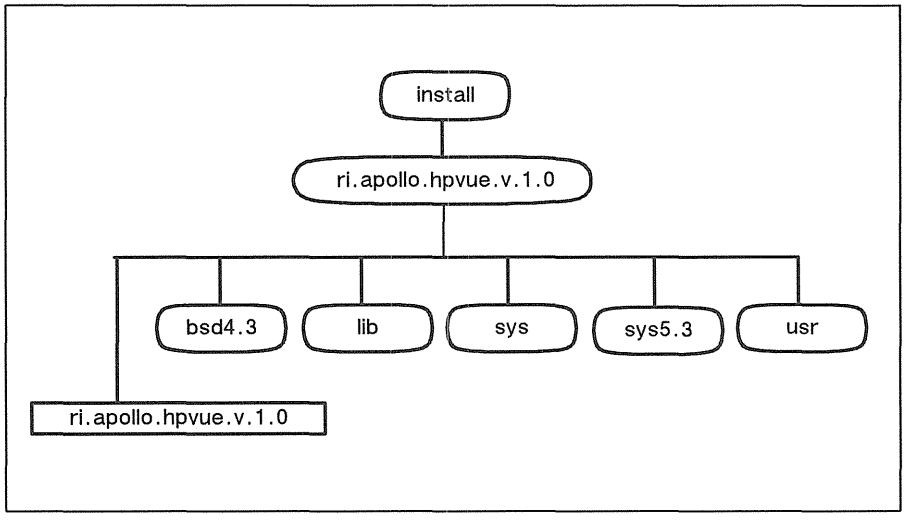
Note that when you install a product update, the release index of the product being updated must reside in the Authorized Area for the **install** program to correctly install the newer version of the product.

Examples

Here are some sample names of product release directories:

ri.apollo.os.v.10.4	version 10.4 of Domain/OS
ri.apollo.os.v.10.4.p	version 10.4 of Domain/OS for Series 10000 machines
ri.apollo.psk8.v.10.3	PSK 8, a set of files to augment version 10.3 of Domain/OS
ri.apollo.lisp.v.4.0	version 4.0 of the Lisp compiler
ri.apollo.pd91_m0224.v.1.0	a patch for m68k machines

The following figure shows the top-level contents of the product release directory for version 1.0 of the product **hpvue**.



Structure of a Product Release Directory

See Also

"mrgri," Chapter 10.

Selection Files

Selection files select a particular product or a subset of a product to be loaded from distribution media into an Authorized Area. You supply the name of the appropriate selection file as a command-line argument to the **distaa** tool to load the desired product or product subset.

HP-Supplied Selection Files

Every Domain software product ships with at least one predefined selection file that selects the entire product for loading. You use this type of selection files when there is more than one product on the distribution media and you want to load only selected products, not all of them. Otherwise, you can use **distaa**'s **-a** option, which loads all products in their entirety from media.

Some products, most notably Domain/OS, also ship with selection files that select subsets of the product for loading (for example, an **aegis_small** configuration of Domain/OS). This allows you to save disk space on the Authorized Area node by not loading components of a product that you don't anticipate using at your site.

HP-supplied selection files have pathnames of the form

authorized_area/**install/templates/apollo/***product_name*.**v.***version*/**aa.***name*

For example, the name of the selection file for version 10.4 of Domain/OS that loads an **aegis_small** environment is *AA*/**install/templates/apollo/os.v.10.4/aa.aegis_small**. The names (**aa.***name*) of a product's selection files, and the subcomponents selected by each selection file, are described in the product's release notes.

User-Created Selection Files

You can use the **cfgsa** tool to create your own selection files that define subsets of a product. You can then use these selection files with **distaa** to load the subsets into an Authorized Area. These selection files have the name **aa.***name*, where *name* is a character string of your choice. You can place these files anywhere on disk; they do not need to reside in an Authorized Area or in any particular location within an Authorized Area.

Relationship to Override Files

Every selection file (whether HP-supplied or user-created) has a corresponding override file. The override file, once activated, restricts the full set of configuration options defined in a product's release index to a set that is consistent with the set of product components associated with the selection file. HP-supplied override files reside in the same directory as the selection files. They have the same name, except the prefix **ov** is used instead of **aa**. The names of user–created override files also have a prefix of **ov**.

Whenever you use a selection file with **distaa** to load a product subset into an Authorized Area, you should activate the corresponding override file (see "Override Files," this chapter). If you don't, a user can potentially define a product configuration that includes product components that do not physically reside in the Authorized Area. This causes numerous errors at installation time.

Editing a Selection File to Distribute a Product

A selection file is an ASCII file that consists of a series of move commands for each component of the product.

Sample Selection File

```
/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      -rootaa
move   ri.apollo.dialog.v.3.4   /examples -nil
```

The first field after each **move** command is the product's release index name. The second field is the name of a component of the product, as defined in the product's release index. The third field specifies the load destination of the component. Initially, the value of the destination field is either **–rootaa** (root Authorized Area) or **–nil**. If the value is **–rootaa**, **distaa** loads the component into the Authorized Area specified on the **distaa** command line. If the value is **–nil**, **distaa** does not load the component at all.

If a product is too large to fit on one disk, you can distribute the product among more than one Authorized Area (located on different disks) as you load it from media. To do this, you edit the appropriate selection file, replacing **-rootaa** with the pathname of the desired Authorized Area for those components you want to distribute. (Do not specify the Authorized Area that you specify on the **distaa** command line.) The directory you specify must exist, although no Authorized Area components, including the **install** subdirectory, need already reside in the directory. After you edit the selection file, you supply the pathname of this selection file as a command-line argument to **distaa** to load the product into the specified Authorized Areas.

When you manually edit a selection file, only change instances of **-rootaa** to the desired Authorized Area. Do not change any destination fields to **-nil**, and do not edit any of the other fields. To exclude subcomponents from loading (to set their destination fields to **-nil**), use the **cfgsa** tool.

See Also

"Loading Products from Media into an Authorized Area," Chapter 5.
Chapter 7, Defining and Loading a Custom Product Subset.
"cfgsa"and "distaa," Chapter 10.
"Override Files" and "Relation of Authorized Areas to Distribution Media," this chapter.

Override Files

An override file restricts the full range of configuration options allowed for a given product (the set of configuration options defined in the product's release index). You can create override files with the **cfgsa** tool or use override files that ship with a product.

An override file (once activated) acts as a constraining filter on both the creation and installation of configurations of a product:

- The configuration questions and possible answers displayed by the **config** program for the product reflect the restrictions defined by the override file; users are forced to create product configurations that conform to the restrictions.
- If a user installs the product with a default configuration file or a configuration file created prior to the restrictions, the override file takes precedence; that is, any configuration choices specified by the configuration file but not allowed by the override file are overridden.
- Similarly, if a user installs products with the **-a** option (install the latest version of all products in the Authorized Area) or the **-u** option (update all products on the node with the latest version in the Authorized Area), the configuration restrictions defined by any active override file for a product are applied to the installation.

HP-Supplied Override Files

For every selection file that ships with a product, a corresponding override file is supplied. Each of these override files defines a set of configuration options that is consistent with the set of product components associated with the selection file. For example, each of the Domain/OS selection files that selects an Aegis environment only for loading has a corresponding override file that removes all configuration options related to the BSD and SysV UNIX environments.

HP-supplied override files have pathnames of the form

authorized_area/**install/templates/apollo/***product_name*.**v.***version*/**ov.***name*

For example, the name of the override file for a small-Aegis environment of version 10.4 of Domain/OS is:

authorized_area/**install/templates/apollo/os.v.10.4/ov.aegis_small**

User-Created Override Files

You can create your own override files for a product with the **cfgsa** tool. With **cfgsa**'s **save** command, you can create just an override file. Or, with **cfgsa**'s **generate** command, you can create both a selection file and a corresponding override file. The **save** command creates an active override file named

authorized_area/**install/overrides/ri.apollo.***product_name*.**v.***version*

The **generate** command allows you to create an override file named **ov.***name* (where *name* is string of your choice) in any directory. The file does not need to reside in an Authorized Area or in any particular location within an Authorized Area.

Activating Override Files

For an override file's configuration restrictions to take effect, the file must reside in the Authorized Area directory **install/overrides** and have the same name as the respective product's release index: **ri.apollo.***product_name*.**v.***version*. Such override files are said to be *active*. Only one override file per product can be active at a time. The installation tools ignore any override files not in the **install/overrides** directory.

Whenever you use a selection file with **distaa** to load a product subset into an Authorized Area, you should activate the corresponding override file. To do this, you move or copy the override file (using standard Aegis or UNIX commands) into the *AA*/**install/overrides** directory, and name it **ri.apollo.***product_name*.**v.***version*. If you don't, a user can potentially define a product configuration that includes product components that do not physically reside in the Authorized Area. This causes numerous errors at installation time.

Note that you can activate *any* override file for a product, provided the set of product components in the Authorized Area is a superset of the restricted set of components defined by the override file. For example, even if a full-blown configuration of Domain/OS including all three environments (Aegis, SysV, and BSD) resides in an Authorized Area, you can restrict users to **aegis_small** configurations simply by activating the HP-supplied **aegis_small** override file.

Deactivating Override Files

To restore the full set of configuration options for a product (as defined by the product's release index), simply delete, move, or rename the active override file, using the appropriate Aegis or UNIX commands. But do this only if the entire product (not a subset) resides in the Authorized Area. Or, if a product subset resides in the Authorized Area, remove an active override file only if you replace it with another override file that corresponds to the product subset (or a set of product components that is smaller than the subset). Otherwise, a user can potentially define a product configuration that includes product components that do not physically reside in the Authorized Area. This causes numerous errors at installation time.

See Also

"Loading Products from Media into an Authorized Area," Chapter 5.
Chapter 6, Restricting Product Configurations.
Chapter 7, Defining and Loading a Custom Product Subset.
"cfgsa," Chapter 10.
"Selection Files" and "Relation of Authorized Areas to Distribution Media," this chapter.

Configuration Files

Most Domain software products are not installed on an all-or-none basis. Rather, most products ship with a set of configuration options that enable you to specify precisely how the product is installed on a node. The configuration options enable you to specify which optional subcomponents of a product you want to install, whether to install the subcomponents as local copies or as links to other nodes, and other operational aspects of the product. A product's release index defines all of the possible configurations of the product that can be installed; a configuration file defines one of these possible configurations.

Default Configuration Files

Every Domain software product ships with one predefined configuration file that defines a default configuration for that product. These files have pathnames of the form

authorized_area/**install/templates/apollo/***product_name*.**v.***version*/**cf.***product_name*

For example, the pathname of the default configuration file for version 10.4 of Domain/OS is *AA*/**install/templates/apollo/os.v.10.4/cf.os**. HP-supplied selection files and override files reside in the same directory as the default configuration files.

You can supply the name of a predefined configuration file as a command line argument to the **install** or **install++** tool to install the default configuration of a product. Beyond this use, you need not concern yourself with default configuration files. To view a product's default configuration (assuming no override file for the product is active), you can run the **config** program, select the product, and then execute **config**'s **show queries** command.

User-Defined Configuration Files

Typically, you create your own configuration files with the **config** tool (by invoking it directly or indirectly with **install++**). You can place these files anywhere on disk; they do not need to reside in an Authorized Area or in any particular location within an Authorized Area.

See Also

"Loading Products from Media into an Authorized Area," Chapter 5.
"config" and "install++," Chapter 10.
"Override Files" and "Relation of Authorized Areas to Distribution Media," this chapter.

Product Release Documentation

The online release notes and other release documentation for Domain software products are contained in the Authorized Area directory **install/doc/apollo**. Each release document has a name of the form *product_name*.v.*version*__*doc_name*. For example, the file containing the release notes for version 10.4 of Domain/OS is **os.v.10.4__notes**. Whenever you install a product on a node, the release documentation for that document is unconditionally copied to the directory **/install/doc/apollo** on the target node.

See Also

"Loading Products from Media into an Authorized Area," Chapter 5.
"Relation of Authorized Areas to Distribution Media," this chapter.

Installation Log File

Whenever a user invokes the **install** tool (either directly or indirectly through the **install++** program), the invocation is logged in a log file in the Authorized Area. The pathname of the log file is *AA*/**install/log_file**, where *AA* is the name of the source Authorized Area specified by the **-s** switch on the **install** or **install++** command line.

The following information is logged:

- The name of the user who invoked **install** and the date and time.
- The command line used.
- The name of the node from which **install** was invoked and the version of Domain/OS running on that node.

excludes.list File

You can exclude product files in an Authorized Area from being installed on target nodes. To do this, you create an ASCII file named **excludes.list** in the Authorized Area's **install/overrides** directory. In the **excludes.list** file, you list the files (one per line) that you do not want installed. Use the absolute pathnames (pathnames beginning with the node entry directory, /) that the files would have if they were installed, not the pathnames of the files as they reside in the product directory in the Authorized Area. Whenever you install products from this Authorized Area, **install** checks the file and does not install the files listed.

You can also create a **preserve.list** file in a *target* node's **/install** directory that lists files on the target node that you do not want overwritten when products are installed on the node. This file is fully described in the "install" section in Chapter 10.

See Also

"install," Chapter 10.

Relation of Authorized Areas to Distribution Media

Domain products are potentially distributed on three types of distribution media: cartridge tape, magnetic tape, and floppy disk. All media types are in **wbak** format. **wbak** format is a format created by the **wbak** command, which HP uses to write Domain objects to the media. Files 2 through *n* on the media contain the product release indexes and the actual products.

File 1 contains the following administrative objects, which are used in an Authorized Area:

- Installation tools (in **install/tools, install/minst**).
- Help files for the installation tools (in **install/help**).
- HP-supplied selection files, override files, and configuration files (in **install/templates/apollo**).
- Online release documentation (in **install/doc/apollo**).
- TOC files (in **install/toc**).

The installation tools and their help files are currently distributed with Domain/OS only. Prior to the release of version 10.3 of Domain/OS (approximately), the tools were also distributed with optional products.

The **minst** tool automatically loads the objects in file 1 into an Authorized Area, in addition to loading the products and product release indexes. If file 1 contains installation tools, **minst** loads the tools only if they are a later version than the tools currently in the specified Authorized Area. The **distaa** tool, however, loads only the products and product release indexes; it does not load any objects from file 1. Therefore, if you use **distaa** to load products, you must restore the appropriate objects from file 1 with the **rbak** command prior to loading the products. The load procedure in Chapter 5 provides the appropriate **rbak** commands for doing this.

If any of the file 1 components get corrupted, removed, or overwritten in an Authorized Area, you can restore them from distribution media using an **rbak** command line similar to the one in "Loading the Installation Tools into an Authorized Area" in Chapter 9.

See Also

"Loading Products from Media into an Authorized Area," Chapter 5.
"Loading the Installation Tools into an Authorized Area," Chapter 9.

Installation Errors and Warnings

A description of some common error and warning messages displayed by the **install** program

This chapter describes some common error and warning messages displayed by the **install** program. **install** displays these messages when it is invoked directly and when it is invoked by the **install++** or **minst** tool. The first section describes error messages; the second, warning messages. In each section, the messages are arranged alphabetically.

We recommend that you always check the installation transcript (the set of messages displayed by the **install** program) for both warnings and errors. The **install** program prefixes all warnings with the label `WARNING:` and all errors with the label `ERROR:`. You can easily locate error and warnings messages by searching the transcript for these character strings. When errors occur, **install** completes execution with the message

```
RAI install has completed with errors.
```

instead of

```
RAI install has successfully completed.
```

If the installation transcript contains errors or warnings, you should correct any problems that exist and then, if necessary, rerun **install** (or **install++**). Errors and warnings often result from transient network problems during the installation. Such problems are usually corrected by running the installation again. Another run of **install** is usually much faster than the first because, by default, **install** copies only those objects that it did not successfully install during the first run.

If the transcript contains a message instructing you to shut down and reboot the node, do not reboot until after you resolve any error conditions. Instructions to reboot appear when you install operating system software or software that changes the system libraries.

Installation Error Messages

This section describes some common error messages displayed by the **install** program. The messages are arranged alphabetically.

ERROR: Cannot access authorized area on ...

The **install** program cannot access the Authorized Area specified on the command line. Check to make sure the pathname after the **-s** option on the **install** (or **install+ +**) command line is correct.

ERROR: Cannot install soft link ... ; already is a local copy

A product file or directory on the target node was manually changed from a soft (symbolic) link to a local copy, the configuration you are installing calls for installing that file or directory as a soft link, and you did not use the **install** (or **install+ +**) **-m** option (do not respect customization). To replace the local copy with a link as directed, rerun the installation with the **-m** option.

ERROR: Could not copy file ... to ...

For some reason, the **install** program cannot copy the named file from the Authorized Area to the target node. This is often caused by a temporary network problem or incorrect permissions set for the **install** tool. Check that the owner of **install** is **root** and the **setuid** bit for **install** is turned on. If necessary, change the permissions with **chmod** or **edacl.** Then reinstall. If the file 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. **-m** should cause the files not installed the first time to be copied to the target.

ERROR: Could not delete existing type ... for ...

The **install** program cannot 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 configure file: ...

The **install** program cannot find a configuration file specified on the command line. Check that the configuration file pathnames are correct.

ERROR: Could not find UID for ...

The named object is cataloged, but a system call invoked by the **install** program cannot locate it. The Authorized Area or the target node may need salvaging. Run **salvol** on the offending disk volume and reinstall.

ERROR: Could not get node UID for ...

The **install** program cannot locate a target node. Make sure the named node is cataloged. If not, try cataloging it with the **ctnode** command and reinstall.

ERROR: Could not hard link from ... to ...

The **install** program cannot create a hard link as required by a product's release index or by the **-l** option, so the installation is aborted. A missing link-to file or directory can cause this. Try reinstalling the product configuration.

ERROR: Could not preserve original ACLs for ...

The **install** program was replacing an object on the target node and encountered an error, such as system call failure, when it attempted to set the ACLs (Access Control Lists) on the object being installed to the ACLs of the object being replaced. When this happens, **install** preserves the object being replaced by renaming it **/install/rai_acl_temp.***number* (on the target node), where *number* is an arbitrary value. Change the name of the preserved file to its original pathname, overwriting the installed object, and try the installation again. Or, change the installed object's ACLs to match those of the original object, using the **chmod** or **edacl** command.

ERROR: Could not recover original copy of ...

The **install** program was replacing an object on the target node and encountered an error, such as system call failure, when it attempted to set the ACLs (Access Control Lists) on the object being installed to the ACLs of the object being replaced. When this happens, **install** preserves the object being replaced by renaming it **/install/rai_acl_temp.***number* (on the target node), where *number* is an arbitrary value. Change the name of the preserved file to its original pathname, overwriting the installed object, and try the installation again. Or, change the installed object's ACLs to match those of the original object, using the **chmod** or **edacl** command.

ERROR: Installation of ... to ... has been aborted.

The product configuration being installed includes a product whose ISP type (m68k or a88k) differs from that of the target node. By default, the **install** program does not install a product whose ISP type is different from the target, because the product may not run correctly or at all and, in some cases, may prevent the target from booting. To override the default behavior, rerun the install using the **–h** option.

ERROR: ... must be a local copy not a soft link

A product file or directory on the target node was manually changed from a local copy to a soft (symbolic) link, the configuration you are installing calls for installing that file or directory as a local copy, and you did not use the **install** (or **install++**) **–m** option (do not respect customization). To replace the link with a local copy, rerun the installation with the **–m** option.

ERROR: ... not found in authorized area

A product in the configuration being installed does not exist in the Authorized Area specified on the command line. Make sure you have specified the correct Authorized Area (the Authorized Area from which the configuration file was created). If you have, reload the missing product into the Authorized Area from media or create a new configuration file from the specified Authorized Area.

ERROR: Original ACLs could not be placed on ...

The **install** program was replacing an object on the target node and encountered an error, such as system call failure, when it attempted to set the ACLs (Access Control Lists) on the object being installed to the ACLs of the object being replaced. When this happens, **install** preserves the object being replaced by renaming it **/install/rai_acl_temp.***number* (on the target node), where *number* is an arbitrary value. Change the name of the preserved file to its original pathname, overwriting the installed object, and try the installation again. Or, change the installed object's ACLs to match those of the original object, using the **chmod** or **edacl** command.

Installation Warning Messages

This section describes some common warning messages displayed by the **install** program. The messages are arranged alphabetically.

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. The owner of **install** should be **root** and the **setuid** bit for **install** should be turned on. If this isn't the case, change the permissions with **chmod** or **edacl** and then reinstall. If the permissions are correct or the reinstall doesn't work after changing the permissions, 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** program cannot write a baseline file on the target node for some reason. This is typically caused by a transient network problem during the installation. Try running the install again. When **install** cannot create a baseline file, it is not prevented from successfully installing the product configuration. However, the target node will not have an accurate account of what is installed on the node. This may effect the **install** program's ability to respect product customization (the manual deletion of objects or manual change of copies to links or vice versa) when you subsequently install products on the node.

WARNING: Could not delete ...

The product configuration called for deleting an object on the target, but the **install** program cannot delete it. Either **install** is running with the wrong permissions (that is, it is not **setuid root**) or a temporary network problem prevented the action. Check that the owner of **install** is **root** and the **setuid** bit is turned on. If this isn't the case, change the permissions with **chmod** or **edacl.** Then reinstall.

WARNING: Invalid configuration for remote install

You invoked the **install++** tool with the **-r** (remote) option and did not specify a configuration file (with the **-c** or **-C** option) on the command line. This causes **install++** to ignore the **-r** option. To use **install++** with **-r**, you must specify a new or existing configuration file on the command line; **install++** cannot execute remotely and create a temporary configuration file.

WARNING: Remote process call failed ...

The **install** program could not start a remote process on a target node, so it ran the process locally. This typically happens when the Server Process Manager (**/sys/spm/spm**) is not running on the target node (a requirement for remote installations) or when a temporary network problem occurs during the installation.

WARNING: Target path file ... could not be found

The **install** program cannot find a target-list file specified on the command line with the **-n** option. Check that the target-list file's pathname is complete and correct.

WARNING: Target path file ... is not readable ...

The **install** program cannot read a target-list file specified on the command line with the **-n** option. Check that the target-list file is readable.

WARNING: ... would install through a link - item is ignored

The product configuration calls for installing an object at a pathname on the target node, and the path contains a directory that resolves to a soft (symbolic) link (a path **/a/b/c** , for example, in which **b** is a soft link). Presumably, the link was created manually after the object was installed on some earlier occasion. The **install** program will not install objects across links in this way, so it did not install the object. If the object must be installed, delete the link and reinstall.

a88k A designation for Domain machines that have a *PRISM* (Parallel Reduced Instruction Set Multiprocessor) ISP (Instruction Set Processor) type. Currently, the Series 10000 workstation is the only machine with an a88k ISP type. *See also* m68k.

Access Control List The set of access rights assigned to an object that determine who has access to the object and the type of access — read, write, or execute. Access Control Lists (or ACLs) are similar but not equivalent to UNIX permissions.

ACL *See* **Access Control List.**

active override file An override file whose configuration constraints have been put into effect by placing the file in the Authorized Area subdirectory **install/overrides** and naming it **ri.apollo.***product_name***.v.***version*.

Aegis One of the three environments (the other two are SysV and BSD) provided as part of the Domain/OS operating-system product. Aegis was developed initially by Apollo Computer as a UNIX-based proprietary operating system.

Authorized Area A directory, with an **install** subdirectory, that acts as a source area for network installations of Domain software products. In addition to containing products, an Authorized Area contains several administrative objects, such as the Domain installation tools.

baseline file A file created by the **install** program that records cumulatively the products that **install** installs on a node. Baseline files reside in the directory **/install/baseline** on the node on which the products are installed.

bldt A command that displays the version of the Domain/OS operating system that is running on the node, the date and time at which the operating system was built, and the node's network ID and node ID.

boot diskless To boot a node from another node's boot volume, rather than the node's own boot volume.

boot volume The logical volume on a disk or storage module, or the cartridge tapes or floppy disks, that contain the objects necessary to boot a node.

BSD One of the three environments (the other two are Aegis and SysV) provided as part of the Domain/OS operating-system product. BSD is based on 4.3 BSD UNIX developed by the University of California at Berkeley.

calendar An interactive program used to set 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 version resides in the **/com** directory.

cfgsa An interactive installation tool used to create selection and override files for a product. **cfgsa** resides in the Authorized Area subdirectory **install/tools**.

closed environment A Domain operating system environment, offered as a configuration choice for pre-SR10.4 versions of Domain/OS, in which the ACLs are set to provide relatively restricted access to the operating system objects.

cmpexe An object that can execute on both a88k machines (Series 10000 workstations) and m68k machines (workstation based on Motorola's 68000 series of microprocessors), or a product that can execute on both machine types. **cmpexe** is short for "compound executable."

compound product A product that can execute on both a88k machines (Series 10000 workstations) and m68k machines (workstation based on Motorola's 68000 series of microprocessors). compound products are created by merging the a88k and m68k versions of a product with the **mrgri** tool.

config An interactive installation tool used to create or modify a file, called a configuration file, that defines a configuration of one or more products to be installed on a node. **config** resides in the Authorized Area subdirectory **install/tools**. It can be invoked directly, or indirectly by the **install++** or **minst** tool.

configuration A set of software products and selected product options installed, or to be installed, on one or more nodes. For example, the 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 the **/sys/help**, **/domain_examples**, and **/usr/man** directories on another node in the network.

configuration file A file, created by the **config** program, that specifies one or more products to be installed on a node and answers to each product's configuration options. The options determine which objects belonging to a product are installed by the **install** or **install++** program and whether the objects are installed as local copies or as links to other nodes.

configure To select a set of products and specify answers to their configuration options prior to installing the products on a node. Products are configured with the **config** program, either invoked directly or indirectly by the **install++** or **minst** tool.

constrain To restrict the full set of configuration choices allowed for a product (the choices defined by the product's release index) by creating an override file with the **cfgsa** tool or activating an HP-supplied override file.

core tools The basic set of Domain installation tools — **cfgsa**, **config**, **distaa**, **install**, and **mrgri** — that do not invoke other installation tools. *See also* layered tools.

crp A command, pronounced "creep" and located in the directory **/com**, used to create a remote process on another node in the network.

ctnode A command, located in the **/com** and **/etc** directories, used to catalog a node in the network and assign a name (*//node_name*) to a node.

deinstall To remove an installed product using the **-A** switch of the **install** or **install++** tool.

diskless boot *See* **boot diskless**.

distaa A noninteractive installation tool that loads software products from distribution media into an Authorized Area. **distaa** resides in the Authorized Area subdirectory **install/tools**. It can be invoked directly, or indirectly by the **minst** tool.

distributed Authorized Area An Authorized Area in which products or parts of products are located on more than one disk or storage module through the use of soft links.

distribution media The set of cartridge tapes, floppy disks, or magnetic tapes on which you receive a release of one or more software products.

dmtvol An Aegis command, located in the directory **/com**, that dismounts a logical volume previously mounted with the **mtvol** command from a disk or storage module.

Domain/OS The operating system for Domain workstations. It is released with three operating environments (Aegis, SysV, and BSD), all of which or some combination of which can be installed. Some Domain/OS objects are common to all three environments while others are environment-specific. Domain/OS releases are numbered SR*x* or SR*x.y*. The Domain/OS product name recognized by the installation tools is **os**.

Domain Server Processor A processor unit without a monitor or keyboard.

DSP *See* **Domain Server Processor.**

excludes.list file A user-created ASCII file in an Authorized Area's **install/overrides** subdirectory that lists files in the Authorized Area to be excluded from installation on target nodes.

expert mode A mode of operation of the **minst** installation tool in which the user is given greater control over the loading and installation of products and must respond to more questions. *See also* novice mode.

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). Creating a hard link increments the file's reference count to 2. *See also* soft link.

install A noninteractive installation tool that installs products on one or more nodes. The products to be installed and their configuration options are specified by a previously created configuration file, which is supplied as a command-line argument to **install. install** resides in the Authorized Area subdirectory **install/tools**. It can be invoked directly, or indirectly by the **install++** tool.

install++ An installation tool that configures and installs products on one or more nodes. It optionally invokes the **config** program to enable users to interactively configure products and then invokes the **install** program to install products. **install++** resides in the Authorized Area subdirectory **install/tools**.

install checking The type of checking (set by the **config** program) that the **install** program performs on installed objects, when you reinstall or update a product, to determine whether it should replace the objects.

install directory The requisite subdirectory of an Authorized Area (*aa_name*/**install**) or the **/install** directory created on all nodes by the **install** program. The latter directory contains: **baseline** files that record the products installed on the node, the release documentation of products installed on the node, and other objects used and/or created by the **install** program to control the installation of products on the node.

invol An interactive program used to initialize a disk or storage module. An offline version (accessed at the Mnemonic Debugger level) resides in each of the **/sau** directories; an online version resides in the **/com** and **/etc** directories.

ISP Type The type of Instruction Set Processor that a Domain machine uses — currently either one based on Motorola's 68000 series of microprocessors (m68k ISP type) or on HP's *PRISM* processor (a88k ISP type).

layered tools The installation tools — **install++** and **minst** — that invoke other installation tools. *See also* core tools.

link A special type of object that points from one place in the network's naming tree to another. *See also* hard link; soft link.

load To restore products from distribution media into an Authorized Area, using the **distaa** or **minst** tool.

m68k A designation for Domain machines that have an ISP (Instruction Set Processor) type based on Motorola's 68000 series of microprocessors. *See also* a88k,

merge To merge two products in an Authorized Area and their release indexes into a single product and single release index, using the **mrgri** tool.

minst An interactive, self-guiding installation tool that loads products from distribution media (by invoking **rbak** and the **distaa** tool) and then optionally installs the products (by invoking the **install++** tool). **minst** resides in the Authorized Area subdirectory **install/tools**.

Mnemonic Debugger A program that resides in each node's boot PROM. It provides a user interface and a skeletal set of commands so you can communicate with the node prior to installation of the operating system and when the node is shut down.

mount A UNIX command, located in the **/etc** directory, that mounts a removable or remote file system on a a disk or storage module, making the files and directories it contains accessible.

mrgri An installation tool used to merge two products in an Authorized Area and their release indexes into a single product and single release index.

mtvol An Aegis command, located in the directory **/com**, that mounts a logical volume on a disk or storage module, making the files and directories it contains accessible.

node entry directory The top–level directory on a node (*//node_name*); a subdirectory of the network root directory.

node ID A unique, unchangeable, hexadecimal ID assigned to each node during the manufacturing process.

not_installed file A file created by the **install** program in a node's **/install** directory that lists files it could not install on that node, due to some problem.

novice mode A mode of operation of the **minst** installation tool in which **minst** supplies default answers to several questions that would otherwise be asked of the user. *See also* expert mode.

open A Domain operating system environment, offered as a configuration choice for pre-SR10.4 versions of Domain/OS, in which the ACLs are set to allow relatively unrestricted access to the operating system objects. Specifically, for most objects, **p** rights are granted to the **world**, meaning any user can change the object's initial ACLs.

optional product A product that can be optionally purchased separate from the operating system, and that requires the operating system to run. Examples include Pascal, FORTRAN, and DSEE.

override file A file that restricts or *overrides* the the full set of configuration choices allowed for a product (the set of configuration choices defined by the product's release index).

partner or partner node A node that provides the Domain/OS operating system for a node that is booted diskless from it.

patch In the Domain installation model, a file or set of files released as an individual product that corrects some problem in Domain/OS or an optional product.

preserve.list file A user-created ASCII file in a node's **/install** directory that lists files not to be overwritten by the **install** program when products are updated on that node.

product In the Domain installation model, a set of objects assigned a name and a version number recognizable by the installation tools, and with a product release index. In practice, it helps to separate products into four classes: the Domain/OS operating system, PSKs, optional products, and patches.

product directory A directory on distribution media or in an Authorized Area that contains the objects that comprise a particular product and the product's release index. In an Authorized Area, product directories have pathnames of the form **install/ri.apollo.***product_name***.v.***version*.

product release index A binary file released with every product that is used internally by the installation tools to control and build a viable, operational configuration of the product on a node. Among other things, the file specifies the product's configuration options and questions, and the product's dependencies on other products.

product subset A subset of product components that can be loaded and installed as a viable, operational configuration (for example, an **aegis_small** configuration of Domain/OS). Products subsets are defined by selection files, which can be created by the user with **cfgsa** or that ship with the product.

Product Support Kit *See* **PSK.**

PSK A subset of the Domain/OS operating system, released as a separate product, that provides support for a new hardware platform or peripheral device, or that provides new functionality. **PSK** stands for Product Support Kit.

rai_acl_temp file A file created by the **install** program when it cannot preserve the ACLs of an object that it is overwriting. The file is a renamed version of the overwritten object, which you can check to determine what the ACLs of the newly installed object should be set to. **rai_acl_temp** files are created in the **/install** directory of the node on which the ACLs could not be preserved.

rbak A command used to restore objects from backup media written by the **wbak** (write backup) command. **rbak** resides in the **/com** and **/usr/apollo/bin** directories.

release index *See* **product release index.**

remote installation An installation of products in which the **install** processes execute remotely on the target nodes, rather than locally on the node on which **install** is invoked. Remote installations are performed by using the **–r** switch with the **install** tool.

salacl A command that salvages Access Control Lists (ACLs) on a specified volume. It merges duplicate ACLs into a single copy and deletes unused ACLs. **salacl** resides in the **/com** and **/etc** directories.

/sau*x* directory A directory (where *x* is an integer) that contains a set of machine–specific utilities and programs. Each node family has its own /sau*x* directory.

selection file A file used with the **distaa** tool to select a particular product or a subset of a product for loading from distribution media into an Authorized Area.

shut A Display Manager command used to kill all processes and shut down a node to the Mnemonic Debugger level.

shutdown A UNIX command, located in the **/etc** directory, used to shut down Domain nodes to the Mnemonic Debugger level or the Phase II boot shell.

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**. Soft links are also called symbolic links. *See also* hard link.

symbolic link *See* **soft link.**

SysV One of the three environments (the other two are Aegis and BSD) provided as part of the Domain/OS operating-system product. SysV is based on UNIX System V developed by AT&T.

target (of an initialization) The disk or storage module that is being initialized; or the workstation or DSP to which that disk or storage module is connected.

target (of an installation) The directory into which you're installing products. The target is typically a node's entry directory (*//node_name*), in which case the term **target node** is often used, or the mount point of a mounted volume.

templates A term sometimes used to refer collectively to the HP-supplied selection and override files that ship with a product. These files reside in the Authorized Area subdirectory **install/templates**.

TOC file An ASCII file released on each set of distribution media that identifies the media and lists all products and product subcomponents on the media. TOC files are restored to the Authorized Area subdirectory **install/toc**.

uctnode A command, located in the **/com** directory, that uncatalogs a node. **uctnode** removes a node's entry directory name from its local copy of the network root directory and optionally from the network root directory.

umount A UNIX command, located in the **/etc** directory, that unmounts a previously mounted file system from a disk or storage module.

wbak A command used to write objects to removable backup media. **wbak** resides in the **/com** and **/usr/apollo/bin** directories.

Index