Data General

Customer Documentation

Customizing the DG/UX™ System

AViiON®
PRODUCT LINE

Customizing the DG/UX™ System

093-701101-00

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice (085-series) supplied with the software.

Ordering No. 093-701101

Printed in the United States of America
Revision 00, June 1991

NOTICE

Customizing the DG/UX™ System

093-701101-00

Revision History:	Effective with:
Original Release - June 1991	DG/UX Release 5.4

Preface

This manual, used immediately following installation and any time thereafter, contains a subset of the most commonly performed system administration tasks. If used immediately after installation, this manual gives the procedures necessary for getting your system to an operational state. Browsing through the table of contents, you can identify those operations that apply to you. For example, if you have just completed installation of an OS client-server system using the procedures in *Installing the DG/UX™ System*, at a minimum, you will turn to this manual for procedures to create more logical disks to accommodate user home directories and the OS client release area, to add user accounts, to add OS clients, and to build and boot OS client kernels. As another example, if you have a number of physical disk devices attached to each OS client computer, you will use procedures to correctly prepare physical disks, create logical disks, and create and mount file systems on the physical disks. Consequently, you may need to build and boot a new kernel.

Also, it can be used at any time following installation to alter the behavior of the operating system (for example, to add an OS client, device, printer, software package, to name few).

The information in this manual is presented in a practical fashion so that you can perform the task quickly and efficiently. For a thorough treatment of all system administration duties and underlying concepts, see *Managing the DG/UX™ System.*

Although the procedures are explained thoroughly, the manual assumes you have experience as a system administrator of some kind of operating system. While knowledge of UNIX® is not required, it is helpful if you know:

- the general file system layout of the UNIX operating system
- how to use UNIX commands
- how to use a shell and work within the UNIX directory structure

Refer to *Using the DG/UX™ System* for background information on these topics.

Manual Outline

This manual is composed of the following chapters and appendixes:

Chapter 1 **Using sysadm to Customize the DG/UX System**
Introduces the System Administration (**sysadm**) utilities that you use to customize your DG/UX system. Logged in as **sysadm** or **root**, you can use any of three user interfaces to the utility: OSF/Motif pull-down menus, traditional ASCII menus, or shell commands.

Chapter 2 **Planning System Resources**
Summarizes important concepts that are relevant to continued allocation of system resources following installation. Helps you identify your physical disk and memory resources, your logical disk requirements for software and work space, an effective arrangement of logical disks on physical disks, and appropriate mount points for the file systems on the DG/UX system.

Chapter 3 **Creating Logical Disks and File Systems**
With plans made in the previous chapter, you use these procedures to create the logical disks (using a series of logical disk planning worksheets) and mount local and remote file systems.

Chapter 4 **Adding User Accounts**
Gives instructions for creating accounts (home directories) for each user on the DG/UX system.

Chapter 5 **Adding Terminals and Modems**
Gives procedures for setting up terminals and modems to operate in the DG/UX environment.

Chapter 6 **Adding Printers**
Gives procedures for configuring a local and a remote printer, setting up a printer filter (which processes files before printing them), and establishing a default printer.

Chapter 7 **Loading and Setting Up Software Packages**
Instructs on transferring the software (obtained from either Data General or a third-party vendor) from a release tape onto disk and configuring the software to run with the DG/UX system.

Chapter 8 **Adding Clients**
Helps link clients (OS and X terminals) to OS servers which supply OS clients with a bootable operating system (primary and secondary) and file system space via a local area network. An X terminal relies on an OS server for its bootstrap. In addition, cookbook instructions are given for setting up an OS client with an attached physical disk, which can provide its own **root** and **swap** logical disk resources. Also, instructions are given for adding an OS client that provides its own local **swap** logical disk resources.

Chapter 9	**Adding Secondary Operating System Releases** Creates a secondary release area for subsequently loading and setting up a second operating system such as a pre-5.4 release of DG/UX or a foreign operating system. Also, gives cookbook procedures for installing the 5.4 release of DG/UX as a secondary release to a pre-5.4 DG/UX system such as 4.32.
Chapter 10	**Adding Physical Devices** Gives instructions for adding physical devices (such as Winchester disks, CD-ROMs, magneto-optical drives, diskette drives, and tape drives). Outlines the formatting sequence that is required for disk devices to accommodate a variety of file types.
Chapter 11	**Building a Kernel** Identifies the circumstances under which you need to rebuild a new kernel (the basis of the DG/UX operating system) to reflect your hardware and software configuration. Highlights the types of kernel building procedures and describes the system configuration file which underlies the kernel.
Chapter 12	**Booting and Logging in to the DG/UX System** Gives different methods for booting the system required to gain access to a newly built kernel and to reinitialize the system following a system panic or failure. Also, describes how to log in to the system following a reboot.
Appendix A	**Editing with vi** Summarizes the basic **vi** (UNIX visual editor) editing commands.
Appendix B	**Displaying the Layout of Your Physical Disks** Explains a **diskman** option that you can use for viewing the arrangement of logical disks on physical disk devices in your hardware configuration.
Appendix C	**Determining tty Lines for Asynchronous Ports** Discusses how the DG/UX system assigns tty lines, gives instructions for using the AViiON System Diagnostics to obtain information about terminal line controllers, and supplies worksheets to help record your tty line configuration.
Appendix D	**Device Naming** Explains in detail the DG/UX common device specification format used to identify standard and nonstandard devices in your hardware configuration. Provides the short and long format used for internal system representation and the device nodes that are created in the **/dev** directory each time you boot your system.
Appendix E	**Planning Worksheets** Is a repository for information you gather for logical disks, printers, clients and OS servers, and tty lines.

Readers, Please Note

Data General manuals use certain symbols and styles of type to indicate different meanings. You should familiarize yourself with the following conventions before reading the manual.

Convention	Meaning
boldface	In command lines and format lines: Indicates text (including punctuation) that you type verbatim from your keyboard. All DG/UX commands, path names, and names of files, directories, and manual pages also use this typeface.
`constant width/ monospace`	Represents a system response on your screen. Syntax lines also use this font.
italic	In format lines: Represents variables for which you supply values; for example, the names of your directories and files, your username and password, and possible arguments to commands.
[*optional*]	In format lines: These brackets surround an optional argument. Don't type the brackets; they only set off what is optional. The brackets are in regular type and should not be confused with the boldface brackets shown below.
[ ]	In format lines: Indicates literal brackets that you should type. These brackets are in boldface type and should not be confused with the regular type brackets shown above.
...	In format lines and syntax lines: Means you can repeat the preceding argument as many times as desired.
`$`, `%` and `#`	In command lines and other examples: The `$` represents the system command prompt symbols used for the Bourne and Korn shell; the `%` is used for the C shell; and `#` is used for the superuser prompt for all three shells.
↵	In command lines and other examples: Represents the New Line key. Note that on some keyboards this key might be called Enter or Return instead of New Line.
< >	In command lines and other examples: Angle brackets distinguish a command sequence or a keystroke (such as **<Ctrl-D>**, **<Esc>**, and **<3dw>**) from surrounding text.

Contacting Data General

Data General wants to assist you in any way it can to help you use its products. Please feel free to contact the company as outlined below.

Manuals

If you require additional manuals, please use the enclosed TIPS order form (United States only) or contact your local Data General sales representative. A list of related documents appears at the end of this manual with the TIPS order form.

For a complete list of AViiON® and DG/UX™ manuals, see the *Guide to AViiON® and DG/UX™ System Documentation* (069-701085). The on-line version of this manual found in **/usr/release/doc_guide** contains the most current list.

Telephone Assistance

If you are unable to solve a problem using any manual you received with your system, free telephone assistance is available with your hardware warranty and with most Data General software service options. If you are within the United States or Canada, contact the Data General Service Center by calling 1-800-DG-HELPS. Lines are open from 8:00 a.m. to 5:00 p.m., your time, Monday through Friday. The center will put you in touch with a member of Data General's telephone assistance staff who can answer your questions.

For telephone assistance outside the United States or Canada, ask your Data General sales representative for the appropriate telephone number.

Joining Our Users Group

Please consider joining the largest independent organization of Data General users, the North American Data General Users Group (NADGUG). In addition to making valuable contacts, members receive FOCUS monthly magazine, a conference discount, access to the Software Library and Electronic Bulletin Board, an annual Member Directory, Regional and Special Interest Groups, and much more. For more information about membership in the North American Data General Users Group, call 1-800-877-4787 or 1-512-345-5316.

End of Preface

Contents

Chapter 1 — Using Sysadm to Customize the DG/UX System

Chapter 2 — Disk Planning

Chapter 3 — Creating Logical Disks and File Systems

Chapter 4 — Adding User Accounts

Chapter 5 — Adding Terminals and Modems

Chapter 6 — Adding Printers

Chapter 7 — Loading and Setting Up Software Packages

Chapter 8 — Adding Clients

Chapter 9 — Adding Secondary Operating System Releases

Chapter 10 — Adding Physical Devices

Chapter 11 — Building Kernels

Chapter 12 — Booting and Logging In to the DG/UX System

Index

Related Documents

Tables

Figures

Chapter 1
Using Sysadm to Customize the DG/UX System

Before you start to use the System Administration (**sysadm**) utilities, it is assumed that:

- You have correctly installed the hardware elements of your DG/UX system using the appropriate hardware documentation.
- You have installed your DG/UX system using the instructions in *Installing the DG/UX™ System*. If you have not, do so now before proceeding.
- As the system administrator, you have assigned passwords to **root** and **sysadm**.

Assuming these conditions to be true, you are ready to use **sysadm** to customize your DG/UX system.

The DG/UX system provides the **sysadm** utilities, a menu-based interface, that you can use to set up and manage the hardware and software elements of your configuration. Selecting from a set of commonly performed procedures, you will customize your DG/UX environment to suit your particular needs.

Logged in as **sysadm** or **root**, you can access **sysadm** to customize your DG/UX system. Three interfaces are available:

- OSF/Motif interface
- ASCII terminal menu interface
- Shell commands

Using sysadm with the OSF/Motif Window Manager

To invoke the OSF/Motif version of **sysadm**, execute **xsysadm** from the command line, shown as follows:

```
# xsysadm ↵
```

NOTE: Or, assuming that the X Window System is installed and running, you can type only **sysadm** to access the X-based version of **sysadm**.

You must be using the OSF/Motif window manager, **mwm**. You may append the usual X Window System client arguments to the command line as you wish. See the OSF/Motif Edition of the *X Window System™ User's Guide*. When you are not using **sysadm**, you can iconify it by using the mouse to click on the dot symbol near the upper right corner of the window. If you use **sysadm** often, you may find it easier to iconify **sysadm** when not needed rather than terminating it and having to restart it when needed again.

The selections in the top level **sysadm** menu reflect the software installed on your system. If other packages are present, your **sysadm** menus may look different.

Figure 1-1 shows the top level **sysadm** menu as it appears in the OSF/Motif interface.

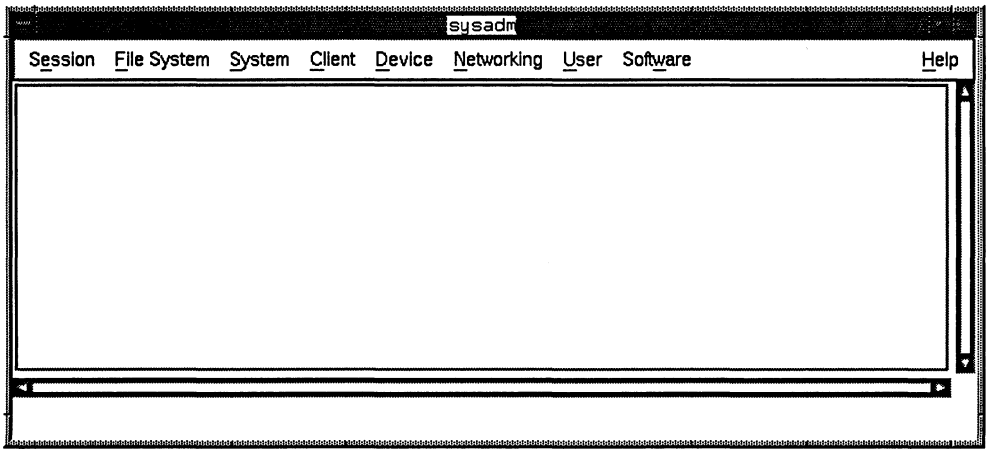


Figure 1-1 *Top Level sysadm Menu in the OSF/Motif Interface*

Reading the Display

The **sysadm** utility's OSF/Motif interface conforms to the conventions described in the OSF/Motif Edition of the *X Window System™ User's Guide*. The main, or top level, menu displays sub-menus in a menu bar at the top of the window. A symbol next to a menu selection tells you something about that selection. A right-pointing arrow symbol means that the selection is a menu. An ellipsis (. . .) means that the selection is an operation that requires information or confirmation from you before proceeding. Menu selections with no symbol next to them are operations that require no more response from you before proceeding.

If an operation name is in a dim or gray font, you may not perform the operation. This restriction occurs when you do not have access to the operation or when your system configuration makes the operation unnecessary or impossible. Typically, you want to invoke **sysadm** as the superuser; otherwise, **sysadm** allows you to do little more than list information about the system.

Navigating Menus

There are several ways to navigate the menu structure. Using the mouse, move the pointer to the desired menu and *select* the menu by clicking the mouse's select button (typically the left button). You continue this way, using the mouse to select the desired menus until you reach an operation you want to perform. With the mouse, select the operation to start it.

Optionally, you may drag the mouse across menus, which means you press and hold the select button while moving the mouse across the desired selections. Releasing the select button when an operation is highlighted begins the operation.

You may also navigate using the keyboard. To navigate using the keyboard, the mouse pointer must be inside the **sysadm** main menu window. Select a menu from the top level menu bar by pressing the <Alt> key while pressing the desired menu's key letter. The key letter is a letter in the title, usually the first letter, that is highlighted with an underscore. For example, the key letter of the Client menu is **C**, so press <Alt-C> to display the Client menu.

After selecting a top level menu, select a sub-menu by pressing the menu's key letter, this time without using the Alt key. For example, select the Client menu with <Alt-C>. Then select the OS Client menu by pressing **O**.

The other way to navigate is with the arrow keys. Your position in the menu tree is marked by a box around the current selection. Press the arrow keys to move this box from selection to selection. When you arrive at the desired operation, press New Line or the space bar to begin.

Answering Queries

The **sysadm** utility displays a form when it needs more information to complete an operation. A form contains queries, which you may answer in any order. Some queries appear next to a small square, a button, which you click to alternate between "yes" (pressed in and shaded dark) and "no" (out and shaded light). Some queries require text input. To type in text, first select the box (making it the input focus) by clicking in the box or by pressing the Tab key until the box appears highlighted. When the box is highlighted, you can type in text. Do not press New Line after typing in a text input box; pressing New Line in a form is the same as selecting the **Next** or **OK** button at the bottom of the form.

Some queries allow you to select one or more values from a list. Select entries in the list by clicking on them. If the list is too long to display completely on the screen, the operation displays the first part of the list in a box that has a scroll bar along on the right side. To scroll through the list, use your mouse to drag the scroll bar up and down. When you see the value that you want in the list, click on it. In some queries, you enter a numerical value by typing a response in a box or by using your mouse to drag an indicator along a horizontal scale.

After answering the form's queries, proceed by pressing New Line or by selecting **OK** or **Next**. If you choose not to perform the operation, select **Cancel**. Select **Reset** to restore the default responses.

Getting Help

The **sysadm** utility provides on-line help in several ways. At the top level menu, the Help menu offers this information:

On Sysadm
Select this option for general information on the **sysadm** utility.

On Interface
Select this option for information on the interface you are using (OSF/Motif interface or ASCII character display).

On Version
Select this option for information on the current version of the **sysadm** utility.

On Help
Select this option for information on obtaining help throughout **sysadm**.

Every menu and operation in **sysadm** has a help message in addition to the description line that you see at the bottom of the **sysadm** main menu window. To read the help message, use your arrow keys, as described in the "Navigating Menus" section above, to highlight the desired menu or operation. Then press F1 (function key 1) to see the help message.

To get help about an operation already in progress, select **Help** at the bottom right of the form.

To get help about a specific query within a form, click on the query so that a dark border appears around it. As an alternative, you may press the Tab key on your keyboard until the dark border appears around the desired query. Once you have selected a query like this, you may get help for it by pressing F1. Pressing F1 without having selected a query displays help

for the operation.

In general, the query that has the dark border (said to have the input focus) is the one to which you can type using your keyboard. Press the Tab key to move input focus from query to query.

Using sysadm on an ASCII Terminal

You may use the character interface to **sysadm** on a normal ASCII terminal or in an **xterm** or **mterm** window. The character interface, rather than presenting windows on the screen and depending on a mouse for input, uses a common scrolling character display. Invoke the character interface with this command line:

```
# asysadm ↵
```

NOTE: Or, assuming that the X Window System is not installed and running, you can type only **sysadm** to access the ASCII-based version of **sysadm**.

Figure 1-2 shows the top level **sysadm** menu as it appears in the ASCII interface.

```
                Main Menu

  1  Session ->            Manage this sysadm session
  2  File System ->        Manage file systems
  3  System ->             Manage DG/UX system databases
  4  Client ->             Manage OS and X terminal clients
  5  Device ->             Manage devices and device queues
  6  Networking ->         Manage network databases
  7  User ->               Manage users and groups
  8  Software ->           Manage software packages
  9  Help ->               Get help on sysadm and its queries
```

Figure 1-2 *Top Level sysadm Menu in the ASCII Interface*

Reading the Display

The character interface to **sysadm** works by presenting you with menus representing classes of objects that you can manage. A menu may contain additional menus or operations that you can perform on the objects. A description of each menu or operation appears next to the menu or operation name.

There are three types of menu selections identified by special symbols after the menu name. The symbol -> indicates that the menu contains a sub-menu. The symbol ... indicates that the selection is an operation on an object, and making this selection will start a series of questions that you must answer before the operation can continue. Finally, if there is no symbol, the selection is an operation that can proceed without any further information from you.

If an operation appears surrounded by square brackets, you do not have sufficient access to perform the operation. Many operations are disabled in this way when you invoke **sysadm** without superuser privilege.

Navigating Menus

Navigate the menus by typing the menu selection number of your choice, then pressing the New Line key. You may also make selections by typing the name of the choice (or as many letters as are necessary to make your selection unambiguous).

To return to the preceding menu, press the caret key (^). You may exit **sysadm** by pressing the **q** key (you will be asked if you really meant to do that before **sysadm** exits). Always follow your entries with the New Line key.

When text to be displayed cannot fit on one screen, **sysadm** presents it one screen at a time. Press New Line to continue, or use the **?** key to get information about what else you can do. For example, using the **p** key redisplays the previous page of output. The pager allows searching as well as explicit positioning to any page. Press the **n** key to quit the presentation.

Table 1-1 presents a summary of the methods for making ASCII **sysadm** menu choices.

Table 1-1 Making ASCII sysadm Menu Choices

User Input	Description
number	Choose menu item by entering number.
name	Choose menu item by entering full name of menu item, such as **Session**, or a string fragment that uniquely identifies the menu item such as **Ses** or **ses** for **Session**. The string is not case-sensitive.
colon-separated-names	To specify menu traversals, you can specify a colon-separated list of names. For example, **Software:Package:Install** or **So:Pack:In**.
?	Print help message, then redisplay menu prompt.
*number***?**	Print help message for a particular menu item, then redisplay menu prompt.
q	Exit **sysadm**. Enter from any menu.
New Line	Redisplay menu.
^ or **..**	Return to the next higher menu.

Answering Queries

When an operation presents a query, a default response is often displayed within brackets. For example, [yes] indicates that an affirmative response is assumed if you press the New Line key.

When only a predetermined set of responses is appropriate, use the ? key to display all your choices. You may then select your choice by number or by the text of the choice itself (as many characters as are necessary to identify your choice unambiguously).

After you select an operation and enter any information that it requires, the operation asks for confirmation before it executes.

Getting Help

The **sysadm** utility provides on-line help in several ways. At the top level menu, the Help menu offers this information:

On Sysadm
Select this option for general information on the **sysadm** utility.

On Interface
Select this option for information on the interface you are using (OSF/Motif interface or ASCII character display).

On Version
Select this option for information on the current version of the **sysadm** utility.

On Help
Select this option for information on obtaining help throughout **sysadm**.

Every menu and operation in **sysadm** has a help message in addition to the description line that you see in the menu display. Enter ? to get help about the current menu, or enter a menu selection number followed by the ? key to get information about a particular selection. You may also use ? to get help and syntax information in any query.

Using the Shell Command Line to Bypass Menus

Bypassing the menu interfaces altogether, you can invoke **sysadm** directly from the command line, supplying the menu selections in this form:

sysadm **-m** *menu-name(s)*

where:

-m selects the menu name.

You specify menu names using colon-separated lists of strings (See Table 1-1 for definitions.)

Using the shell is a convenient shortcut if you are already familiar with **sysadm.**

Examples follow:

 # sysadm -m File:Local ↵

 # sysadm -m F:L ↵

Both commands perform the same operation; they start **sysadm** at the Local Filesys menu. The second example shows the operations represented as unambiguous string fragments.

Convention Used for Showing sysadm Menu Choices in this Manual

Since you have three methods available for navigating **sysadm** to select operations, a convention is adopted in this manual for showing a generic pathway that you follow through the menus regardless of the interface you use. An example of the convention used to show building an automatically configured kernel, starting at the Sysadm Main Menu, follows:

```
System -> Kernel -> Auto Configure
```

The strings, such as `System` and `Kernel`, are unique menu items. The arrow (`->`) symbolizes the traversal through the menus. Examples of their implementations follow:

Figure 1-3 shows the Kernel Menu as it appears in the OSF/Motif interface:

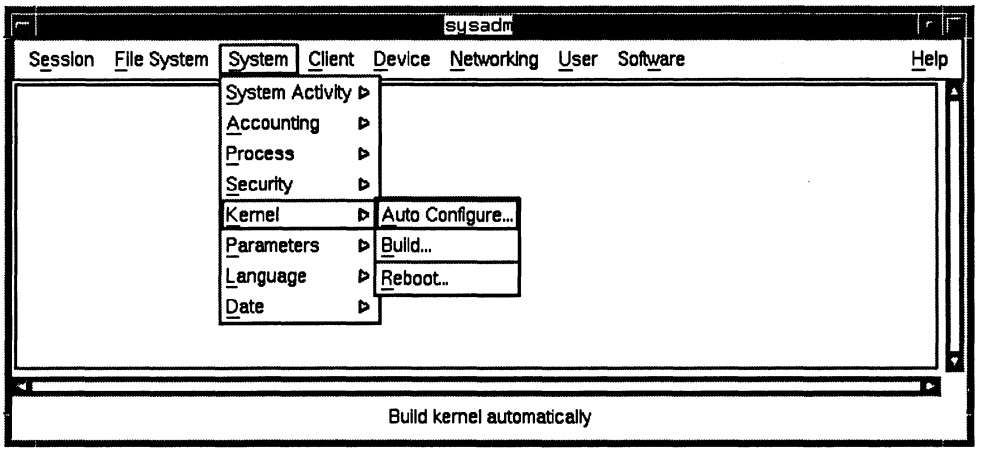


Figure 1-3 *OSF/Motif Kernel Menu*

Figure 1-4 shows the Kernel Menu as it appears in the ASCII terminal menu interface.

```
                Kernel Menu

   1  Auto Configure ...    Build kernel automatically
   2  Build ...             Build custom kernel

   3  Reboot ...            Reboot kernel

Enter a number, a name, ? or <number>? for help, ^ to return to the
previous menu, or q to quit:
```

Figure 1-4 *ASCII Interface Kernel Menu*

Using the OSF/Motif interface to select the Auto Configure operation from the Kernel Menu, click on these choices:

```
System -> Kernel -> Auto Configure
```

Using the ASCII terminal menu interface to select the Auto Configure operation from the Kernel Menu, supply an entry to the following prompt:

```
Enter a number, a name, ? or <number>? for help, or q to quit: System:Kernel:Auto ↵
```

Alternatively, you could choose to enter the desired string or number at each menu level.

```
# sysadm -m System:Kernel:Auto ↵
```

For all three choices, at the conclusion of the menu traversal, you will answer prompts to actually perform the operation.

Using Manual Pages

This manual makes frequent references to "manual pages" which provide a complete description of all DG/UX commands. A manual page is an on-line document that contains a technical description of all attributes for each command, system call, or special file, collectively referred to as utilities, in the system. There are corresponding manual pages that are printed in the following manuals:

User's Reference for the DG/UX™ System
Programmer's Reference for the DG/UX™ System (Volumes 1, 2, and 3)
System Manager's Reference for the DG/UX™ System

Each of these manuals covers specific categories of commands, which are identified by a number in parentheses. Table 1-2 gives each numeric category, the command type, and the manual in which it is located.

Table 1-2 Finding Manual Pages in the Reference Manuals

Number	Command Type	Manual Title
(1)	User commands and application programs	*User's Reference for the DG/UX™ System*
(5)	Miscellaneous features	*User's Reference for the DG/UX™ System*
(1) and (2)	Programmer commands and system calls	*Programmer's Reference for the DG/UX™ System (Volume 1)*
(3C), (3E), (3G), (3M), (3N), (3R), (3S), (3W), (3X)	C programming libraries, general library routines, mathematical library routines, networking support utilities, remote procedure call routines, standard I/O library routines, multinational language set routines, specialized libraries	*Programmer's Reference for the DG/UX™ System (Volume 2)*
(4), (5), and (6)	File format, miscellany, and communications protocol	Programmer's Reference for the *DG/UX™ System (Volume 3)*
(1M) and (4M)	Administrative commands and application programs, and file formats for administrative files	*System Manager's Reference for the DG/UX™ System*
(7)	System special files	*System Manager's Reference for the DG/UX™ System*
(8)	System maintenance procedures	*System Manager's Reference for the DG/UX™ System*

The parenthetical number following each command identifies its location in the reference manual.

You can access an on-line man page using the **man** command. For example, to see the manual page for the **ls** command, enter the following.

```
# man ls ↵
```

Many manual pages are too long to fit on one screen, so they scroll quickly up and off the screen. To insert a pause between screens, you can use the **more** command and the shell's pipe operator (|) along with the **man** command to control the output display. For example:

```
# man ls | more ↵
```

You enter DG/UX commands in lowercase characters; a command in uppercase letters is unrecognizable to the DG/UX system.

At the bottom of the first screen, this message is displayed:

```
-- More --
```

Press the space bar to display the next screen.

Using the Planning Worksheets

Worksheets are provided in Appendix E to help you customize (plan logical disks, local and remote printers, OS server and client networking, and tty lines) your DG/UX system. Making decisions ahead of time speeds up the customization process considerably. Each chapter, where appropriate, instructs you on completing a particular worksheet in preparation for a particular customization task.

Returning to the Shell

During a **sysadm** session, you can easily return to the shell (Bourne, C, or Korn). How you return depends on whether you are using the OSF/Motif-based **sysadm** or the ASCII terminal **sysadm.**

From OSF/Motif-based **sysadm**, your shell runs as a background job and occupies a separate window. To return to the shell, simply move the mouse back to the shell prompt in the shell window. To return to **sysadm**, move the cursor back to the **sysadm** window.

From ASCII-based **sysadm** prompt, escape to the shell by executing the appropriate shell escape command. Examples for the C shell, Bourne shell, and Korn shell follow:

```
# !csh ↵
# !sh ↵
# !ksh ↵
```

From the shell, to return to ASCII **sysadm**, type:

```
# exit ↵
```

End of Chapter

Chapter 2
Disk Planning

This chapter helps you to plan your remaining disk resources. Using procedures described in *Installing the DG/UX™ System*, you have already allocated resources for the DG/UX system contained on logical disks named **swap**, **root**, and **usr**. In addition, you may have created logical disks named **usr_opt_X11** and **usr_opt_aview**, among others.

The following sections about disk planning assume that you are familiar with these terms:

physical disk	Hardware medium used for storing data whose capacity, measured in megabytes (Mbytes), is identified by a model number.
logical disk	Reserved area, or subdivision, (measured in 512-byte blocks) on one or more physical disks onto which data can be loaded. Logical disks are uniquely named using DG/UX file-naming conventions: 31 characters maximum, using a combination of alphabetic characters, numbers, the period (.), the comma (,), the hyphen (-), and the underscore (_).
logical disk pieces	The individual units which are assembled to comprise a logical disk. The units are composed of 512-byte blocks. Pieces do not have to be contiguously arranged on a physical disk; they can span multiple physical disks. A logical disk contains at least one logical disk piece.
file system	Organization of the logical disk space into a hierarchical structure that contains data.
DG/UX directory structure	Arrangement of hierarchically structured file systems that reside on logical disks. The DG/UX directory structure emanates from the **/** (root) directory.
mount point	Directory location within the DG/UX directory structure at which a file system that resides on a logical disk is placed.

Before you start disk planning, you need to answer these basic questions:

- How do I intend to use logical disks? That is, what software packages do I have and how much space do they need? What other special work areas do I need?
- What physical devices do I have and how much space do they provide?
- How do I map software to logical disks?
- How do I map logical disks to physical disks?
- Where should I put the file systems on logical disks in the DG/UX directory structure?

Appendix E provides a series of logical disk planning worksheets to help you customize your system. These worksheets contain tables that you complete to identify the names of logical disks, sizes, possible arrangements in pieces, and mount points in the DG/UX directory structure. Time spent in preparation speeds the customization process.

If you have experience customizing the DG/UX system or other operating systems, you may prefer to go directly to the worksheets in Appendix E.

Planning How to Use Logical Disks

The first thing you must do is determine how you intend to use logical disks. Typical purposes for logical disks are as follows:

- OS client space
- Multiple OS releases
- Miscellany
  - — User home directories
  - — Software packages
  - — Work directories
  - — Tools directories
  - — Temporary space
  - — Extra swap space

The following sections will help you determine how to use logical disks.

An example of a logical disk name is provided for each type of logical disk type. You may select any logical name that you want, however. Name each logical disk using DG/UX file-naming conventions.

OS Client Space

If you have an OS client-server configuration, you must create the following logical disks on the OS server:

- OS client directory (**srv**)
- OS client root space (**srv_root**)
- OS client swap space (**srv_swap**)
- OS client dump space (**srv_dump**)

NOTE: If you have OS clients of secondary releases, refer to the section "Multiple Release Areas" for information.

As you read the following sections that discuss the OS client logical disks in detail, record the OS client logical disk sizes on the Logical Disk Planning Worksheet in Appendix E.

OS Client Directory (srv)

The logical disk for the OS clients' directory holds OS client file systems such as root directories and swap space. A typical name for this logical disk is **srv** and it is typically mounted at **/srv**.

The **/srv** logical disk should be large enough to accommodate the **sysadm** database, which is approximately 5,000 blocks. You should create a single logical disk named **srv** with a size of 5,000 blocks.

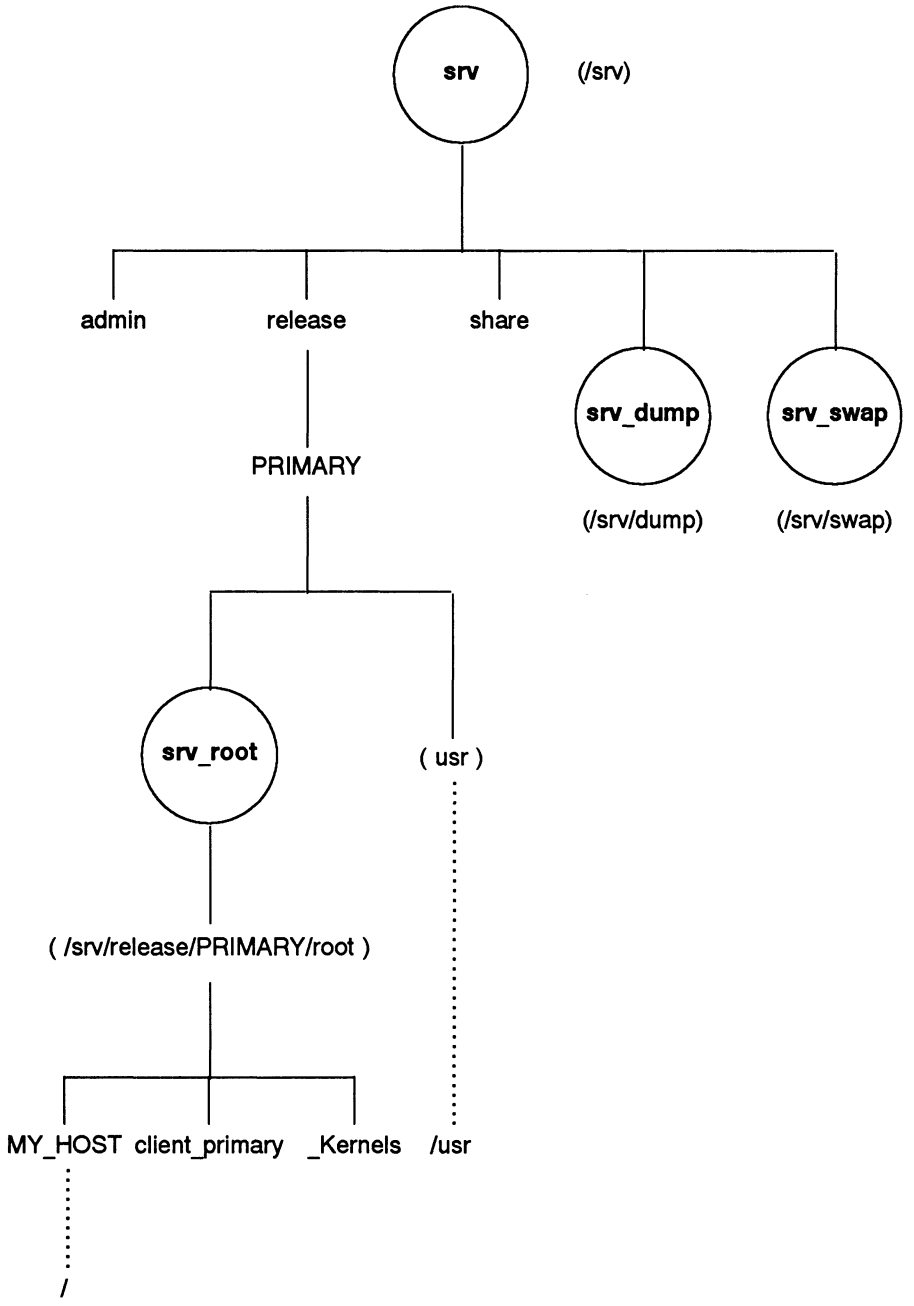
OS Client Root Space (srv_root)

The OS client root space is a single logical disk that contains all the root directories for all clients. The directory pathname for this logical disk is **/srv/release/PRIMARY/root** by default. The root directory contains subdirectories that correspond to each client. Instructions in Chapter 8 explain how to add an OS client and mount its file systems.

For the DG/UX system's default root file system, each OS client needs the same amount of space as the OS server: 40,000 blocks. To calculate the size of the logical disk, multiply the number of OS clients by 40,000.

The command that you use to build kernels for OS clients allows you to link all OS clients to the same kernel image, saving disk space. But sharing kernels in this way can result in weakened security because any user can access and change the kernel image. Should you decide to make such links anyway, remember that for OS clients to share a kernel, their root directories (and the directory containing the kernel) must all be on the same logical disk. Thus, you should not distribute OS client root directories over different logical disks.

Figure 2-1 shows a typical **/srv** file system.



Legend:

Circles represent the logical disks.
Pathnames within parentheses indicate mount points.
... indicate symbolic links to the named directories.

Figure 2-1 *./srv File System*

OS Client Swap Space (srv_swap)

Swap space is the temporary storage location of an active page from a process on a logical disk. A page is stored (or paged) in swap space when there are more active processes than can simultaneously fit into the computer's main memory. When memory resources become available, the temporarily suspended page is sent back into main memory for execution.

The amount of swap space that you need depends on the amount of physical memory in your machine, the nature and number of the applications you run, and the number of users on the system. If your programs allocate large portions of memory, you may need more swap area. Insufficient swap area can result in the termination of running processes and errors such as `From System: out of paging area space` at the system console. If you encounter such an error, you need to create more swap space or reduce your system load. You can also create multiple swap logical disks to supplement existing swap logical space.

Like the OS server, OS clients (including OS clients of a secondary release) require swap space. Unlike the OS server, an OS client's swap space is provided by an actual file on the OS server. These files are located in the **/srv/swap** file system, whose logical disk is usually called **srv_swap.**

Use the following formula to compute swap space for OS clients.

number-of-OS-clients * 50,000 *blocks-per-OS-client)* * *overhead*

(4 * 50000) * 1.17

234,000 blocks

OS Client Dump Space (srv_dump)

You need to allocate space for OS client system dumps. When a system panics or hangs, you write the contents of the system's memory to a file or tape so that Data General engineers can diagnose the problem.

On a system with a tape device, you configure your kernel so that it dumps to tape. On an OS client without a tape device, you configure the system so it dumps over the network to a file on a physical disk device attached to the OS server. If you don't create a **dump** logical disk, ensure that **/srv** is sufficiently large to accommodate a dump.

The maximum amount of space you need for **/srv/dump** is equal to the total size of physical memory on all of your OS client systems (including OS clients of secondary releases). In practice, however, you need less space than this for two reasons:

- OS clients do not need to make systems dumps all at the same time.
- You do not keep system dump files online for very long. Space for one or two system dumps is probably sufficient.

If each OS client has 16 Mbytes (16,777,216 bytes or 32,768 blocks) of physical memory, then 33,000 blocks is sufficient to hold one system dump. Remember to add 17% for file system overhead. Thus, the total size would be 38,600 blocks (33,000 + (.17 * 33,000)).

Multiple Release Areas

NOTE: This section applies only to the OS client-server configuration in which multiple operating systems are being used. If you are not installing a subsequent OS release, skip this section.

While the 5.4 release of the DG/UX system is the most common operating system resource an OS server provides to its OS clients, it can offer multiple releases of an operating system to OS clients, each of which is located in a separate release area. For example, an OS server can support earlier versions of the DG/UX system or other operating systems such as SunOS. The number of releases an OS server can support is restricted only by available physical disk resources and system performance.

While the 5.4 release of the DG/UX system is available to OS clients from **/srv/release/PRIMARY**, you locate a secondary release at **/srv/release/***release-name*, where *release-name* is the name for the OS release. Examples of OS release names are **dgux_432** or **SunOS**.

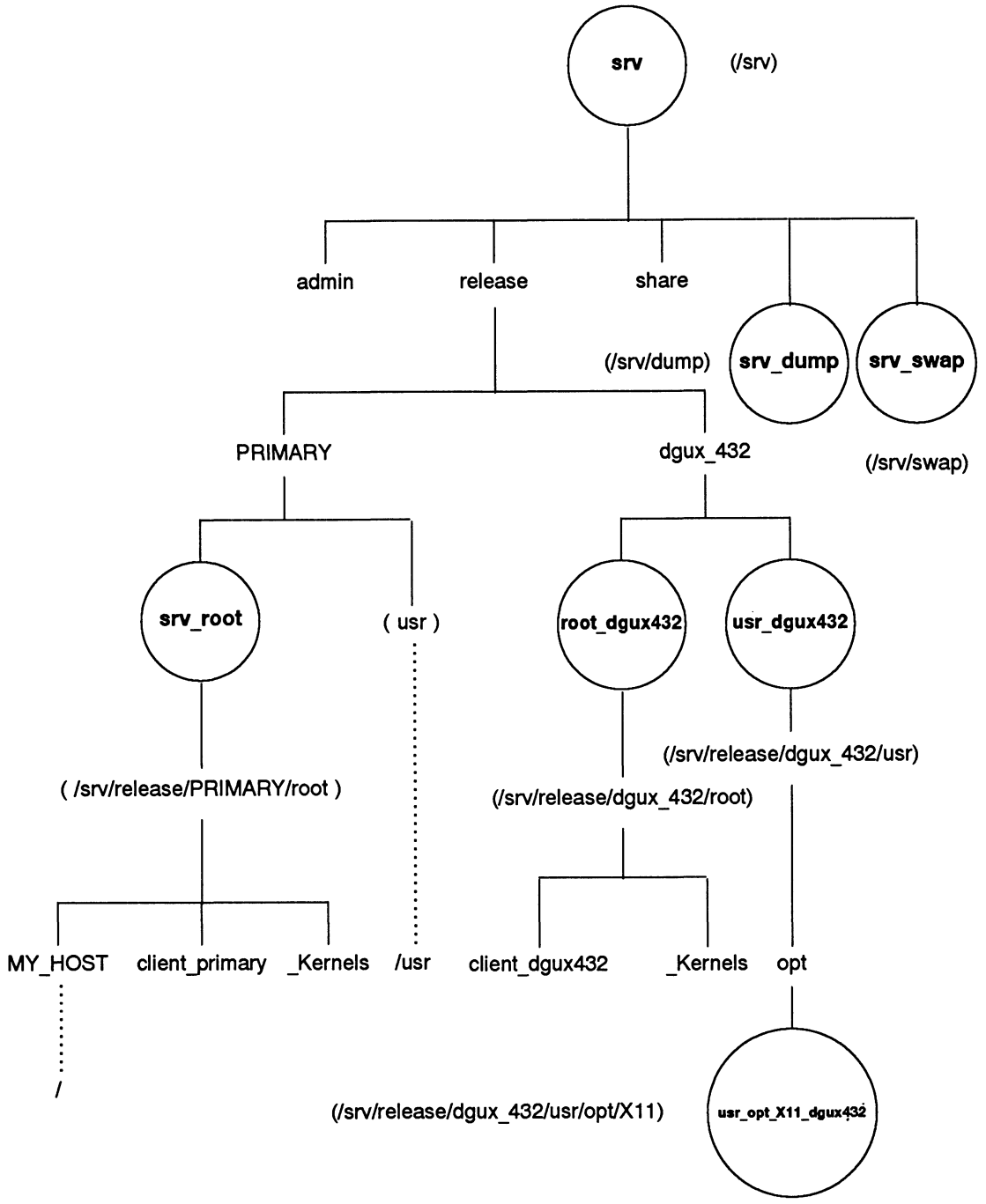
The number of logical disks you create for a secondary release area depends on the particular OS release you are installing. If you are creating an OS release area for an operating system other than the DG/UX system, consult the documentation and release notice that accompanies that operating system. If you are creating a secondary release area for the 4.32 release of the DG/UX system, you must create three logical disks:

- OS client root space on secondary release (**root_dgux432**).
- OS client usr space on secondary release (**usr_dgux432**).
- OS client X11 space on secondary release (**usr_opt_X11_dgux432**).

The logical disk names in parentheses are examples only; you can name the logical disks whatever you want. Notice that you do not have to create a dump area or a swap space. Those resources are available from the **/srv** directory structure (for example, **srv_dump** and **srv_swap**).

A difference between the **/srv/release/PRIMARY** and the **/srv/release/***secondary-release* directory structures is that the latter requires an explicit logical disk for the **/srv/release/***secondary-release***/usr** and **/srv/release/***secondary-release***/usr/opt/X11** file systems.

Figure 2-2 shows a typical example of a secondary release area, **/srv/release/dgux_432**, for example.



Legend:

Circles represent the logical disks.
Pathnames within parentheses indicate mount points.
... indicate symbolic links to the named directories.

Figure 2-2 *Isrv/release/dgux_432 File System*

OS Client Root Space (root_dgux432)

The OS client root space is a single logical disk that contains all the root directories for all clients. You can mount this logical disk where you want; an example of a directory mount point is **/srv/release/dgux_432/root**. The root directory contains subdirectories that correspond to each client. Instructions are provided in a later chapter for adding an OS client and mounting its file systems.

For the DG/UX system's default root file system, each OS client needs the same amount of space as the OS server: 40,000 blocks. To calculate the size of the logical disk, multiply the number of OS clients by 40,000.

OS Client usr Space (usr_dgux432)

The **usr** logical disk, whose file system mount point can be **/srv/release/dgux_432/usr**, is reserved for system-level programs, facilities, and software packages. The **/srv/release/dgux_432/usr** directory holds subdirectories that will contain such things as database and configuration files, administrator commands, stand-alone utilities and bootstraps, and user commands.

If you do not accept the default logical disk size of 160,000 blocks, record the desired logical disk size on the Logical Disk Planning Worksheets in Appendix E.

X11 Package Space (usr_opt_X11_dgux432)

The **usr_opt_X11_dgux432** logical disk is required only if you have purchased the Client-Server AViiON System Package and you intend to install the DG/UX X Window System. This logical disk, whose file system mount point is **/srv/release/dgux_432/usr/opt/X11**, contains the DG/UX X Window System package. The package includes X11, Looking Glass, documentation, and an X server development environment.

If you do not accept the default logical disk minimum size of 105,000 blocks, record the desired size on the Logical Disk Planning Worksheets in Appendix E. Do not reduce the size of this logical disk.

Miscellany

Each site must determine how to set up home directories, work directories, and tools directories. These logical disks vary in size according to the work requirements placed on them.

User Home Directories (home)

A home directory is useful for containing each individual's work on the system. It also contains files that customize each user's shell, electronic mail environment, and X Window environment (for example, through the **.login** or **.profile**, **.mailrc**, and **.Xdefaults** setup files).

You should create one logical disk (named **home** as an example) to accommodate all users' home directories.

A user's home directory requires a variable amount of space depending on the work the user does and the files the user accumulates. As an example, suppose you determine that each user's home directory needs 40,000 blocks in which to save mail, write memos, collect product specifications, and accommodate various temporary files that the system or other programs produce, such as scratch files. With five users, your logical disk would need to be 200,000 blocks.

You must account for any OS clients' home directories as well as directories for local users. Chapter 3 contains instructions for adding user accounts.

Software Packages (usr_opt_pkg)

You can create a separate logical disk for each software package, or you can create a single logical disk for all software packages. If you put all software packages on one logical disk, it must be sufficiently large to accommodate the sum of the individual software packages. Economical disk planning for such a logical disk may be difficult to forecast. To save disk space, consider making a logical disk for each package so that you use exactly the amount of physical disk space that is required per software package. Examples of software packages, whether from Data General Corporation or from other vendors, are databases, spreadsheets, debugging tools, and publishing systems. The release notice that accompanies the software package specifies its size requirements.

When calculating the size of a logical disk for a software package, add 17 percent of the package size (in Mbytes) for file system overhead requirements. For example, suppose you wanted to create a logical disk for a software package that was 100 Mbytes.

software-package-size-requirement + overhead = logical-disk-size-in-Mbytes

100 + 17% = 117 Mbytes (or 59904 blocks)

Most often, you should mount logical disks for software packages at **/usr/opt** in the DG/UX file system. For example, suppose the 100 Mbyte software package was named **pkg**. You might create a logical disk named **usr_opt_pkg** and mount it at **/usr/opt/pkg**. That way, the name of the logical disk reminds you of its mount point.

Work Directories

Work directories, like software development build areas or large databases, may serve as common work areas for your system's users. If such work directories would be too large for a single physical disk unit, or if you suspect that disk I/O performance could deteriorate during multiuser access, you can break up the logical disk for work directories into multiple pieces and distributing them across multiple physical disks.

Tools Packages

Tools are commonly placed in **/usr/local/bin**. The DG/UX system has a **/usr/local** mount point included in the **/usr** file system. But additional tools packages can be another candidate for a logical disk. An appropriately named tools directory is easily recognizable and accessible to users on your network. While you may allow read and write access to a work directory, you may choose to limit users to read-only access to a directory containing tools.

Temporary File Space (tmp)

User programs use temporary space when they start and as they execute. Large program compilations, heavy network traffic, and large database I/O activities require temporary file space.

To segregate temporary file space from the **/** directory, you can create an explicit logical disk for temporary file space and mount it on the **/var/tmp** directory. By default, 40,000 blocks are allocated to the **root** logical disk for its file system. After the **/** file system is loaded, 12 Mbytes remain as free space which can be used for **/var** and **/tmp**. All subdirectories of **/var** use the same space. In addition, you can create separate logical disks for the **mail** and **news** subdirectories of **/var**.

Extra Swap Space (extra_swap)

With procedures described in *Installing the DG/UX™ System*, you created a logical disk for swap space of 50,000 blocks. However, additional space may be required to satisfy your applications' demand for virtual memory. You should consider adding more swap space if the message `From System: out of paging area space` appears on the system console or when the number of users or application load on the system increases. Use of swap space varies considerably from application to application.

NOTE: Extra swap space is not needed for OS clients.

The operating system automatically balances paging activity among multiple swap areas in a system. The arrangement of swap space logical disks of roughly the same size for each physical disk in a system offers maximum performance. The maximum number of swap areas system-wide is eight.

Assessing Physical Disk Capacity

The capacity of a physical disk is identified by its model number. Read your hardware installation and setup manual and the packing list to determine the model numbers of your disk devices. Table 2-1 lists the disk device model numbers and corresponding sizes in megabytes and blocks.

Table 2-1 Disk Device Model Numbers and Corresponding Capacities

Disk Device Model Number	Size (Mbytes)	Number of 512-Byte Blocks
6491 SCSI (full-height)	322	659456
6539 SCSI (half-height)	179	366592
6554 SCSI (full-height)	662	1355776
6662 SCSI (half-height)	332	679936
6685 SCSI (full-height)	1040	2129920
6740 SCSI (full-height)	1040	2129920
6442 ESDI (full-height)	327	669696
6555 ESDI (full-height)	648	1327104
6661 ESDI (half-height)	330	675840
6541 SMD	1066	2183168
6542 SMD*	2132	4366336
5070DR Optical WORM Disk	2458	5033164
5070S Optical WORM Disk	900	1843200
6627 Erasable Optical Disk	590	1208329
6627 Erasable Optical Disk	650	1331200
3.5-inch diskette	0.72	1440
3.5-inch diskette	1.44	2949
5.25-inch diskette	0.36	720
5.25-inch diskette	0.72	1440
5.25-inch diskette	1.20	2457
7902 HADA**	5005	10250625
7917 HADA**	5005	10250625
7922 HADA**	5005	10250625

* This disk model contains two disk units of 1066 Mbytes each, producing a total capacity of 2132 Mbytes.

** Each High-Availability Disk Array Subsystem contains five half-height or two full-height and one half-height SCSI disk devices. You need to find out the specific SCSI devices and corresponding capacities to determine the capacity for each of the HADA disk devices.

To determine the remaining space on your boot disk, refer to the Installation Planning Worksheets you completed in *Installing the DG/UX™ System.* You may prefer to check the capacities of your disk devices using the **diskman** "Display a Physical Disk's Layout" option (see Appendix B).

In addition to the model number and capacity, you also need to know the DG/UX common device specification for each physical disk device. Record the DG/UX common device specification and capacity on the logical Disk Planning Worksheet in Appendix E. For more information on DG/UX device-naming conventions, refer to Appendix D.

Arranging the Logical Disks on Physical Disks

The simplest strategy for arranging logical disks is to place them contiguously on a single physical disk until all space has been used. If you have unrestricted resources, you may choose to leave some space between logical disks for future logical disk expansions. If on the other hand you have limited disk resources, you may want to create a logical disk in pieces, taking advantage of available disk free space on one or more physical disks.

For example, you could assemble six pieces to form a 620,000-block logical disk. You could spread these pieces over three physical disks as follows:

Logical Disk Piece	Physical Disk	Size (in Blocks)
1 of 6	**sd(insc(0),0,0)**	20,000
2 of 6	**sd(insc(0),0,0)**	60,000
3 of 6	**sd(insc(0),0,0)**	153,263
4 of 6	**sd(insc(0),1,0)**	128,912
5 of 6	**sd(insc(0),1,0)**	128,912
6 of 6	**sd(insc(0),2,0)**	128,913
TOTAL		620,000

You can assemble up to 32 pieces to form a logical disk.

Another reason to create a logical disk in pieces is to speed access to the information on the disk. For example, if you arrange a frequently accessed logical disk such as a database in pieces across multiple physical disks, those disk devices can share the I/O load, thus improving performance.

When you create a logical disk in pieces, you can place frequently accessed logical disks (or pieces) in the center of a given disk for quicker read-write access. It is generally easier for a read/write head to seek to the center than the beginning or end of a physical disk.

Other methods for tuning disk I/O performance are covered in *Managing the DG/UX™ System.*

If you are customizing an existing system, you can examine layouts of your physical disk devices using the Diskman "Display a Physical Disk's Layout" option. See Appendix B for details.

Deciding Where to Mount the File Systems of a Logical Disk

After you determine the size and arrangement of your logical disks, you need to decide where to put the logical disks in the DG/UX directory structure. This place is called the mount point; it provides a directory pathname for the file system that is placed on the logical disk. Figure 2-3 shows a typical file system with logical disks and their mount points.

This figure illustrates the arrangement of logical disks in the hierarchically structured DG/UX file system. Each mount point specifies the absolute pathname of a logical disk (that is, the location of the logical disk in the file system). In this typical DG/UX file system, in addition to the required logical disks, there are three more logical disks and mount points.

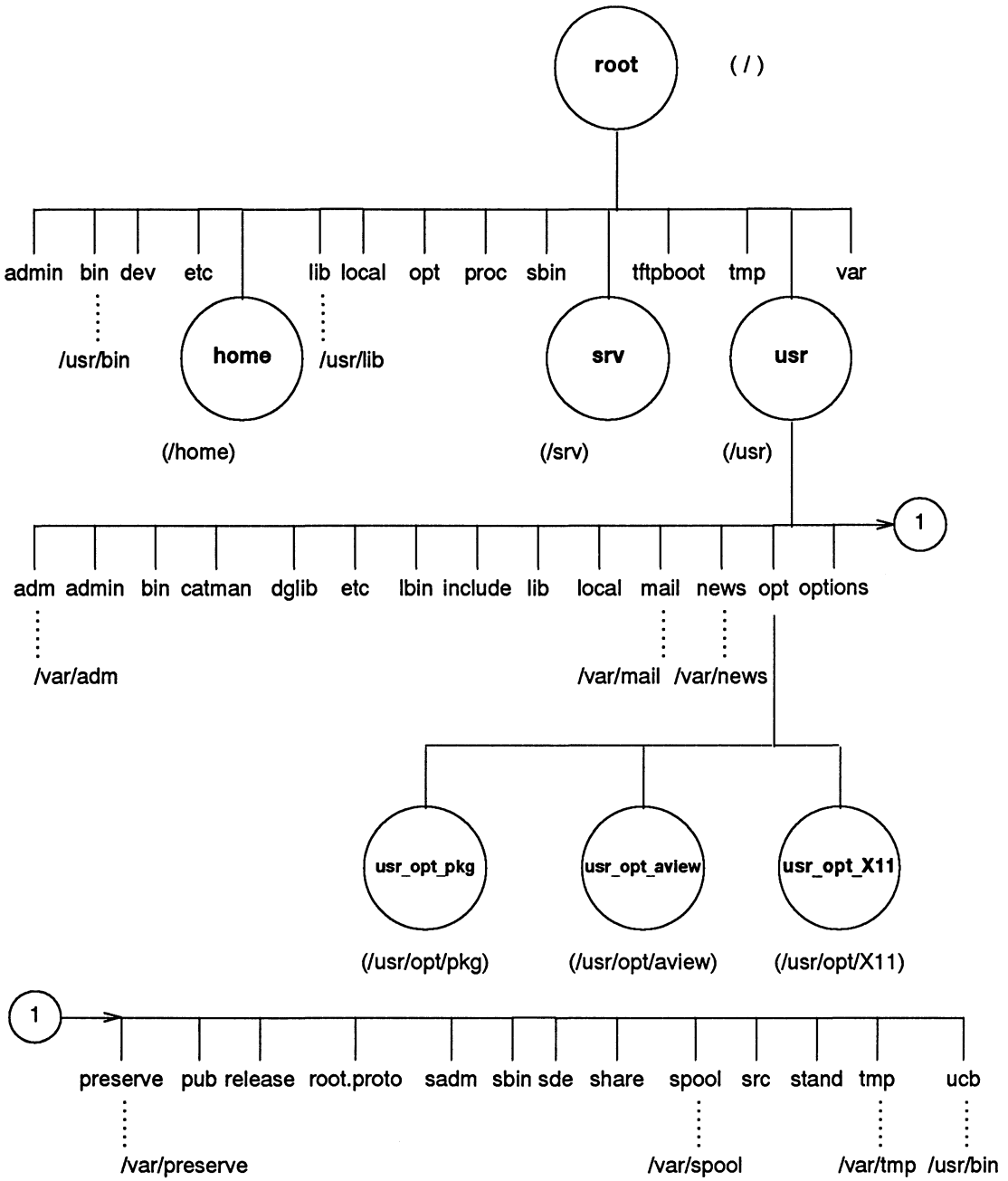
Logical Disk Name	Mount Point
home	**/home**
srv	**/srv**
usr_opt_pkg	**/usr/opt/pkg**

Using Memory File Systems

You can create a file system in memory without using a logical disk. Memory file systems are faster than normal file systems on physical disks. Storing temporary files and heavily-accessed executables in these file systems avoids heavy disk I/O. But because writable data can be destroyed if the system crashes, you should store only read-only data in memory file systems.

Typically, memory file systems are small. Obviously, you should use less than your system's memory capacity for the file system. The computer's power-up display reports the computer's physical memory size in Mbytes. For example, if you have 16 Mbytes of physical memory, you should use considerably less than 32,768 blocks for a memory file system.

While you do not create a logical disk for the memory file system, you do have to mount it as you would a file system on a physical disk. See Chapter 3 for information about how to add a local file system. See *Managing the DG/UX™ System* and the **mfs**(4) manual page for more information on memory file systems.



Legend:

Circles represent the logical disks.
Pathnames within parentheses indicate mount points.
... indicate symbolic links to the named directories.
Circled numbers connect the two parts of the diagram.

Figure 2-3 *Typical Logical Disks and Mount Points on the DG/UX File System*

End of Chapter

Chapter 3
Creating Logical Disks and File Systems

In this chapter you will create each logical disk for which you planned in Chapter 2. These topics are covered:

- Registering a physical disk
- Creating a logical disk
- Adding a local file system
- Adding a remote file system

It is assumed that you have formatted the physical disks on which you intend to create logical disks. If you have completed the instructions in *Installing the DG/UX™ System*, all disks are formatted. If you have recently added a new physical disk to your hardware configuration, you must format it before proceeding. See Chapter 10 for instructions for adding physical disks.

Registering a Physical Disk

Registering a physical disk makes its logical disks known to the system. You cannot access logical disks on a device unless you register the physical disk. Furthermore, you must register a physical disk before you can add a file system on the device. If the physical disk has no system areas (for example, the device is a DOS-formatted diskette), you do not need to register it. Also, do not register a CD-ROM device. It has been previously formatted according to either the High Sierra or ISO 9660 standard, which obviates registration.

1. From the **sysadm** main menu, select the following options:

   ```
   Device -> Disk -> Manage
   ```

   These selections take you to the **diskman** Main Menu, from which you make more selections.

2. Select option 1, Physical Disk Management Menu.

3. Select option 1, Register, Deregister or List Registered Physical Disks.

4. Select option 1, Register a Physical Disk.

5. Enter the physical device's specification using the Physical Disk Planning Worksheets that you completed in Appendix E.

A typical dialogue follows:

```
=======================================================

Register a Physical Disk

=======================================================

Enter the physical disk specification in DG/UX
common format: sd(insc(0),1,0) ↵

Physical disk sd(insc(0),1,0) has been registered.
```

If the device is already registered or is not formatted (see Chapter 6 for procedures on preparing a new device), an error message is displayed.

To list the registered physical disks, select option 3 from the Physical Disk Registration Menu. A typical display follows:

```
=======================================================

List Registered Physical Disks

=======================================================

Currently registered physical disks are:

        sd(insc@7(FFF8A000),1,0)
        sd(insc@7(FFF8A000),0,0)

Press New Line to continue.
```

This display converts the device name from short to long format. See Appendix D for information on the DG/UX common device specification format.

Creating a Logical Disk

Use these procedures to create a logical disk.

1. From the **sysadm** main menu, select the following options:

   ```
   Device -> Disk -> Manage
   ```

   These selections take you to the **diskman** Main Menu, from which you make more selections.

2. Select option 2, Logical Disk Management Menu.

3. Select option 1, Create a Logical Disk.

4. Enter the name of the logical disk that you want to create using the Logical Disk Planning Worksheets that you completed in Appendix E, answering prompts to provide the following information:

   - Physical disk name in DG/UX common device specification format
   - Logical disk name
   - Whether or not the logical disk will be formed in pieces
   - Location of logical disk (or logical disk pieces) on the physical disk
   - Logical disk (or piece) size (in blocks)

Suppose you want to create a logical disk named **accounts** on physical device **sd(insc(0),0,0)**. After you specify the name of the physical device, **diskman** asks whether you want to display the disk layout.

```
Enter the logical disk name: accounts ↵
Logical disk piece 1:
Enter the physical disk specification in DG/UX common format: sd(insc(0),0,0) ↵
Do you want to display the layout of the physical disk? [no]
```

Suppose you display the disk layout.

```
Do you want to display the layout of the physical disk? [no] yes ↵

System Areas on Physical Disk sd(insc(0),0,0):


       Area Name              LD Piece      Physical Disk     Size
                               Number        Address
                                             of Area

      Primary System Area      ......              0          8
    System Bootstrap Area      ......              8        500
    Secondary System Area      ......            508          8
  Primary Bad Block Table      ......            516          3
        Primary LDP Table      ......            519          9
     Bad Block Remap Area      ......            528        189
Secondary Bad Block Table      ......            717          3
      Secondary LDP Table      ......            720          9
                     swap      1 of 1            729      50000
                     root      1 of 1          50729      40000
                      usr      1 of 1          90729     240000
              usr_opt_X11      1 of 1         330729     105000
            usr_opt_aview      1 of 1         330729      10000
             (Free Space)      ......         340729     290325

Total physical disk size: 631053 blocks.
Unallocated space: 290325 blocks.

Do you want to switch to another physical disk instead? [no] ↵
```

Suppose that based on the information displayed, you decide to place the accounts logical disk at the next available disk address, in this case 340729, and that you want the logical disk arranged in one 200,000-block piece. Notice that **diskman** provides the next available disk address as the default for the starting block of your logical disk.

```
Enter the physical disk address of the starting block
of the logical disk piece: [340729] ↵
Enter the size in blocks of the logical disk piece 1 [290325]: 200000 ↵
Do you want to specify any more pieces for this logical disk? [no] ↵
The logical disk 'accounts' has been created.
```

When you create a logical disk, **diskman** gives you the opportunity to create a file system on it. It prompts you for options to the **mkfs**(1M) command, which **diskman** executes to create the file system. See the **mkfs**(1M) manual page for available options. Suppose you specify no flags or options to **mkfs**.

```
Do you want to make a file system on this logical disk? [yes] ↵
Enter the flags and options you want to specify: ↵

Making a file system on logical disk 'accounts' ...
Made a file system on logical disk 'accounts'.

Press New Line to continue. ↵
```

Repeat this procedure to create each logical disk. When finished, return to the **diskman** main menu by entering the ^ character and the **sysadm** main menu by entering **q**.

Adding a Local File System

File systems that you create at your host are considered local. When you use **sysadm** to add a local file system, you add an entry in the file system table (**/etc/fstab**) and you mount the file system. Each file system listed in the **fstab** file is mounted automatically each time you boot your system.

NOTE: A file system must be mounted before you can access it.

To add a local file system, follow this path through **sysadm**:

```
File System -> Local Filesys -> Add
```

There are four types of local file systems that you can add, each of which is related to the particular device type. They are:

- DG/UX file system on a disk device
- Memory-resident file system
- File system on an ISO-9660 or High Sierra formatted CD-ROM device
- File system on a DOS-formatted diskette

The procedures to add each of these types of file systems is given in the following sections.

Adding a DG/UX File System on a Disk Device

The most common file system type is a DG/UX file system on a disk device. When prompted by **sysadm** for the file system type, this is the default. To add the file system contained on the logical disk named **accounts** at mount point **/accounts**, you would answer the **sysadm** prompts as follows:

```
File System Type: [dg/ux] ↵
Logical Disk: accounts ↵
Mount Directory: /accounts ↵
```

Next you are prompted for the write permission of the file system.

```
Write Permission: [Read/Write] ↵
```

By default, file systems are readable and writable You can override the default by specifying the desired permission, such as read-only (**ro**).

Next, you are prompted for dump frequency.

```
Dump Frequency: [Daily] ↵
```

Dump frequency determines how often your file system is archived onto tape. You can base a file system's dump frequency on the importance of the data and its volatility. Setting this value alone does not fully enable archiving. See the **dump2**(1M) and **dumpcycle(4M)** manual pages for more information. Legal values are:

d Daily, weekly, and monthly

w Weekly and monthly

m Monthly only

x No archiving

The next three prompts concern attributes of the file system-checking facility **fsck**(1M). The **Fsck Pass Number** indicates the order in which the DG/UX system checks for corrupted file systems when the system is booted.

```
Fsck Pass Number: [1] ↵
```

Values range from 0 to 9; 0 means never checked; 1, checked on first pass; and 9, checked last. Here, the default of 1 is selected. See the **fsck**(1M) manual page for more information.

Fsck Logging enables or disables file system logging.

```
Fsck Logging? [no]
```

The log collects file system modifications, which facilitates a quick recovery of the file system when the system crashes. Since logging has some negative impact on runtime write performance, logging is recommended primarily when rapid recovery and high availability are crucial.

An exportable file system allows other hosts connected to the network to mount it on their own DG/UX directories. The next prompt asks whether you want the file system to be exportable.

```
Exportable? [no]
```

If you answer `yes`, you are prompted for export options, which restrict how an exported file system is accessed. Making a file system exportable creates its entry in the **/etc/exports** file.

```
Export Options:
```

Specify export options with a comma-separated list of characteristics. Precede the first option with a hyphen (-) and no intervening space. Permitted characteristics are as follows:

secure Requires OS clients to use a secure protocol when accessing the directory.

ro Exports the directory read-only. Read-write is the default.

rw=*host-name*[**:***hostname*]...
Exports the directory read-only to all systems except those specified by this option. Systems specified in this option have read-write access to the directory. Allowing

another system read-write access to the directory does not override normal file and directory permissions. Read-write is the default.

anon=*uid*
A user can access an exported file system only if their password entries exist in the remote system's **/etc/password** file. Alternatively, users can access the exported file system by way of the **anon** option, which establishes an effective user ID for anonymous users. Superusers (user ID **0**) can access an exported file system only through the **root** option (see next option). The default anonymous user ID is **−2**. The value **−1** disables anonymous access.

root=*hostname*[**:***hostname*]...
Gives root (superuser) access only to superusers from the specified hosts. By default, superusers from other systems do not have superuser access to the directory. A superuser is any user whose user ID is **0** (**root** and **sysadm**, for example).

access=*client*[**:***client*]...
Gives mount access to each OS client listed. An OS client can be represented as *hostname* or *netgroup* (see the **netgroup**(5) manual page). Each client in the list is first checked for in the **/etc/netgroup** database and then the **/etc/hosts** database. The default value allows any computer to mount the given directory.

The final query asks if you wish to mount and export the file system to which you answer `yes`. After you answer all the prompts for information, the file system is mounted at the desired mount point.

```
Mount and export the file system? [yes]  ↵
Ok to perform the operation? [yes] ↵

File system added: /account
File system mounted: /account
File system exported: /account
```

Repeat this procedure for each logical disk that you created in the previous section. When finished, return to the **sysadm** main menu by entering **q**. Consult your Logical Disk Planning Worksheets that you completed in Appendix E for details.

Adding a Memory File System (ramdisk)

A memory file system or ramdisk is a file system that resides in the computer's memory rather than on a physical disk. If your AViiON computer has sufficient physical memory, you can create a file system in memory without creating a logical disk. Memory-resident file systems are faster than normal file systems on physical disks and are ideal for storing temporary files and heavily accessed executables to avoid heavy disk I/O. Read-only data is the best candidate for storage in memory as writable data will be destroyed if the system crashes.

The computer's power-up display reports the computer's physical memory size in megabytes. For example, if you have 16 Mbytes (32,768 blocks), you should keep this limit in mind when storing data in a memory file system. You risk a degradation in system performance if the size of a memory file system becomes too large.

To add a memory file system, follow this path through **sysadm**:

```
File System -> Local Filesys -> Add
```

To create a memory-resident file system, enter **ramdisk** at the `File System Type` prompt:

```
File System Type: [dg/ux] ramdisk
```

The next two queries ask for the name of a device file in the **/dev** directory that is to be associated with the file system, and the mount point directory. The device file will be created when the file system is mounted. It may not already exist.

```
Device File: ramdisk ↵
Mount Directory: /ramdisk ↵
```

NOTE: A logical disk was not created.

Next, you must choose whether or not the memory file system should be wired in memory.

```
Use Wired Memory? [no] ↵
```

By default, a memory file system is not wired, which means that the OS may swap parts of the file system to disk to free needed physical memory. If wired, the entire file system remains in the system's physical memory. Make sure your computer has enough physical memory to hold the file system before mounting it with this option.

Next, you specify the maximum number of 512-byte blocks that the memory file system can use.

```
Maximum File Space: [2048] ↵
```

The system does not allocate the memory until the file system requires it; therefore, entering a high value will not necessarily use up a lot of memory. The system is allowed to allocate either the maximum size that you specify or the amount of memory available, whichever is less.

Next you are prompted for the maximum number of files that can be present in the memory-resident file system.

```
Maximum File Count: [16384]
```

Finally you are asked whether you want to make the file system exportable.

```
Exportable? [no] yes ↵
Export Options:
```

An exportable file system allows other hosts connected to the network to mount it on their own DG/UX directories. Answer **yes** to export the file system. See "Adding a DG/UX File System on a Disk Drive" for a review of the export options.

The dialogue ends with **sysadm** asking whether the file system should be mounted.

```
Mount and export the file system? [yes] ↵
Ok to perform the operation? [yes] ↵

File system added: /ramdisk
File system mounted: /ramdisk
File system exported: /ramdisk
```

As you can see, messages appear that confirm that the memory-resident file system has been added, mounted, and exported. When finished, return to the the **sysadm** main menu by entering **q**.

Adding a File System for a CD-ROM Device

A CD-ROM is a compact disk-read-only memory device, which is formatted in either the High Sierra or ISO 9660 standard.

To add a file system on a CD-ROM device, follow this path through **sysadm**:

```
File System -> Local Filesys -> Add
```

Enter **cdrom** at the `File System Type:` prompt.

```
File System Type: [dg/ux] cdrom ↵
```

The next two queries ask for the name of the device file associated with the CD-ROM device in the **/dev** directory that is to be associated with the file system, and the mount point directory. Assume that you want to mount the file system on a device whose node is **/dev/pdsk/1** (short format) at mount point **/accounts**.

```
Device file: /dev/pdsk/1 ↵
Mount Directory: /accounts ↵
```

The file system is read-only by default.

Next, you are asked whether you want to make the file system exportable. An exportable file system allows other hosts connected to the network to mount it on their own DG/UX directories. The export options are the same as for a DG/UX file type.

```
Exportable? [no] yes ↵
Export Options:
```

See "Adding a DG/UX File System on a Disk Drive" for a review of the export options.

Finally, you are asked whether you want to mount and export the file system.

```
Mount and export the file system? [yes]  ↵
Ok to perform the operation? [yes] ↵

File system added: /account
File system mounted: /account
File system exported: /account
```

After you answer all the prompts for information, the file system is mounted at the desired mount point. When finished, return to the the **sysadm** main menu by entering **q**.

Adding a File System for a DOS-Formatted Diskette

A DOS file system is on an MS-DOS-formatted diskette.

To add a DOS file system, follow this path through **sysadm**:

```
File System -> Local Filesys -> Add
```

To add a DOS file system, answer **dos** at the `File System Type:` prompt.

```
File System Type: [dg/ux] dos ↵
```

The next two queries ask for the name of the device file that is associated with the DOS-formatted diskette drive in the **/dev** directory, and the mount point directory. Assume that you want to add a file system associated with device file **/dev/pdsk/1** at mount point **/accounts.**

```
Device File: /dev/pdsk/1 ↵
Mount Directory: /accounts ↵
```

The next prompt asks you to specify the write permission of the file system. The file system is readable and writable by default, which you can override by specifying the desired permission, such as read-only (**ro**).

```
Write Permission: [Read/Write] ↵
```

The next prompt asks you to specify file permissions. Enter the permissions mask that you want for the files in the DOS file system. For example, 0777 gives all users read, write, and execute permissions.

```
File Permissions: [0777] ? ↵
```

For more information on permissions masks, see the **chmod**(1) manual page.

The next prompt asks whether you want to make the file system exportable. If you choose to export the file system, this operation adds an entry for the file system to the **/etc/exports** file. If you choose not to export the file system, this operation removes any entry for the file system from the **/etc/exports** file.

```
Exportable? [no] yes ↵
```

Exporting a file system gives other systems in your network access to the file system by allowing them to mount it on their own system. Allowing another system to mount your file system does not necessarily give them superuser access to the file system. You may grant such access by specifying the correct export option, which you specify at the next prompt

```
Export Options:
```

The export options are the same as for a DG/UX file type. See "Adding a DG/UX File System on a Disk Drive" for a review of the export options.

The final query asks if you wish to mount and export the file system to which you accept the default `yes`.

```
Mount and export the file system? [yes]
Ok to perform the operation? [yes] ↵

File system added: /account
File system mounted: /account
File system exported: /account
```

After you answer all the prompts for information, the file system is mounted at the desired mount point. When finished, return to the the **sysadm** main menu by entering **q**.

Adding a Remote File System

In addition to mounting file systems on logical disks that you create locally, you may mount remote file systems. Remote mounting provides access to a single file system from multiple hosts on a LAN.

As an example, you may want to access a file system containing account information about a client of your company; its logical disk is named **minn_brands** and is mounted at **/accounts/midwest/minn_brands** on remote host **igor**. To mount it at **/midwest** on your DG/UX file system, follow this path through **sysadm**:

```
File System -> Remote Filesys -> Add
```

First, supply a mount directory, which is the location on your DG/UX file system at which the remote file system is to be mounted.

```
Mount Directory: /midwest ↵
```

Next, enter the remote host name and the mount point on that host.

```
Remote Host Name: igor ↵
Remote Mount Directory:  /accounts/midwest/minn_brands ↵
```

Next, specify the write permissions (the default is read/write) and whether you prefer a hard or soft mount of a remote file system.

```
Write Permission: [Read/Write] ↵
NFS Mount Type: [Hard]? ↵
```

With a hard mount, if the remote host crashes, your request to access a remote file may hang. However, a hard mount gives more reliability for write operations. If the remote file system were soft mounted and if the remote host crashed, your I/O request would not hang. You are advised to accept the hard mount default for read/write file systems to avoid the possible loss of data on write operations.

Next, select whether or not NFS should retry in background mode if the server's mount daemon does not respond.

```
Retry in background? [yes] ↵
```

Finally, you are asked to mount the file system.

```
Mount and export the file system? [yes] ↵
Ok to perform the operation? [yes] ↵
File system added: /midwest
File system mounted: /midwest
```

After you answer all the prompts for information, the file system is mounted at the desired mount point. Repeat this procedure to continue mounting the remainder of your remote file systems. When finished, return to the the **sysadm** main menu by entering **q**.

End of Chapter

Chapter 4
Adding User Accounts

This chapter provides instructions for establishing accounts (home directories) for each user on the DG/UX system. These topics are covered:

- Establishing a default set of login account parameters
- Creating a user group
- Creating a user account
- Assigning a password

Establishing a Default Set of Login Account Parameters

NOTE: This operation does not establish the group itself; rather, it creates a default set of login account parameters that are associated with the group.

Follow this path through **sysadm** to establish a default set of login account parameters.

```
User -> Login Account -> Defaults -> Set
```

The **sysadm** program steps you through a series of prompts. At each prompt, press New Line to accept the displayed default or enter a new value.

The first prompt asks for a group name, which identifies a group of users who have access to a given file system.

```
Group: [general] ↵
```

The group name is limited to 32 alphanumeric characters, the first one being alphabetic. However, you are advised to keep the group name short, about eight characters. Specify a group name that is unique within the NIS domain.

Next, you are prompted for the base directory, which is the location of all users' home directories.

```
Base Directory: [/home] ↵
```

Check your Logical Disk Planning Worksheets that you completed in Appendix E for the mount point for the users' home directories.

The next prompt asks for a skeleton directory.

```
Skeleton Directory: [/etc/skel] ↵
```

The skeleton directory contains the prototypes of startup files such as **.profile**, **.login**, and **.cshrc** files for each user's home directory.

Finally, you are prompted for the login shell.

```
Shell Program: [/sbin/sh] /usr/bin/csh ↵
OK to perform operation? [yes] ↵
```

There are several shells (command-line interpreters) from which you can operate after you log in to the DG/UX system. Enter **/sbin/sh** to select the Bourne shell (the default), **/usr/bin/ksh** to select the Korn shell, and **/usr/bin/csh** to select the C shell.

Refer to *Managing the DG/UX™ System* for more information on establishing a group, and refer to *Using the DG/UX™ System* for information on the shells.

Creating a User Group

Membership of users in a group grants them certain file and directory privileges while excluding other users' access outside the group. A user can belong to multiple groups; however, the user's shell is associated with only one group at a time.

To create a user group, follow this path through **sysadm**:

```
User -> Group -> Add
```

You are first prompted for a group name, which can contain up to 32 alphanumeric characters, starting with a letter. The group name must be unique.

```
This host is not the NIS master.  Only the local Group
database will be used.
Group Name: testers ↵
```

You are next asked for the group ID (or GID), which is a number between 0 and 60,000 inclusive. Numbers less than 100 are reserved for system use. The GID must also be unique. The default GID is the highest assigned GID plus one.

```
Group ID: [1] 101 ↵
```

Finally, you are asked to supply the login names of users who will be members of the group. Separate each login with a comma. The login names must already exist as entries in your local **passwd** file, or in the global NIS database, whichever is applicable.

```
Group Members: johnson, dupree ♪
OK to perform operation?  [yes]
Group, 'testers', has been added.
```

Creating a User Account

A user login account establishes each user's working environment. Follow this path through **sysadm** to add a login account.

```
User -> Login Account -> Add
```

The **sysadm** program steps you through a series of prompts. At each prompt, press New Line to accept the displayed default or enter a new value.

Before prompting you for a login name, **sysadm** presents a message about which password database is affected by the changes you are about to make.

```
This host is not the NIS master.  Only the local Password
database will be used.
```

By default, an OS server is set up to be an NIS client, not an NIS master, and so the local password database is used. If an OS server is an NIS master the **yppasswd** database is used instead of the local database. Chapter 7 of this manual describes procedures to follow to become an NIS master. Refer to *Managing ONC™/NFS® and Its Facilities on the DG/UX™ System* for more information on NIS.

The first prompt asks the login name of the user whose account you are adding to the database.

```
Login Name: johnson ♪
```

Each login name must unique. It can contain a combination of these characters: **a-z A-Z 0-9 - and _ .** It should not exceed 8 characters, and it must begin with a letter.

The next prompt asks if you want to implement password aging, a security feature that lets you control how long a password may be in effect before the user is forced to change it.

```
Add password aging for the login account? [no] ♪
```

If you select password aging, the operation presents two additional queries at the end of the dialogue. The next prompt asks for a user ID, which is a number in the range 0 to 60000 assigned sequentially. ID numbers 0 to 100 are reserved for system use.

```
User ID: [25] 101 ♪
```

By default the system offers the next unassigned ID, which is the last user ID plus one.

The next prompt asks for a group name, which signifies the users who can access specific files.

```
Group: [general] testers ♪
```

The group name must already exist. For information about how to create groups, see "Creating a User Group."

Next, you are prompted for the user's home directory.

```
Home Directory: [/accounts/johnson] ↵
```

Next you are prompted for a comment about the user.

```
User Comment: Samuel Johnson, 942-9667 ↵
```

An appropriate use of a user comment is to supply the user's full name and telephone number.

Next you specify the user's login shell.

```
Shell Program: [/sbin/sh] /usr/bin/csh ↵
```

You can accept the default group shell or select another.

After you create the user's home directory, assign a temporary password to the user.

```
Create home directory for johnson? [no] yes ↵
Set an initial password for johnson? [yes] ↵
```

If you answered **yes** to the prompt for password aging, the next two prompts follow:

```
Minimum number of weeks before johnson may change his password: (0-64) [0] ↵
Maximum number of weeks until johnson must change his password: (0-64) [4] ↵
```

The first query asks for the minimum number of weeks that may pass before the user may change his password. Password changing will fail if the user attempts to do so too early. The second query asks for the maximum number of weeks that may pass before the user may change his password. If the time expires before the user changes his password, he will be prompted for a new password the next time he logs in. You are advised to accept the default choices for both.

You are next asked for permission to add the user's account after which you are prompted for user's password.

```
OK to perform operation?  [yes] ↵
User 'johnson' has been added.
New password: ↵
Re-enter new password: ↵
```

In this example, a temporary password **johnson-temp** is assigned (which you confirm a second time), assuming that the user will immediately change it to one that only she or he knows. If you want, you can press the New Line key at the `New password:` prompt, not assigning a password and leaving it to the user to do later. Refer to *Using the DG/UX™ System* for information on changing passwords.

Repeat this procedure as many times as needed. When finished, return to the **sysadm** main menu by entering **q**.

Assigning a Password

You (the user, not the system administrator) are advised to change the temporary password to one of your choice immediately to ensure system security. Use these guidelines to create a password:

Create passwords for each user login, noting these conventions.

- Your password must be different from the login name.
- Your password should not be a circular shift of the login name.
- A password must have at least six characters. At least two characters should be upper- or lowercase alphabetic characters (a-z and A-Z), and at least one character should also be a numeric or special character, such as ?, !, @, $, 0 through 9, or a space.

To assign a new user password, log in with the appropriate login name.

At the shell prompt, type either the **passwd** or **yppasswd** command followed by the user's login name and New Line. If you installed the NIS package, you use **yppasswd**; otherwise, you type **passwd** followed by New Line. A typical dialogue for changing the temporary password to a new one follows:

```
$ yppasswd ↵
Changing yp password for johnson
Old yp password: ↵
New password: ↵
Retype new password: ↵
yellow pages passwd changed on in-squal.rg.com
```

You type the old password followed by the new password, which you are asked to confirm. If you type the password correctly both times, the passwords match and the system displays the shell prompt. To ensure security, the password does not appear on the screen as you type. Refer to the **yppasswd**(1) or **passwd**(1) manual page for more information.

End of Chapter

Chapter 5
Adding Terminals and Modems

This chapter provides instructions for setting up terminals and modems to operate in the DG/UX system environment. It assumes that you have completed hardware installation of the terminal using the appropriate hardware documentation.

This chapter is more heavily oriented to terminals; however, printers and modems can be connected to terminal line controllers. Printers are covered in Chapter 6. In this chapter, the term "port" is used to refer to terminals and modems.

This chapter covers these topics:

- Determining a port's tty line number
- Adding a single terminal
- Adding a single terminal to an existing terminal group
- Adding a group of identical terminals
- Creating a port monitor to manage groups of identical terminals
- Adding a modem
- Listing terminals, port monitors, and port services.

Determining a Port's tty Line Number

The DG/UX system automatically assigns a tty line number to each attached port in the hardware configuration when the system boots. A tty line number takes the form:

tty*x*

where:

x is a sequentially assigned number. As examples, **tty00** refers to the first port, **tty01** the second port, and so on. A file with the name of the tty line number is created in the **/dev** directory each time the system boots.

If you have a variety of different terminal types on multiple line controllers in an OS server configuration, you need to figure out each terminal's tty line number, which can be rather complex.

Refer to Appendix C for information on determining tty lines for asynchronous ports.

Adding a Single Terminal

You can add a single terminal to a new or existing hardware configuration following this path through **sysadm**:

```
Device -> Port -> Terminal -> Add
```

The **sysadm** program steps you through a series of prompts. At each prompt, press New Line to accept the displayed default or enter a new value.

The first prompt asks you to enter the tty device number of the terminal you are adding.

```
Tty device(s): tty00 ↵
```

Enter **?** to display your choices, and check the planning worksheets that you completed in Appendix E. See Appendix C for information on determining tty lines.

The next query asks for the label from the **/etc/ttydefs** file to use for setting the initial termio (terminal I/O) settings.

```
TTY Definition Label: [9600] ↵
```

The **ttydefs** labels appear as the first field in each line of the **/etc/ttydefs** file. For example, a typical asynchronous terminal label is **9600** (which is the default). If 9600 is not the label you want, enter **?** to display your choices.

The next prompt requests a value for the TERM variable that identifies your terminal type.

```
TERM Variable: vt100 ↵
```

This value must correspond to a terminal definition in the **/usr/lib/terminfo** database. To see the entire list of **terminfo** definitions, execute this shell command:

```
# ls -RC /usr/lib/terminfo/* | more ↵
```

Your terminal hardware documentation should specify the correct terminal type. You may also refer to *Managing the DG/UX™ System* for valid Data General terminal TERM variable settings.

The next prompt asking for a disabled response message lets you supply a message to be displayed to users who attempt to log in to a disabled port.

```
Disabled response message: ↵
```

Next, you are asked to choose whether the terminal port service should be enabled or disabled when it is added.

```
Initial state: [ENABLED]  ↵
```

Press New Line to accept the ENABLED default.

Finally, press New Line to accept the values that you have entered.

```
OK to perform operation? [yes] ↵

Adding default port monitor, ttymon1 ...
Port monitor ttymon1 has been added.
Terminal /dev/tty00 have been added.
```

The final messages inform you that the default port monitor **ttymon1** was successfully added to control the single terminal whose tty line number is **tty00**. A port monitor configures and controls access to terminals. By default, a port monitor named **ttymon1** is assigned to manage a single terminal.

NOTE: The **ttymon** port monitor replaces the **getty** and **uugetty** programs from pre-5.4 releases of the DG/UX system.

Adding a Group of Identical Terminals

Use this operation to add a group of terminals having the same terminal definition label and terminal type. The label comes from the **/etc/ttydefs** file, which establishes initial terminal I/O settings. The labels appear as the first field in each line of the **ttydefs** file. For example, you could add ten terminals having an asynchronous terminal label of **9600** and a **vt100** terminal type.

The **ttymon** port monitor replaces the **getty** and **uugetty** programs from pre-5.4 releases of the DG/UX system.

The number of terminals you attach to a port monitor depends on your particular configuration and your performance requirements. If you have more than 128 terminals in your configuration, you may want to divide them among several port monitors. Go to the next section for information to create multiple port monitors to which you can then add the desired groups.

If each terminal in a group has a different definition label and terminal type, you must add the terminals one at a time. Use the instructions in the preceding section.

Follow this path through **sysadm** to add a group of identical terminals:

```
Device -> Port -> Terminal -> Add
```

The **sysadm** program steps you through a series of prompts. At each prompt, press New Line to accept the displayed default or enter a new value.

The first prompt asks for the tty device numbers of the terminals you want to add. Type **?** for a list of tty devices from which to choose.

```
Tty device(s): ? ↵
Choices are

     1  tty00
     2  tty01
     3  tty02
     4  tty03
     5  tty04
     6  tty05
     7  tty06
     8  tty07
     9  tty08
    10  tty09
    11  tty10
```

A tty device number is assigned automatically to each attached terminal in a hardware configuration when the system is booted. To find out the correspondence between the actual terminal and its tty number, see Appendix E. In this example, tty devices 0 through 10 are selected by menu choices 1 through 11. Select a range of tty devices using a comma-separated or dash-separated numeric list.

```
Tty device(s): 1-10 ↵
```

The entry 1-10 corresponds to tty00 through tty09.

The next prompt asks for the TTY definition label.

```
TTY Definition Label: [9600] ↵
```

The ttydefs labels appear as the first field in each line of the **/etc/ttydefs** file. For example, a typical asynchronous terminal label is **9600.**

The next prompt requests a value for the TERM variable in effect the first time a user logs in.

```
TERM Variable: vt100 ↵
```

The value must correspond to your terminal setting, which is a terminal definition in the **/usr/lib/terminfo** database. To see the entire list of **terminfo** definitions, execute this shell command:

```
# ls -RC /usr/lib/terminfo/* | more ↵
```

Your terminal hardware documentation should specify the correct terminal type. You may also refer to *Managing the DG/UX™ System* for information on valid Data General terminal TERM variable settings.

Next, supply a message to be displayed to users who attempt to log in to a disabled port.

```
Disabled response message: This terminal is disabled.
```

If you are adding a group of terminals, supply an appropriate message. Or, you may press New Line if you do not wish to supply a message.

Next, choose whether the port's service should be enabled or disabled.

```
Initial state: [ENABLED]  ↵
```

For a group of terminals, you should press New Line to accept the ENABLED default.

Finally, press New Line to accept the values that you have entered.

```
OK to perform operation? [yes] ↵
Adding default port monitor, ttymon1 ...
Port monitor ttymon1 has been added.
Terminals /dev/tty00-tty09 have been added.
```

The final messages inform you that the default port monitor **ttymon1** was successfully added to control the group of identical terminal whose tty line numbers are **tty00** through **tty09.**

The default port monitor **ttymon1** is created only if no **ttymon** port monitors have been defined on the system. If more than one **ttymon** port monitor exists on the system, you will be asked for the name of the desired port monitor.

Creating a Port Monitor to Manage Groups of Terminals

Port monitors are used to configure and control access to groups of terminals. The number of terminals you attach to a port monitor depends on your particular configuration and your performance requirements. If you have more than 128 terminals in your configuration, you may want to divide them among several port monitors.

Follow this path through **sysadm** to create a port monitor.

```
Device -> Port -> Port Monitor -> Add
```

This operation creates a directory structure for the new port monitor and adds an entry for it in the Service Access Controller's database. You assign a port service to each terminal that is controlled by the port monitor.

After you indicate that you want to add a port monitor, the **sysadm** program steps you through a series of prompts. At each prompt, press New Line to accept the displayed default or enter a new value.

The first prompt asks you to specify the type of port monitor you are adding.

```
Port monitor type: [ttymon] ↵
```

Select from two types of port monitors: **ttymon** or **listen**. If you are adding a port monitor for typical asynchronous tty lines, press New Line to accept the default **ttymon**. The **listen** port monitor is for Transport Layer Interface (TLI)-based network systems.

Next, provide a name for the port monitor.

```
Port monitor tag: ttymon2 ↵
```

The default port monitor **ttymon1** is created automatically when the first terminal is added. If no **ttymon** port monitor exists, you must create one using the port monitor menu.

Next, specify the command to start the port monitor and the version of the port monitor.

```
Command to start port monitor: [/usr/lib/saf/ttymon] ↵
Version number: [1] ↵
```

You can usually accept the default **/usr/lib/saf/ttymon** and release **1**, respectively.

Next, select the initial run state for the port monitor: **STARTED** or **STOPPED**.

```
Initial run state: [STARTED] ↵
```

The default choice, STARTED, means that the Service Access Controller starts the port monitor immediately and each time the Service Access Controller starts at boot time. If you select STOPPED, the port monitor does not become active until you start it explicitly using **sysadm** or the **sacadm**(1M) or **admportmonitor**(1M) commands. Accepting the default, STARTED, is recommended.

Next, specify the state of the port monitor when started: **ENABLED** or **DISABLED.**

```
Start state: [ENABLED] ↵
```

If you select ENABLED (the default), the port monitor begins management duties and accepts connection requests when started. If you select DISABLED, the port monitor monitors its ports but does not accept connection requests.

Next, specify the restart count, which is the number of times the Service Access Controller should attempt to restart the port monitor if it fails.

```
Restart count: (0-10) [0] ↵
```

If the port monitor cannot be restarted after this number of tries, it is placed in the FAILED state.

Next, specify the full pathname of the file that contains the port monitor configuration script, if any.

```
File name of configuration script: ↵
```

When the monitor starts, it copies the contents of this file to the port monitor configuration script file. The port monitor reads configuration files only when it starts. Changes to a script file made while the monitor is running will not be in effect until you restart the monitor. For more information on configuration scripts, see the **doconfig**(3N) manual page.

Next, you may supply a comment about the port monitor to be displayed when you select an operation that lists information about the port monitor.

```
Comment: Port monitor to control ttys on hallway12. ↵
```

Finally, confirm the selections that you have made.

```
OK to perform operation: [yes] ↵
Port monitor ttymon2 has been added.
```

The final message informs you that the named port monitor **ttymon2** was successfully added.

Adding a Modem

To add one or more modems, you must use the Port Services Menu rather than the Terminal Menu. The tty definition label for a modem typically begins with **M** and is followed by the modem speed. For example, a tty definition label for a 1200-baud modem line is **M1200.** Model labels are located in the first field in each line of the **/etc/ttydefs** file. A **ttymon** port monitor can monitor both modem and terminal lines.

Create an explicit port monitor to configure and control access to one or more modems. Then create a port service for the port monitor that manages the modems. Alternatively, you can use the same port monitor to control both terminal and modem lines.

Creating a Port Service for a Modem

After you have a port monitor for a modem, you must assign a port service for each modem to a port monitor.

Follow this path through **sysadm** to create a port service for a modem.

```
Device -> Port -> Port Service -> Add
```

The **sysadm** program steps you through a series of prompts. At each prompt, press New Line to accept the displayed default or enter a new value.

The first prompt asks for the name of the controlling port monitor to which you are assigning services.

```
Controlling port monitor for service: [tcp (listen)] ttymon3 ♪
```

Choose a **ttymon** type. Entering **?** lists the currently added port monitors.

Next, supply a unique descriptive tag that identifies the service.

```
Port service tag: tty06modem ♪
```

The tag can consist of up to 14 alphanumeric characters.

Next, supply the user login name of the owner of the service process.

```
Service Userid: [root] ♪
```

You can usually accept the **root** default. Any alternative username must begin with an uppercase or lowercase letter.

The next prompt asks whether you want to create a **utmp**(4) entry for the service whenever it is invoked.

```
Create utmp entry? [yes] ♪
```

If the service has a **utmp** entry, information about it will be available to commands such as **who**(1), **write**(1), and **login**(1). For normal login services you will always choose `yes`.

Next, supply the full path name of the file that contains the port service configuration script, if any.

```
File name of configuration script: ↵
```

When the service starts, it copies the contents of this file to the port service configuration script file. The port service reads configuration files only when it starts. Changes to a script file made while the service is running will not be in effect until you restart the service. For more information on configuration scripts, see the **doconfig**(3N) manual page.

Next, you optionally can supply a description about the port service.

```
Comment: Tty modem port. ↵
```

What you enter here is displayed when you select an operation that lists information about the port monitor.

Next, specify the state of the port when started: **ENABLED** or **DISABLED**.

```
Initial state: [ENABLED] ↵
```

For a modem, accept the ENABLED default.

The next prompt is for the port service version number.

```
Version number: [4] ↵
```

Accept the default value, **4**.

Next, specify full pathname of the port device (for example, **/dev/tty06**) that is associated with the service.

```
Path name of terminal device: /dev/tty06 ↵
```

The next prompt is for the label from the **/etc/ttydefs** file to use for setting the initial termio (terminal I/O) settings.

```
TTY Definition Label: [9600] M1200 ↵
```

The ttydefs labels appear as the first field in each line of the **ttydefs** file. For modem lines, use the label that starts with M and is followed by the modem speed. For example, use M1200 for a 1200-baud modem line. Type ? for help on choices.

Next, specify the full pathname of the service to be invoked following a successful modem connection.

```
Service command: [/usr/bin/login] ↵
```

We recommend that you choose the default, **/usr/bin/login**.

Next, select whether or not to force a hangup before initializing the line. You can usually select the default.

```
Hangup? [yes] ↵
```

Next, specify whether or not the **ttymon** port monitor should invoke the service without displaying a prompt as soon as it receives a carrier indication.

```
Connect on Carrier? [no] ↵
```

Answer `yes` only if you are sure of the baud rate and that no prompting is necessary.

Next, specify whether or not the port is used for calling out and for receiving incoming calls.

```
Bidirectional? [no] ↵
```

Making a port bidirectional allows outside users to connect to the port, and when the port is free, allows programs such as **uucico(1M)**, **cu(1)**, and **ct(1)** to use it for dialing out.

The next prompt asks for the number of New Line characters for which **ttymon** waits before sending out the prompt. This query is designed to accommodate a modem that sends out one or more lines of information before it is ready to receive data.

```
Wait-read value: [none] ↵
```

If 0, **ttymon** waits for any character before sending the prompt. If you select the default, **none**, **ttymon** sends a prompt without waiting for characters to be typed.

Next, specify the time interval in seconds that the port monitor is open and inactive before hanging up.

```
Timeout: (0-600) [0] 30 ↵
```

If the port monitor detects no typed characters over the line for the specified number of seconds, it terminates the connection. To specify that the line should never time out, specify 0 seconds. Values range from 0 to 600.

Next, enter the message that the port displays when it establishes a connection and places a port in the ENABLED state.

```
Prompt message: [login: ] ↵
```

Next, provide a comma-separated list of STREAMS modules that you want to push.

```
Modules to be pushed: ↵
```

After popping all modules already on the stream, these modules are pushed in the order in which you specify them. For more information, see the *Programmer's Guide: STREAMS*.

Next, provide a message displayed by the port monitor when a user attempts a connection on a disabled port.

```
Disabled response message: ♪
```

Use "\n" and "\t" to specify the New Line character and the Tab character, respectively.

Finally, confirm your entries.

```
OK to perform operation?  [yes] ♪
Port service modemservice has been added.
```

The final message informs you that the named port service **modemservice** was successfully added.

Listing Terminals, Port Monitors, and Port Services

Sysadm offers choices for listing terminals, port monitors, and port services and selected variables.

Listing Terminals

For a current list of terminals and selected variable settings, follow this path through **sysadm:**

```
Device -> Port -> Terminal -> List
```

A typical list follows.

```
/dev/tty00
        port monitor: ttymon1
       ttydefs label: 9600
               state: ENABLED
       TERM variable: vt100
        disabled msg: \nThis terminal is disabled.\n
```

Listing Port Monitors

For a current list of port monitors and selected variable settings, follow this path through **sysadm:**

```
Device -> Port -> Port Monitor -> List
```

A typical list follows.

```
PMTAG           PMTYPE         FLGS  RCNT STATUS     COMMAND
ttymon1         ttymon          -     3    ENABLED    /usr/lib/saf/ttymon #
ttymon2         ttymon          -     0    ENABLED    /usr/lib/saf/ttymon
Port monitor to control ttys on hallway 12.
```

Listing Port Services

For a current list of port services and selected variable settings, follow this path through **sysadm**:

```
Device -> Port -> Port Service -> List
```

A typical list follows.

```
PMTAG           PMTYPE          SVCTAG          FLGS ID       <PMSPECIFIC>
ttymon1         ttymon          tty00           u    root     /dev/tty00 -
- /usr/bin/login - 9600 - login:  Needed for modem in shipping dept.  #
```

Where To Go Next

For more information on determining tty lines for asynchronous ports, see Appendix C.

To set your terminals up for compatibility with DG/UX system standards, see *Managing the DG/UX™ System.*

For more information on tty device numbers (or device node names), see Appendix D.

To ensure that the correct terminal line controller name is listed in the system configuration file, which is the basis for the kernel, see Chapter 11 and Appendix C.

If you need to rebuild a kernel, you must also boot the newly built kernel. See Chapter 12.

End of Chapter

Chapter 6
Adding Printers

Before you can use a printer that is installed in your system (connected to your workstation or remotely to another computer), you must first configure it. While many **sysadm** operations are available for tuning the behavior of printers, this chapter discusses only adding and setting a default printer to establish basic operation. For more information on printer tuning, refer to *Managing the DG/UX™ System.*

It is assumed that you have completed hardware installation of the printer using the appropriate hardware documentation.

This chapter covers these tasks:

- Printer planning
- Ensuring that the kernel recognizes a local printer
- Determining a port's tty line number
- Adding a local printer
- Adding a remote printer
- Listing the status of added printers
- Adding a printer filter
- Setting a default printer

Collecting Printer Configuration Information

To configure local and remote printers, use the Printer Planning Worksheets located in Appendix E to record the following information about each printer.

printer name A printer name can contain these characters: `a-z A-Z 0-9 _`. The name should begin with a letter or digit and be no more than 14 characters long. Examples of printer names are **lp1**, **draft**, and **laser1**.

printer model Specifies the interface scripts that the LP service uses to drive the printer. Table 6-1 lists the default choices.

Table 6-1 Default Printer Models

Printer Models	Description
standard	Default that supports most line and laser printers
dg455x	DG Laser Printer (4557, 4558)
remshlp	Remote Line Printer (via TCP/IP)
termprinter	TermServer Line Printer

In most cases, **standard** is acceptable. If you choose another model, however, you may need to customize its interface scripts which are located at **/var/spool/lp/model**. See *Managing the DG/UX™ System.*

device file The port to which you connected the printer. For example, **/dev/tty01** is the device file for a printer on the port assigned **tty01**. If your computer has a dedicated lineprinter port, the pathname is **/dev/lp**.

printer type Specifies a printer's attributes. Device types are listed in the **terminfo** database located at **/usr/lib/terminfo/*x*/*y*** where *x* is a single-letter name for a directory containing the printer type file *y*. For example, the entry **/usr/lib/terminfo/P/PS** is the terminfo entry for Data General's Postscript printers. Printer type `unknown` is the default. Table 6-2 lists the valid printer types:

Table 6-2 Valid Printer Types

Printer Type	Description
printer	Generic line printer; 132 columns.
printer-80	Generic line printer; 80 columns.
PS	Generic PostScript printer.
PSR	PostScript printer; pages stack in reverse order.
lpr	Line printer; 132 columns.
lpr-80	line printer; 80 columns.
citoh	C.Itoh 8510 printer.
daisy	Daisy brand printers.
qume	Qume Sprint 11.
la100	DEC LA100 Printer.
ln03	DEC LN03 Laser Printer.
epson2500	Epson LQ-2500; low resolution.
epson2500-hi	Epson LQ-2500; high resolution.
epson2500-80	Epson LQ-2500; low resolution; 80-column.
epson2500-hi80	Epson LQ-2500; high resolution; 80-column.
hplaserjet	HP LaserJet I.
hplaserjetII	HP LaserJet II.
hplaserjetIII	HP LaserJet III.
ibmgraphics	IBM Graphics printer.
ibmcolor	IBM Color printer; 5:6 aspect ratio.
ibmcolor-1x1	IBM Color printer; 1:1 aspect ratio.
ibmxl	IBM Proprinter XL; 5:6 aspect ratio; low resolution.
ibmxl-1x1	IBM Proprinter XL in AGM; 1:1 aspect ratio; low resolution.
ibmxl-1x1-hi	IBM Proprinter XL in AGM; 1:1 aspect ratio; high resolution.
ibmxl-80	IBM PRoprinter XL; 5:6 aspect ratio; low resolution; 80-column.
ibmxl-1x1-80	IBM Proprinter XL in AGM; 1:1 aspect ratio; low resolution; 80-column.
ibmxl-1x1-hi80	IBM Proprinter XL in AGM; 1:1 aspect ratio; high resolution; 80-column.
7550	Hewlett Packard 7550A.

Ensuring that the Kernel Recognizes the Local Printer

When adding a local printer to an existing hardware configuration, ensure that the appropriate line controller to which the printer is attached is listed in the system configuration file. (Do not make entries in the system file for remote printers.) A printer can be attached to these types of line controllers:

duart()	Integrated duart terminal line controller (number 0)
duart(1)	Integrated duart terminal line controller (number 1)
syac	Systech terminal line controller (number 0)
syac(1)	Systech terminal line controller (number 1)
lp()	Integrated parallel line printer controller

A tty line number must be available for the printer being attached. Refer to the next section "Determining a Port's tty Line Number" and Appendix C for information on determining the printer's tty line number, and Appendix E for the tty line planning worksheets for an entry for the last tty line assigned.

View the system file to confirm the entry of the appropriate terminal line controller. The default name of the OS server or stand-alone system's system file is **system.aviion**, which is located at **/usr/src/uts/aviion/Build**. The default name of the OS client's system file is **system.diskless**. To list the names of devices that were configured automatically when you booted the installer system during the installation process, change to the directory containing the system file and use **more** to view the file.

```
# cd /usr/src/uts/aviion/Build ↵
# more system.aviion ↵
```

Figure 6-1 shows an excerpt of a typical OS server's system file.

```
sd(insc(0),*)    ## all SCSI disks on Integrated SCSI adapter
st(insc(0),*)    ## all SCSI tapes on Integrated SCSI adapter
duart()          ## integrated Duart terminal line controller
lp()             ## integrated parallel line printer controller
syac()           ## Systech terminal line controller
syac(1)          ## second Systech terminal line controller
syac(2)          ## third Systech terminal line controller
```

Figure 6-1 *Configured Devices in OS Server System File*

Figure 6-2 shows an excerpt of a typical OS client's system file.

```
Typical Hardware Devices for an OS Client:
#
# Note that your system can have an lp() controller or a second
# duart() but not both!  You'll have to determine which situation
# applies.
#
#
    kbd()           ## Workstation keyboard
    grfx()          ## Workstation graphics display
    duart(0)        ## Dual-line terminal controller (number 0)
#   duart(1)        ## Dual-line terminal controller (number 1)
    inen()          ## Integrated Ethernet controller
    lp()            ## Integrated parallel line printer controller
```

Figure 6-2 *Configured Devices in OS Server Client System File*

The terminal line controller to which the new printer is attached must be present in the system file before you add a local print using **sysadm**. If the appropriate terminal line controller does not appear in the system file, make the appropriate entry and rebuild and boot the kernel (see Chapters 11 and 12).

NOTE: As an alternative to viewing the system file, type the **sysdef**(1M) command for the current list of configured devices. Refer to Chapter 11 for information on using **sysdef**(1M) to check the kernel.

Determining a Port's tty Line Number

The DG/UX system automatically assigns a tty line number to each attached printer serial port in the hardware configuration when the system boots. A tty line number takes the form:

tty*x*

where:

x is a decimal number containing at least two digits that are sequentially assigned. As examples, **tty00** refers to the first port, **tty01** the second port, and so on.

If you have a printer attached directly to the workstation's parallel printer port, the **/dev/lp** entry is created each time the system boots. For serial printers, a file is created bearing the name of the tty line number in the **/dev** directory each time the system boots.

Refer to Appendix C for information on determining tty lines for asynchronous ports.

Adding a Local Printer

A local printer is one that is physically attached to your computer. You must supply information about a local printer before it is operational.

Follow this path through **sysadm**:

```
Device -> Printer -> Devices -> Add
```

The **sysadm** program steps you through a series of prompts. At each prompt, press New Line to accept the displayed default or enter a new value.

The first prompt asks for the printer name.

```
Printer name: laser ↵
```

Enter the printer name that you recorded on the Printer Planning Worksheet in Appendix E.

The next two queries ask if the printer should be immediately enabled and ready to accept print requests.

```
Enable? [yes] ↵
Accept? [yes] ↵
```

Enabling permits the LP scheduler to submit waiting print jobs to the printer. Accepting allows users to submit their print jobs to the queue. Disabling the printer while accepting jobs may be desirable if the printer is temporarily malfunctioning.

Since you are adding a local printer, you answer **yes** to the next prompt.

```
Local printer? [no] yes ↵
```

Next, enter the pathname of the physical printer device recorded on the Printer Planning Worksheet.

```
Device: /dev/tty01 ↵
```

When asked about printer type, stty options, and other print options, you can usually accept default values. They characterize the I/O behavior of the printer.

```
Printer type: [printer-80] ↵
Stty Options: ↵
Other Print Options: ↵
```

Stty option requirements are generally handled by the interface program automatically. However, consult your device's release notice for the possibility of other stty options.

Rather than accept the default value of `Other Print Options:`, you can specify values as space-separated entries on a single line. Table 6-3 lists allowable options and default values:

Table 6-3 Print Options and Default Values

Option	Default Value
length=*lines-per-page*	63
width=*characters-per-line*	40
cpi=*characters-per-inch*	10
lpi=*lines-per-inch*	6
nobanner or **banner**	Prevents or allows use of **-o nobanner** option (to inhibit printing of banner page) with the **lp**(1) command. If **nobanner** is selected, use of **-o nobanner** is prohibited.

An example of how to change default values follows:

length=50 width=50 cpi=11 lpi=4 nobanner

NOTE: These printer-specific defaults are derived from the **/usr/lib/terminfo/P/PS** database.

The next prompt asks for the input types allowed:

```
Input types allowed: [simple] ↵
```

The default, `simple`, refers to an ASCII file. Examples of other legal values for input types are as follows:

troff	Device-independent output from the **troff** text-formatting processor.
tex	DVI format files.
plot	Plotting instructions for Tektronix displays and devices.
raster	Raster bitmap format for Varian raster devices.
cif	Output of BSD **cifpbt** processor.
fortran	ASA carriage control format.
postscript	PostScript language.

You can specify more than one input file type by entering more than one of these values separated by spaces on a single line. For example,

troff tex postscript

You may need to know the input type when you submit a print job to a printer with the **lp**(1) command and the **T** option. For example,

 # lp -T postscript bigfile ↵

The next prompt asks whether the printer is attached to a terminal used for logging into the system.

```
Device is also a login terminal: [no] ↵
```

You can usually accept the `no` default. Otherwise, answer `yes` and the printer is not enabled but is ready to accept input immediately.

Next, you are prompted for how the printer recovers from a fault.

```
Fault Recovery: [continue] ↵
```

Choices are as follows:

continue Before resuming printing, waits for the fault to clear and reprints the current page.

beginning Waits for the fault to clear, and immediately prints the entire job over again.

wait Stops the job and disables the printer. You have to explicitly enable the printer to resume printing.

The next query asks you how you want to be notified of possible printer malfunctions.

```
Alert Messages: [mail] ↵
```

Choices are as follows:

mail Invokes the electronic **mailx**(1) program, which issues a mail message to the system administrator.

write Sends a brief message to the terminal at which the system administrator is logged in.

quiet Ignores the problem.

none Does not send any alert messages for printer malfunctions.

The alert interval specifies the number of minutes apart an alert message about a printer is sent to the system administrator.

```
Alert Interval: [0] ↵
```

0 means only one alert per printer fault.

Next, you can restrict printer access to specific user logins.

```
Users to Allow: [all] ↵
```

The `all` default allows all users to access the printer. You can specify those who can access the printer or those who are denied access. If you accept the default `all`, you then can use the deny prompt to list exceptions.

```
Users to Deny:  none ↵
```

The deny list is ignored unless the allow list is set to `all`. If you list specific users in the allow list, then only those listed will be allowed to use the printer. Specify entries in a space-separated list. The choice `none` is valid for the deny list only.

The next query allows you to specify the types of printer forms to be used on the printer.

```
Forms to Allow: [none]
```

The default `all` allows all forms to be used. You can either list explicitly the forms to be used, or you can specify those that cannot be used. If you allow all types, the deny prompt is presented to which you can list exceptions.

```
Forms to Deny: ↵
```

Here, specify entries in a space-separated list. The choice `none` means that no forms may be used on the printer.

Next, you are asked whether you want to copy an existing printer's interface program.

```
Copy an existing printer's interface program? [no] ↵
```

If you want to use a model interface program, accept the default **no**. If you enter **yes**, you are prompted for the name of the program. Type **?** for a list of existing printer interface programs, which are located in **/etc/lp/interfaces**.

You are then prompted for the desired model.

```
Model [standard] ↵
```

Consult your Printer Planning Worksheet for this information.

Next, supply a printer description, which you can use to identify the name of the system administrator, her or his phone number, or the location of the printer.

```
Printer Description: Draft laser printer on hallway 21.
```

The note accompanys the printer name whenever you list the status of added printers.

Finally, you are prompted to confirm your choices:

```
OK to perform operation? [yes] ↵
Printer laser has been added.
Printer laser has been enabled.
Printer laser has been set to accept requests.
```

After answering these questions, your local printer is added and ready for operation.

Adding a Remote Printer

A remote printer is one remotely connected to another computer that you can access via an Ethernet LAN.

To add a remote printer, you must perform these operations:

- Determine whether the remote printer is running on a 5.4 version DG/UX (or BSD or SunOS) or a pre-5.4 release of the DG/UX system.
- Add the remote host to share printer resources.
- Add a printer on the remote host.

Is the Remote Printer Running on a 5.4 or pre-5.4 Version of DG/UX?

When establishing access to a remote printer, you need to know whether the host (to which the remote printer is attached) is running the 5.4 or pre-5.4 version of the DG/UX system. Pre-5.4 versions of DG/UX use **remsh**(1) to transfer print requests to a remote host whereas the 5.4 version uses the V.4 AT&T System **lp**(1) networking functions to perform remote print requests. Both BSD and SunOS also use **lp**(1) networking functions. The **remsh**(1) and **lp**(1) commands have different requirements for establishing remote printer access.

NOTE: In this chapter, procedures that apply to DG/UX 5.4 also apply to BSD and SunOS.

If the remote printer uses the 5.4 version of DG/UX, skip this section and go to the section "Adding a Remote Host to Share Printer Resources." Otherwise, continue reading the next section.

Enabling Remote Printer Access on Pre-5.4 DG/UX Systems

You can enable remote login permissions through entries to one of two files:

/.rhosts The user's **.rhosts** file (located in the user's home directory) contains a list of the names of the remote hosts to which the user can log in. Thus, logins to remote hosts are restricted to particular users. An entry in the user's **/.rhosts** file assures a higher level of security than the **/etc/hosts.equiv** file.

/etc/hosts.equiv The host's **/etc/hosts.equiv** file contains a list of the names of remote hosts to which the local host can connect. No prompt for a user password is issued when the user issues the **rlogin**(1C) or **remsh**(1C) request to a remote host. Thus, logins to remote hosts are restricted to particular hosts. An entry in a host's **/etc/hosts.equiv** file assures a lower level of security than the user's **/.rhosts** file.

Select the preferred method for enabling remote printer printer access on a pre-5.4 DG/UX system.

Assigning User Access to Remote Hosts

Create an **/.rhosts** file in the user's home directory or, if it already exists, edit the file to insert the name of the remote host to which you want to connect. The file format contains a list of names of remote hosts; one host name per line starting in the first column. An example follows:

```
# more .rhosts ↵
gyramax
goose
mad-max
matilda
```

The user on this host can remotely access the remote hosts listed in this file.

The administrator of the remote host to which you desire access must mount your home directory before your entries in your **/.rhosts** file take effect. Refer to Chapter 3 for instructions on adding a remote file system.

Assigning Host Access to Remote Hosts

To allow host access to remote hosts, you will create or edit the **/etc/host.equiv** file, which is also known as the trusted hosts database. A remote user is considered local if the name of its host appears in this file. A line which contains a host name means that anyone logging in from that host is trusted.

To make an entry in the **/etc/host.equiv** file, follow this path through **sysadm** and respond to the prompts:

```
Networking -> TCP/IP -> Trusted Hosts -> Add
```

The first prompt asks for the name of the host to which the specified user login can remotely access.

```
Host Name: max-server ↵
```

You are next asked to specify the authorized user who has login privileges.

```
User Name: [all] johnson ↵
```

Finally, press New Line to accept the values that you have entered.

```
OK to perform operation?  [yes] ↵
johnson on max-server has been added.
```

The named user login on the named remote system in the **/etc/host.equiv** file can access the host using the remote shell (**rsh** or **remsh**) or the remote login (**rlogin**). Refer to the **rsh**(1), **remsh**(1), and **rlogin**(1) manual pages for more information.

Next, you can add remote hosts to share printer resources.

Adding a Remote Host to Share Printer Resources

NOTE: The procedures in this section are not required for remote hosts running a pre-5.4 version of the DG/UX system.

Sharing print resources means that your computer can use remote printers, and remote computers can use your locally attached printers. You must add a remote host with which to share printer resources before you can add a printer on that host. Perform this procedure at your local host and at the remote host with whom you wish to share printer resources.

Follow this path through **sysadm**:

```
Device -> Printer -> Systems -> Add
```

The **sysadm** program steps you through a series of prompts. At each prompt, press New Line to accept the displayed default or enter a new value.

The first prompt asks you to specify the names of systems with which to share printer services.

```
System name: goose  gyramax ↵
```

Enter one or more host names, the names separated by spaces.

Next, choose the printer scheduler type used by the remote system.

```
Scheduler Type: [s5] ↵
```

Valid choices are `s5` for AT&T System V (the default) and `bsd` for BSD and SunOS.

Next, specify the number of minutes that the connection with the remote system may remain idle before terminating.

```
Connection Timeout: [n] ↵
```

The default `n` specifies no timeout. To specify that the connection should terminate as soon as it becomes idle, type `0`.

Next, specify the number of minutes to wait, following a disconnection of service with a remote system, before attempting to re-establish the connection.

```
Retry Period: [10] ↵
```

Type **no** if you do not want the print service to retry the connection. Instead, it retries when it receives a job request to send to the remote system. Type **0** if you want the print service to retry the connection immediately after any disconnection. Press New Line to accept the `10` default to retry after ten minutes.

Next, enter a descriptive note about the remote system, such as the name of its system administrator, a phone number, or location.

```
Comment: System administrator is Pat Silva. ↵
```

Do not use single quotation marks in the note. The note appears when you list information on the system using the list operation.

Finally, confirm your entries.

```
OK to perform operation?  [yes] ↵
"goose" has been added.
"gyramax" has been added.
```

Adding a Printer on a Remote Host

NOTE: The procedures in this section are required for remote hosts running either 5.4 or a pre-5.4 version of the DG/UX system.

Follow this path through **sysadm**:

```
Device -> Printer -> Devices -> Add
```

The **sysadm** program steps you through a series of prompts. At each prompt, press New Line to accept the displayed default or enter a new value.

At the first prompt, enter the name you recorded on the Printer Planning Worksheet.

```
Printer name:  rmt-laser ↵
```

Next, press New Line to enable the LP scheduler to submit waiting jobs to the printer.

```
Enable? [yes] ↵
```

Next, press New Line to allow users to submit their print jobs to the queue.

```
Accept? [yes] ↵
```

You may want to disable the printer while accepting jobs if the printer is temporarily malfunctioning.

Next, press New Line to specify that you are adding a remote printer.

```
Local printer? [no] ↵
```

Next, supply the name of the remote host to which the printer is attached. You must have added this host through `Device -> Printer -> Systems -> Add`.

```
Remote Host: gyramax ↵
```

Next, enter the name of the remote printer.

```
Remote Printer Name: post ↵
```

Next, identify whether the remote printer is configured in a system that is using a pre-5.4 release of the DG/UX system.

```
The remote scheduler is PRE-DG/UX 5.4: [no]
```

A pre-5.4 release of the print scheduler uses the **remsh**(1) command to transfer your print request to the remote system whereas the 5.4 release of the scheduler uses the V.4 AT&T System **lp**(1) networking functions to perform remote print requests.

NOTE: BSD and SunOS operating systems do not use a pre-5.4 release of the print scheduler.

If the remote scheduler is a 5.4 release, you can still answer the preceding prompt with **yes** (for a pre-5.4 scheduler). Answering **yes** in this case, however, will restrict the remote printing features offered by the new 5.4 **lp** subsystem. For example, you will be unable to cancel a remote printer job or check its status. Furthermore, you must make entries in either your **/.rhosts** file or the host's **/etc/hosts.equiv** file (see "Enabling Remote Printer Access on Pre-5.4 DG/UX Systems" for this information). Also, you sacrifice remote printing speed.

If you specify that the remote system uses a pre-5.4 release of the DG/UX system, the following prompt appears:

```
The remote scheduler is PRE-DG/UX 5.4: [no] yes ↵
Input types allowed: [simple] ↵
```

The default, `simple`, refers to an ASCII file. Examples of other legal values for input types are as follows:

troff	Device-independent output from the **troff** text-formatting processor.
tex	DVI format files.
plot	Plotting instructions for Tektronix displays and devices.
raster	Raster bitmap format for Varian raster devices.
cif	Output of BSD **cifpbt** processor.
fortran	ASA carriage control format.
postcript	PostScript language.

You can specify more than one input file type by entering more than one of these values separated by spaces on a single line. For example,

troff tex postscript

You may need to know the input type when you submit a print job to a printer with the **lp**(1) command and the **T** option.

Next, you are prompted for how the printer recovers from a fault.

```
Fault Recovery: [continue] ↵
```

Choices are as follows:

continue Before resuming printing, waits for the fault to clear and reprints the current page.

beginning Waits for the fault to clear, and immediately prints the entire job over again.

wait Stops the job and disables the printer. You have to explicitly enable the printer to resume printing.

Next, specify you how you want to be notified of printer malfunctions.

```
Alert Messages: [mail] ↵
```

Choices are as follows:

mail Invokes the electronic **mailx**(1) program, which issues a mail message to the system administrator.

write Sends a brief message to the terminal at which the system administrator is logged in.

quiet Ignores the current problem. It is a useful designation for a known malfunction condition.

none Does not send any alert messages for printer malfunctions.

Next, specify the number of minutes apart an alert message about a printer is sent to the system administrator.

```
Alert Interval: [0] ↵
```

0 means only one alert per printer fault.

Next, you are asked whether you want to copy an existing printer's interface program.

```
Copy an existing printer's interface program? [no] ↵
```

If you want to use a model interface program, accept the default **no**. If you enter **yes**, you are prompted for the name of the program. Type **?** for a list of existing printer interface programs, which are located in **/etc/lp/interfaces**.

Next, supply a printer description, which you can use to identify the name of the system administrator, her or his phone number, or the location of the printer.

```
Printer Description: Laser printer on hallway 22.
```

The note accompanys the printer name whenever you list the status of added printers.

Finally, confirm your choices.

```
OK to perform operation? [yes] ↵
Printer laser has been added.
Printer laser has been enabled.
Printer laser has been set to accept requests.
```

After answering these questions, your remote printer is added and ready for operation.

If you specify that the remote system uses a 5.4 release of the DG/UX system, the following prompt appears:

```
The remote scheduler is PRE-DG/UX 5.4: [no] ↵
Users to Allow: [all] ↵
```

The `all` default allows all users to access the printer. You can specify those who can access the printer or those who are denied access. If you accept the default `all`, you then can use the deny prompt to list exceptions.

```
Users to Deny:  none ↵
```

The deny list is ignored unless the allow list is set to `all`. If you list specific users in the allow list, then only those listed will be allowed to use the printer. Specify entries in a space-separated list. The choice `none` is valid for the deny list only.

The next query allows you to specify the types of printer forms to be used on the printer.

```
Forms to Allow: [none] ↵
```

The **all** answer allows all forms to be used. You can either list explicitly the forms to be used, or you can specify those that cannot be used. If you allow all types, the deny prompt is presented to which you can list exceptions.

```
Forms to Deny: ↵
```

Next, supply a printer description, which you can use to identify the name of the system administrator, her or his phone number, or the location of the printer.

```
Printer Description: Draft laser printer on hallway 21.
```

The note accompanys the printer name when you list the status of added printers.

Finally, you are asked to confirm your entries.

```
OK to perform operation? [yes] ↵
Printer laser has been added.
Printer laser has been enabled.
Printer laser has been set to accept requests.
```

After answering these questions, your remote printer is added and ready for operation.

Listing the Status of Added Printers

After you have added printers, you can check their status by following this path through **sysadm**.

```
Device -> Printer -> List
```

The **sysadm** program asks you to specify the printers for which you want information. Specify **all** to get a full report. Accept the defaults for the other two prompts that appear.

```
Printer name(s): all ↵
List Type: [setup] ↵
OK to perform operation?  [yes] ↵
```

A typical report follows

```
scheduler is running
system default destination: laser
device for laser: /dev/tty03
laser accepting requests since Mon May 13 11:34:03 EDT 1991
printer laser is idle. enabled since Mon May 13 14:50:14 EDT 1991.
available.
```

You can also list a report of a single configured printer and its characteristics.

```
Device -> Printer -> Devices -> List

Printer name(s): laser ↵
List Type: [setup] ↵
OK to perform operation: [yes] ↵
```

The default list type, `setup`, displays information about your printer's configuration parameters.

```
Laser is idle. enabled since Mon May 13 15:11:54 EDT 1991. available.
        Form mounted:
        Content types: postscript
        Printer types: unknown
        Description:
        Connection: laser@starship
        Interface: /usr/lib/lp/model/remshlp
        On fault: alert with "mail johnson" once
        After fault: continue
        Users allowed:
                (all)
        Forms allowed:
                (none)
        Banner required
        Character sets:
                (none)
        Default pitch:
        Default page size:
        Default port settings:
```

Alternatively, specify the list type `state` to display the printer name, whether the printer is enabled or accepting, and the number of requests waiting. For example,

```
Printer Name         Enabled?  Accept?      Requests
------------         --------  -------      --------
laser                enabled   accepting           0
```

Adding a Printer Filter

A printing filter converts a file to the control language of the specified printer and passes the file to the communications program that sends it to the printer. If, for example, you use the PostScript language as an input file type, you will need to add a filter before the printer is functional.

Follow this path through **sysadm**.

```
Device -> Printer -> Filter -> Add
```

The **sysadm** program steps you through a series of prompts. At each prompt, press New Line to accept the displayed default or enter a new value.

First, you are asked to specify a filter name.

```
Filter name: print1 ↵
```

Supply an appropriate name.

Next, you are prompted to supply the name of a filter whose description you wish to copy.

```
Filter to Copy: [default] ↵
```

Choose one of the following values:

1 postprint
2 download.fd
3 dpost.fd
4 postdaisy.fd
5 postdmd.fd
6 postio.fd
7 postior.fd
8 postmd.fd
9 postplot.fd
10 postprint.fd
11 postreverse.fd
12 posttek.fd
13 default

Next, supply the input file type, which specifies the types of content that the filter can accept.

```
Input Types: [any] simple ↵
```

You can usually choose the default value, **any**. Examples of other legal values for input types are as follows:

troff	Device-independent output from the **troff** text-formatting processor.
tex	DVI format files.
plot	Plotting instructions for Tektronix displays and devices.
raster	Raster bitmap format for Varian raster devices.
cif	Output of BSD **cifpbt** processor.
fortran	ASA carriage control format.
postcript	PostScript language.
simple	ASCII file.

The input type should match either the output type of another filter or an input type that users specify on the **lp** command line with the **-T** option. For example

```
# lp -T postscript bigfile ↵
```

After you specify the input type, specify the output type.

```
Output Types: [any] postscript ↵
```

The output type specifies the content that the filter produces. The input and output types should match.

For both input and output types, you can specify more than one value on a single line separated by spaces. For example,

```
troff tex postscript
```

Next, specify the types of printer that can print output from this filter.

```
Printer Types: [any] ↵
```

Check Table 6-2 for a list of printer types. There may be printers that are compatible with the filter's output type, but for some reason are undesirable as output devices for the filter. Therefore, you should explicitly name the printers that can be used as output devices for the filter; any configured printer is allowed.

Next, specify the printer name.

```
Printer name: [any] ↵
```

You can type a space-separated list of valid printer names.

Next, indicate whether you want to change the recommended options needed by this filter.

```
Modify Options? [no] ↵
```

Next, designate the filter speed.

```
Filter speed: [slow] ↵
```

You can designate a filter as fast or slow. When a print request requires a fast filter, the print service assigns a printer to the request at the same time it starts a filter. If a filter is slow, however, the printer is unnecessarily occupied, waiting for the filter to pass through its output. A slow filter, on the other hand, makes more effective use of a waiting printer. It executes slow filters in the background and does not tie up the printer while it executes fast filters in the foreground as the printer actively prints.

Next, specify the command line the printer service uses to invoke the filter.

```
Program: /usr/lib/lp/postscript/postprint ↵
```

Your response should include the full filter pathname and any command-line options. Refer to *Managing the DG/UX™ System* for full details on specifying a command line.

Finally, confirm your choices.

```
OK to perform operation?  [yes] ↵
```

After answering these questions, your filter is added and ready for operation.

Setting a Default Printer

Follow this path through **sysadm** to set a default printer:

```
Device -> Printer -> Devices -> Default
```

With a defined default printer, you do not need to use the **-d** option with the **lp**(1) command unless you want to explicitly print on a non-default printer.

A sample dialogue follows.

```
Printer name: [laser] ↵
OK to perform operation?  [yes] ↵
The current default printer is laser.
```

The last printer you configured will be automatically supplied as the default printer name.

End of Chapter

Chapter 7
Loading and Setting Up Software Packages

Adding a software package (obtained from either Data General or a third-party vendor) is accomplished in two steps: loading from the release tape and setting up the software to run with the DG/UX system. If you received a Data General release tape and postponed loading and setting up of any of the bundled packages (such as the DG/UX X Window system, TCP/IP, ONC, or NFS), you can use these procedures to do so now.

Also, if you are loading or setting up the ONC/NFS package, there are several more steps required for setting up your AViiON computer as an NIS master or NIS server.

Before you load and set up a package, read its release notice. You must create the required logical disks and their file systems, and mount them on the DG/UX file system before you can load a package (see Chapter 3).

Loading a Software Package from the Release Tape

These instructions assume that the package conforms to Data General package specifications. A Data General package (such as TCP/IP) can be placed in the normal DG/UX release area. Most vendors' packages will be loaded at **/usr/opt/***package-name*, where *package-name* is the name of the package. However, the package's release notice gives the definitive location.

If you are installing packages for the first time, you may be unfamiliar with how to insert a release tape in a tape drive. If so, use these procedures:

1. Make sure that your release tape is write-protected. Orient the arrow to the "SAFE" position.

2. Slide the load/eject latch on the tape drive as far as it will go. The direction in which you slide the latch depends on the type of tape drive you are using; usually to the left or up position.

3. Depending on the type of drive, the metal plate on the release tape should face either down or left.

4. Gently push the tape into the drive until it is secure. Then slide the latch in the opposite direction to lock it in place. The drive makes a whirring sound as the read/write heads engage the beginning of the tape.

To load a package, follow this path through **sysadm** :

```
Software-> Package-> Load
```

The first two prompts query you for the tape device and whether or not the tape device is connected and a cartridge tape is correctly installed in its drive.

```
Tape Device: [/dev/rmt/0] ↵
Is /dev/rmt/0 ready? [yes] ↵
```

Ensure that the tape device name is correct. You can usually accept the default response by pressing New Line.

You are next prompted for the package name. In this example, type ? for a list of the packages on the release tape.

```
Package Name(s): [all] ? ↵
Choices are

1 all
2 X11
3 aview
4 nfs
5 onc
6 tcpip
```

You can select one or more packages for loading. To specify more than one package, type each package's name followed by a comma (,). In this example, only TCP/IP is selected for loading.

```
Package Name(s) [all] tcpip
OK to perform operation? [yes] ↵

Loading the package tcpip ......

Finished loading packages.
```

To make explicit selections, however, enter a list of the names of the desired files separated by commas or spaces. For example:

```
Package Names(s) [all] X11, aview, tcpip ↵
```

You can also select packages to load using numbers or a range of numbers. For instance, a quicker way of choosing the same packages would be:

```
Package Names(s) [all] 2-3,6 ↵
```

The exact messages written to your screen depend on the particular package being loaded. You will see the green indicator light and you will hear the tape drive advance and rewind, which are indications that the tape is being read. You will be notified of the completion of the package load.

Setting Up Packages

After loading, you must set up software packages using **sysadm**. Setup varies according to the type of software package: If you are setting up the DG/UX ONC/NFS and TCP/IP packages, you must collect networking information about them beforehand. If you are setting up packages at an OS server, you need information about the network. (Instructions for package loading and setup for an OS client are in Chapter 8.)

OS Server Network Planning

This section is applicable only if you are setting up the DG/UX ONC and TCP/IP packages on the OS server. An OS Server Network Planning Worksheet is provided in Appendix E to help you prepare for loading the ONC and TCP/IP packages. It lists network parameters for which you need to supply values. Completing the worksheets before you begin the package setup procedures will speed up the customization process considerably.

If you plan to install the ONC/NFS package, you must supply one parameter during the installation process.

NIS domain name	Directory containing NIS maps located in **/etc/yp/***domain-name*. Machines having this directory as their default NIS domain share the data found in its maps.

Refer to *Managing ONC/NFS and Its Facilities on the DG/UX™ System* for background information on ONC parameters.

When you install the TCP/IP package, you must supply at most five parameters:

Host name	A unique name you assign to your AViiON hardware that can contain up to 31 alphabetic and numeric characters. However, you are advised to keep the names short. Host names that relate to the use or location of the system are particularly helpful in networked environments where hosts may share file systems. Examples of host names are **fred**, **jamaica**, and **writer_doc**. Do not use the capitalized names **MY_HOST** or **PRIMARY**; these names are reserved by the system.
Internet address	The Internet address of the host being set up is provided by the network system administrator. An example of an Internet address is **128.223.2.1**. In this example, **128.223** is to the network number, **2.1** is the host number. The dots are field separators.
Do you subnet?	Subnetting allows you to combine multiple physical networks into a single logical network. You must answer "yes" or "no" to the query about subnetting.
Network mask	A hexadecimal bit pattern that specifies the number of bits used to identify the network part of an Internet address. **0xff000000** and **0xffff0000** are examples of network masks. A network mask is required only when your network is subnetted.

Controller device name — Device used to connect an AViiON computer to the local area network. Valid names take the form: *controller-typecontroller-num*, where *controller-type* can be **inen** (integrated Ethernet controller), **hken** (Interphase VME Ethernet controller), or **dgen** (Data General second generation integrated Ethernet controller). The *controller-num* for controller type **inen** can be only **0**; valid controller numbers for **hken** are **0-7**; for **dgen**, **0-1**. Valid examples are **inen0**, **hken3**, or **dgen0**. You are queried about the controller device only if your system has more than one network interface.

Refer to *Managing TCP/IP on the DG/UX System* for more information.

To set up a software package, follow this path through **sysadm**:

```
Software -> Package -> Set up
```

The next two sections show typical dialogues for setting up all DG/UX packages and only one package, TCP/IP.

Setting Up All DG/UX Software Packages

Figure 7-1 provides a typical dialogue for setting up all DG/UX software packages for an OS client. The same messages apply to an OS server; the difference being that an OS client host name will be seen instead of an OS client server name. In the following examples, **mrkean** is the OS client host name.

```
Client(s) to set up: [none] mrkean ↵
Package Name(s):  [all] ? ↵

Select 'all' to set up package(s) in the root file systems of all
clients attached to the selected release area.  Select individual
clients if you want to set up the package only for this client.

Select nothing or 'none' to set up the package only in your own root
file system and not in the root file system of any clients.  Typically,
clients set up their own packages.

Choices are

1 all
2 X11
3 X11.lg
4 dgux
5 aview
6 nfs
7 onc
8 tcpip

Package Name(s) [all] ↵
OK to perform operation? [yes] ↵
```

Figure 7-1 *Setting Up All DG/UX Software Packages*

The choices listed will depend on the packages that were loaded and set up on the OS server.

The next stage of initialization is **dgux** package setup, which takes about three minutes and produces these messages:

```
    Setting up dgux in mrkean.root.

    Setting up DG/UX...

    Initializing DG/UX prototype files.......................

    Creating DG/UX run level links.....................

    Initializing sysadm(1M)..............

    Continuing setup of DG/UX...........

    NOTE:  See /srv/release/PRIMARY/root/mrkean/var/setup.d/log/dgux.root
           for a detailed account of the root setup of DG/UX.
```

Setting Up X11

The following messages appear when setting up **X11**.

```
Setting up X11 in mrkean root.
Package X11 has been successfully set up in mrkean root.
Package setup for X11 is complete.
```

Setting Up AView

The following messages appear when setting up **AView**.

```
Setting up aview in mrkean root.
Package aview has been successfully set up in mrkean root.
Package setup for aview is complete.
```

Setting Up NFS

The following messages appear when setting up **NFS**.

```
Setting up nfs in mrkean root.

    Setting up NFS in mrkean root...........

    Creating NFS run level links.........

    Initializing NFS prototype files.........

    NOTE:  See /srv/release/PRIMARY/root/mrkean/var/setup.d
```

```
        /log/nfs.root for a detailed account of the root
        setup of NFS.

Package nfs has been successfully set up in mrkean root.
Package setup for nfs is complete.
```

Setting Up ONC

When setting up ONC, you supply the name of the NIS domain (See the OS Server Network Planning Worksheets that you completed in Appendix E for this information.)

```
Setting up onc in mrkean root.

    Setting up ONC in mrkean root.....

    Creating ONC run level links.....

    Initializing ONC prototype files.....

    Enter the NIS Domain name []: work-net ♪
    [work-net] Correct? [yes]:

    NOTE:  This host will first run as an NIS client.

    NOTE:  See /srv/release/PRIMARY/root/mrkean/var/setup.d
           /log/onc.root for a detailed account of the root
           setup of ONC.

Package onc has been successfully set up in mrkean root.
Package setup for onc is complete.
```

Setting Up TCP/IP

When setting up TCP/IP, you must respond to several prompts. Check your OS Server Network Planning Worksheets that you completed in Appendix E for this information. Also, refer to *Managing TCP/IP on the DG/UX™ System* for background information. The setup script probes the hardware configuration for network controller devices.

```
Setup package tcpip in mrkean root

    Setting up tcpip...
```

The dialogue for setting up the TCP/IP package on an OS client follows:

```
    The following queries refer to the primary network interface.

    Enter host name: mrkean ♪
    [mrkean] Correct? [yes] ♪
    Enter host Internet address: 128.222.8.60 ♪
    [128.222.8.60] Correct? [yes] ♪
```

```
Is your local network subnetted? [no] yes ♪
Enter the network mask: 0xffffff00 ♪
[0xffffff00] Correct? [yes] ♪

NOTE:  Using inen0 as the primary network interface controller.

Setting up tcpip ............

NOTE:  See /var/setup.d/log/tcpip.root file for a
       verbose description of the package setup for root.

Package tcpip has been successfully set up in mrkean root.

Package setup for tcpip is complete.
```

The setup procedures for TCP/IP and other packages is complete.

The superuser prompt (#) for input is displayed on your screen after all packages are set up.

Setting Up a Single Software Package

Figure 7-2 provides a typical dialogue for setting up only one DG/UX software package for a stand-alone system.

```
Set up package?  [yes] ↵
Package Name(s):  [all] ? ↵

Choices are

1 all
2 X11
3 aview
4 nfs
5 onc
6 tcpip

Package Name(s) [all] tcpip ↵
OK to perform operation? [yes] ↵
Release Area? [PRIMARY] ↵
Setup package tcpip in usr? [yes] ↵
Processing setup scripts for package tcpip

Setup package tcpip in usr.

    NOTE:  The /usr/bin/rsh command now executes as a remote
           shell(remsh), not as a restricted shell(restsh).
    NOTE:  See /usr/root.proto/var/setup.d/log/tcpip.usr file
           for a verbose description of the package setup process.

Setup package tcpip in MY_HOST root.

    Setting up tcpip...

    Enter host name: mrkean ↵
    [mrkean] Correct? [yes] ↵
    Enter host Internet address: 128.222.8.60 ↵
    [128.222.8.60] Correct? [yes] ↵
    Is your local network subnetted? [no] yes ↵
    Enter the network mask: 0xffffff00 ↵
    [0xffffff00] Correct? [yes] ↵

    NOTE:  Using inen0 as the primary network interface controller.

    Package setup for tcpip has completed.
```

Figure 7-2 *Typical Messages for Single Software Package Setup*

The prompts that appear on your screen depend on the particular package being set up. When prompted for the name of the package, type ? to list the packages in the tape's table of contents. Pressing New Line accepts the default.

The package's release notice gives the information you need to set up the package.

Completing NIS Package Setup

You need to complete NIS setup only if your AViiON computer is connected to a network or you intend to add OS clients to an OS server. For the OS client-server configuration, by default, your AViiON computer is assumed to be an NIS client. To alter the default, follow the instructions in this section.

To set up your AViiON computer as the first (or only) NIS master server in your NIS domain, see *Managing ONC/NFS and Its Facilities on the DG/UX™ System.*

The next two sections give instructions for setting up your AViiON computer to perform one of the following roles.

- NIS Master
- NIS Server

Choose the appropriate section.

Setting Up Your AViiON Computer as an NIS Master

Perform these procedures to set up your AViiON system as an NIS master in an NIS domain that already has a master.

1. Edit **/etc/nfs.params** and set the value of **ypserv_START**.

```
ypserv_START="MASTER"
```

2. In **/etc/nfs.params**, set the value of **yppasswd_ARG**.

```
yppasswd_ARG="/etc/passwd -m passwd"
```

3. Type these commands from the shell:

```
# ypinit -m ↵
# yppasswd /etc/passwd -m passwd ↵
```

4. If you want to run your system as an NIS server as well as a master, execute this command:

```
# ypserv ↵
```

Setting Up Your AViiON Computer as an NIS Server

Perform these procedures to set up your AViiON computer as an NIS server in an existing NIS domain:

1. Edit **/etc/nfs.params** and set the value of **ypserv_START**:

```
ypserv_START="SERVER"
```

2. Type these commands from the shell:

```
# ypinit ↵
# ypserv ↵
```

For more information on the NIS facility, see the ONC/NFS Release Notice and the manual *Managing ONC/NFS and Its Facilities on the DG/UX™ System.*

Where To Go Next

Adding and setting up software packages may require that you rebuild and reboot the kernel. If you added a third-party vendor software package, check its release notice for information on any requirements to add or change a tunable variable in the system configuration file. Refer to Chapters 11 and 12 for information on building and booting a kernel.

End of Chapter

Chapter 8
Adding Clients

You will use the procedures in this chapter if you installed an OS server to which you want to link client computers. An OS server supplies a bootable operating system and file system space via a local area network (LAN) to an OS client. Before you add OS clients, you must create the appropriate logical disks to accommodate them. If you have not created logical disks for OS clients, refer to Chapter 2 before continuing with these instructions.

An X terminal provides X Window System graphics support. Lacking its own disk and operating system, the X terminal depends on an OS server for its secondary bootstrap. OS clients and X terminals are referred to as clients in this chapter.

This chapter includes a discussion of these topics:

- Types of OS clients
- Types of OS releases
- Network planning for clients
- Adding clients to network databases
- Adding an X terminal
- Building an OS diskless client kernel
- Adding an OS diskless client to the release
- Booting the OS diskless client kernel
- Adding an OS client with local **root** and **swap** logical disks
- Adding an OS client with a local **swap** logical disk

Types of OS Clients

In an OS client-server configuration, the OS server supplies a bootable operating system and file system space via an Ethernet local area network (LAN) to an OS client. (The OS comprises the **/srv/dump**, **/srv/swap**, **/srv/root**, and **/usr** file systems on the the **srv_dump**, **srv_swap**, **srv_root**, and **usr** logical disks, respectively.) Traditionally, an OS client with no attached physical disk receives its entire operating system and file system space from an OS server. However, if an OS client has an attached physical disk, you may choose to put part of the operating system on the local physical disk to maximize system resources and to improve system performance. The three types of required logical disks (**root**, **usr**, and **swap**) can be divided between two physical disks, one attached to the OS server and the other to an OS client. The three OS client-server configurations follow:

Diskless — The OS client has no physical disk and receives its entire operating system and file space from the OS server over an Ethernet LAN.

Local root and swap — An OS client with an attached physical disk provides its own local **root** and **swap** logical disk resources, while receiving its **usr** logical disk resources from the OS server via an Ethernet LAN.

Local swap — An OS client with an attached physical disk provides only its own local **swap** logical disk resources, while receiving its **root** and **usr** logical disk resources from the OS server via an Ethernet LAN.

Since the diskless client is the most common type of OS client, its addition to a configuration is discussed first. However, the final two major sections in this chapter provide instructions for configuring an OS client with local **root** and **swap** logical disk resources and an OS client with only local **swap** resources.

Types of OS Releases

In addition to providing the 5.4 primary release of the DG/UX system to its OS clients, an OS server can also provide other (secondary) releases of DG/UX or a foreign operating system. The operating system releases are arranged in a parallel hierarchical structure in the DG/UX file system. The 5.4 release of DG/UX is located at **/srv/release/PRIMARY**; secondary releases are located at **/srv/release/***release-name*, where *release-name* is the name of the secondary release such as **dgux_4.32**.

After the secondary release area is created (procedures are in Chapter 9), the addition of OS clients to both primary and secondary release areas is identical. An OS client can be attached to only one release at a time.

The examples in this chapter emphasize the addition of OS clients to the primary release because it is the most common. If you are adding clients to a secondary release, you can substitute your *release-name* for each instance of PRIMARY in the pathnames given in this chapter.

Network Planning for Clients

Two Client Network Planning Worksheets are provided in Appendix E to help you prepare for installation. It lists network parameters for which you need to supply values. Completing the worksheets before you begin the installation procedures for clients will speed up the installation process considerably.

If you have experience installing the DG/UX system or other operating systems, you may prefer to go directly to the Client Network Planning Worksheets.

Host name	A unique name you assign to your AViiON hardware that can contain up to 31 alphabetic and numeric characters. However, you are advised to keep the names short. Host names that relate to the use or location of the system are particularly helpful in networked environments where hosts may share file systems. Examples of host names are **fred, jamaica,** and **writer_doc.** Do not use the capitalized names **MY_HOST** or **PRIMARY**; these names are reserved by the system.
Internet address	The Internet address of the host being set up is provided by the network system administrator. An example of an Internet address is **128.223.2.1.** In this example, **128.223** refers to the network number, **2.1** refers to the host number. The dots are field separators.
Ethernet address	Host address that is unique to the particular hardware. This address is preset at the factory. It consists of six colon-separated 2-digit fields in the form *nn:nn:nn:nn:nn:nn.*
NIS domain name	A named set of NIS maps located in **/etc/yp/***domain-name*. Machines having this directory as their default NIS domain share the data found in its maps.
Do you subnet?	Subnetting allows you to associate multiple physical networks into a single logical network. You must answer "yes" or "no" to the query about subnetting.
Network mask	A hexadecimal bit pattern that specifies the number of bits used to identify the network part of an Internet address.. **0xff000000** and **0xffff0000** are examples of network masks. A network mask is required only when your network is subnetted.

File-Naming Conventions

When creating files, follow standard DG/UX file-naming conventions. A file name can contain the characters:

```
a-z  A-Z  0-9  !  %  +  ,  .  :  @  _  ~  -
```

A file name cannot contain the / character. The file name can be up to 255 characters in length. However, you are advised to keep it short.

Adding Clients to Network Databases

To add each client (OS and X) to a number of network databases, you perform the following tasks.

- Add a client to the hosts database
- Add a client to the ethers database
- Add a client to the trusted hosts database

CAUTION: If you installed the ONC/NFS package on your system, and you declared the OS server as an NIS master (covered in Chapter 7), you set up OS clients in NIS hosts and ethers databases instead of local hosts and ethers databases. You must delay the addition of clients to a release so that the additions of OS clients to the NIS hosts and ethers databases will have had time to propagate through the system. Therefore, you should add clients to the NIS hosts and ethers databases well in advance (for example, over night) of adding clients to the operating system release, which is covered in "Adding an OS Client to a Release."

Specifying a Client Internet Address

The hosts database contains a list of Internet address and client host name pairs to enable network communications between the OS server and its clients.

To add an entry to the hosts database, follow this path through **sysadm** and respond to the prompts:

```
Networking -> TCP/IP -> Hosts -> Add
```

NOTE: The following dialogue assumes a networked environment in which the OS server has also been declared as the NIS master. (Declaration of an OS server as the NIS master is provided in Chapter 7.)

The first prompt asks for the hosts database to use. If you are networked, press New Line to accept the default value NIS.

```
Hosts database to use: [NIS (YP)] ↵
```

NOTE: If the OS server is not an NIS master, the first message is not displayed. Instead, the following message is displayed:

```
This host is not the NIS master.  Only the local Hosts
database will be used.
```

This message reports that the OS server is not the NIS master. The operation allows you to insert a new entry in the local hosts database for each client.

The next two prompts query you for the host name and the Internet address of the OS client added. Consult your Client Network Planning Worksheets that you completed in Appendix E for this information.

```
Host Name: mad-max ↵
Internet Address: 123.227.2.14 ↵
```

You may also specify an alternate name (alias) for your host. For example, if your department uses a lengthy host-naming convention, you may choose a shorter alias for convenience.

```
Alias List: george ↵
```

Finally, press New Line to accept the values that you have entered.

```
OK to perform operation?  [yes] ↵
mad-max has been added.
```

After answering these questions, the client is added automatically to the appropriate hosts file.

Repeat this procedure to add as many clients as desired to the hosts database.

Specifying a Client Ethernet Address

The ethers database contains a list of Ethernet address and client host name pairs to enable network communications between the OS server and its clients.

To add an entry to the ethers database, follow this path through **sysadm** and respond to the prompts:

```
Networking -> TCP/IP -> Ethers -> Add
```

NOTE: The following dialogue assumes a networked environment in which the OS server has also been declared as the NIS master. (Declaration of an OS server as the NIS master is provided in Chapter 7.)

The first prompt asks for the ethers database to use. If you are networked, press New Line to accept the default value NIS.

```
Ethers database to use: [NIS (YP)] ↵
```

NOTE: If the OS server is not an NIS master, the first message is not displayed. Instead, the following message is displayed:

```
This host is not the NIS master.  Only the local Ethers
database will be used.
```

This message reports that the OS server is not the NIS master. The operation allows you to insert a new entry in the local ethers database for each client.

The next two prompts query you for the host name and the Ethernet address of the OS client added. Consult your Client Network Planning Worksheets that you completed in Appendix E for this information.

```
Host Name: mad-max ↵
Ethernet Address: 08:00:1B:03:32:43 ↵
```

Finally, press New Line to accept the values that you have entered.

```
OK to perform operation?  [yes] ↵
mad-max has been added.
```

After answering these questions, the client is added automatically to the appropriate ethers file.

Repeat this procedure to add as many clients as desired to the ethers database.

Allowing Client Remote Login to the OS Server

Permission to access remote hosts via a remote login may be helpful for such tasks as checking the print queue for a printer that is attached directly to the OS server.

To add user login names to the trusted hosts database, follow this path through **sysadm** and respond to the prompts:

```
Networking -> TCP/IP -> Trusted Hosts -> Add
```

The first prompt asks for the host name of the OS server machine to which the specified user login can remotely access.

```
Host Name: max-server ↵
```

You are next asked to specify the authorized user who has login privileges.

```
User Name: [all] johnson ↵
```

Finally, press New Line to accept the values that you have entered.

```
OK to perform operation?  [yes] ↵
johnson on max-server has been added.
```

The named user login on the named remote system (the OS client) in the **/etc/host.equiv** file can access the OS server system by using the remote shell (**rsh** or **remsh**) or the remote login (**rlogin**). Refer to the **rsh**(1), **remsh**(1), and **rlogin**(1) manual pages for more information.

Adding an X Terminal

An X terminal client is a type of graphics terminal that provides X Window System graphics support. Lacking its own disk and operating system, the X terminal depends on an OS server for its secondary bootstrap.

Before you add an X terminal, make sure that you have entered its Internet and Ethernet addresses to **/etc/hosts** and **/etc/ethers**.

To add an X terminal follow this path through **sysadm** and respond to the prompts:

```
Client -> X Terminal -> Add
```

You are first prompted to specify the host name of the client. Next, you are asked to supply the pathname of the bootstrap file for this release. Press New Line to accept the default. A link to this file will be made for each client.

```
Client Host Name: gyramax ↵
Bootstrap File: [/usr/opt/X11/xtd/avx30boot] ↵
OK to perform operation? [yes] ↵
Xterminal gyramax has been added.
```

The final message indicates that a link was established between the client and the bootstrap.

Building a First-Time Custom Kernel for an OS Client

OS clients as well as the OS server must have kernels to provide operating system services. (Refer to Chapter 11 for complete information on kernels.)

The OS server must build each OS client its first kernel only to establish an OS client's basic operation. After an OS client is operational, however, it can build its own kernel, if necessary. You may choose to build one kernel for all OS clients, or you may choose to build individual kernels for each OS client.

The standard location for all kernels of the primary release (5.4) of the DG/UX system is **/srv/release/PRIMARY/root/_Kernels**. Multiple OS clients sharing the same kernel are linked to a common kernel from their **/** directory. By convention, the common kernel is named **dgux.diskless**.

To build a first-time kernel for a client, follow this path through **sysadm** and respond the prompts.

```
System -> Kernel -> Build
```

The first prompt asks for the name of the system configuration file.

```
System configuration file name: [aviion] diskless ↵
[system.diskless] Correct? [yes] ↵
```

If you are adding multiple OS clients, choose whether to build a common kernel for all OS clients or individual kernels for each OS client. If all OS clients have identical hardware and software configurations (for example, none has any attached devices), you can build a common kernel for all OS clients to share.

A primary advantage of sharing a common kernel is in saving physical disk space. Conversely, a disadvantage is a reduction of security. Because all kernels have root access to the kernel, any user with superuser privilege on an OS client computer can alter the kernel, and effectively change the common kernel.

If, however, an OS client differs in its hardware configuration from the common OS clients (for example, it has an attached parallel line printer), a unique kernel is required. A primary advantage of creating a unique kernel is that superusers on the OS client computer can change their own kernels, but no one else's. A disadvantage of creating a unique kernel is that more physical disk space is used.

A conventional name for a system file for a commonly shared kernel is **system.diskless**. Entry of **diskless** as the system file name extension yields **system.diskless**. If you enter a system file name that doesn't exist, a new one is created from scratch. The new file contains a series of concatenated prototype system files, named **system.***package***.proto**, where *package* can be **nfs**, **tcpip**, or any other software package name. These files are located in **/usr/src/uts/aviion/cf**. Also, the system file contains a template of typical OS client devices, which you must edit explicitly.

If that system file already exists, it will not be overwritten; rather, it is opened for editing. For instance, if you added a device to your hardware configuration since the system file was last generated, you will have to add the device name explicitly to the list. The system file is located in **/usr/src/uts/aviion/Build**, which is a symbolic link to **/var/Build**.

You are given a chance to verify the selected system file name. The host name is often used to name a kernel for this host. However, if you build more than one kernel, you will need to develop a convention to distinguish them.

The next prompt asks for whom you want to build this kernel.

```
Build for this host or for OS client(s) of
this host: [this host] OS client ↵
```

Type **OS Client** in either upper- or lower-case.

Finally, you escape to the **vi** editor to edit the system file. (Refer to Appendix A for a **vi** command summary.) Alternatively, you can specify the editor of choice if you knows its location on your system.

```
Editor: [/usr/bin/vi] ↵
```

Editing the System Configuration File for an OS Client

The system file is lengthy; it spans several screens. Configuration variables that do not apply to your system are ignored (or commented out). A line that is commented out contains a # in the first column. You are advised to read the contents of the system file before you edit it.

The three sections that you will be most concerned with are:

- OS client configuration variables
- General configuration variables
- Typical OS client hardware devices

Figure 8-1 shows an excerpt from a system file named **system.diskless.**

```
OS Client Configuration Variables:
#
# These configuration variables specify that the root and swap
# file systems will be mounted over NFS.
#

    NETBOOTDEV          "inen()"
    ROOTFSTYPE          NETWORK_ROOT
    SWAPDEVTYPE         NETWORK_SWAP
#-----------------------------------------------------------------
# OS Client Hardware Devices:
#
# These hardware devices are found on most operating system
# clients.  You must add any other hardware devices found on
# the client (using the examples below as a guide), and delete
# any devices which are not found on the client.
#

    kbd()               # -- keyboard
    grfx()              # -- graphics display
    inen()              # -- integrated Ethernet controller
    duart()             # -- integrated Duart terminal line controller
#-----------------------------------------------------------------
# General Configuration Variables:
#
# The NODE variable controls your nodename for uname(1) and uucp(1).
#

    NODE                "diskless"

#-----------------------------------------------------------------
# Typical AViiON OS client device configuration

# Note that a system can have an lp() controller or a second duart()
# but not both.  Determine which situation applies.


#    kbd()              # -- keyboard
#    grfx()             # -- graphics display
#    sd(insc(),*)       # -- all SCSI disks on integrated SCSI adapter
#    st(insc(),*)       # -- all SCSI tapes on integrated SCSI adapter
#    inen()             # -- integrated Ethernet controller
#    duart()            # -- integrated Duart terminal line controller
#    duart(1)           # -- second Duart (if present on system)
#    lp()               # -- integrated printer controller (if present)
```

Figure 8-1 *Excerpts from System File for an OS Client*

OS Client Configuration Variables

NETBOOTDEV, ROOTFSTYPE, and SWAPDEVTYPE are used for providing resources to the OS client over the network. Make sure that there are no leading comment symbols preceding these variables.

General Configuration Variables

NODE refers to the host name of your computer. After you supply the name of the system file extension, "**diskless**," this name becomes the automatic NODE name. However, each time the OS client is rebooted, this value will be overridden by the host name provided during TCP/IP package setup (package setup is covered in Chapter 7). The node name is used by **uname**(1) command to report the name and other attributes of the current system. Also, the **uucp**(1) command uses the node name for performing file transfers on UNIX systems. It is also presented as part of the log-in banner message when you login to your system. The node name corresponds with the host name that you supplied during TCP/IP setup. The node name is restricted to 31 characters maximum. Also, variable DUMP is now set up through a new command named **dg_sysctl**(1M) that you type at the shell. Refer to Chapter 12 and the **dg_sysctl**(1M) manual page for more information.

The master files located in the **/usr/etc/master.d** directory contain a complete list of general configuration variables and default values. You can override a given default by setting the variable to the desired value in the system file. Read about the tunable variables in *Managing the DG/UX™ System*. You are accepting the defaults by not changing any values assigned to variables in the system file.

OS Client Hardware Devices

The typical OS client device configuration list shows a minimal arrangement of devices. If the OS client has additional devices that are not specified here, enter them. An example of typical devices you can specify follow.

```
kbd()              # -- keyboard
grfx()             # -- graphics display
sd(insc(),1)       # -- SCSI disk 1 on integrated SCSI adapter
st(insc(),4)       # -- SCSI tape 4 on integrated SCSI adapter
inen()             # -- integrated Ethernet controller
duart()            # -- integrated Duart terminal line controller
lp()               # -- integrated printer controller
```

Another list of typical OS client devices are provided in another file location from which to choose.

By perusing the system file, you will see a series of concatenated software package prototype files. If you have just loaded a new software package on your system, check its release notice for information on possible variable tuning. If you need to tune a variable, either enter or modify it in the appropriate location in the system file.

After you have finished editing the system file, you can write the file and quit the editor by issuing the **ZZ** command.

Continued Kernel Build for the OS Client

The messages that appear during a kernel build are different between a system that does not have existing OS clients and one that has existing OS clients. The following messages appear for a system that has no existing OS clients.

```
There are no operating system clients of this machine
for which to link the new kernel.  The new kernel may be
used by any clients added later.  The kernel pathname will
be /srv/release/PRIMARY/root/_Kernels/dgux.diskless.
Continue with the build?  [yes] ↵
Configuring system...
Building kernel...
Successfully built dgux.diskless.
```

The following message appears for a system that does have existing OS clients.

```
Link the new kernel to which OS clients: [all] ↵
```

You must link the new kernel to **/dgux** before you can use it. If you do not link the new kernel to **/dgux**, the existing kernel (if one exists) remains linked to **/dgux**. Accept the `yes` default to link the new kernel to **/dgux**. The kernel build resumes. The system file is then configured (a C language source file is generated) and a new kernel is produced. See the **config**(1M) manual page for more information on kernel building.

A common source of many kernel build failures is the inadvertent absence of a comment symbol (#) which is used to flag descriptive notes to be ignored. Be sure you comment out all text that is to be ignored. Check your spelling and verify the device names specified in DG/UX common device specification format.

After the new kernel is built, the bootable kernel file is in the **/srv/release/PRIMARY/root/_Kernels** directory as **dgux.***name* (**dgux.diskless** is an example) and is also linked to **dgux** for the specified client.

The new kernel will take effect only by booting it. The new OS client kernel must now be booted at the OS client machine. Booting instructions are provided in a later section.

Adding an OS Client to a Release

After the OS server is operational (DG/UX 5.4 and/or a secondary release is running), you must attach each client to the desired release. You will perform the following tasks.

- Create a default OS client set
- List the contents of a default OS client set
- Modify a default OS client set
- Add an OS client to the client default set
- List an added OS client

CAUTION: If you installed the ONC/NFS package on your system, and you declared the OS server as an NIS master (covered in Chapter 7), you set up OS clients in NIS hosts and ethers databases instead of local hosts and ethers databases. You must delay the addition of clients to an operating system release so that the additions of OS clients to the NIS hosts and ethers databases will have had time to propagate through the system. Therefore, you should add clients to the NIS hosts and ethers databases well in advance of adding clients to the operating system release, which is covered in "Adding an OS Client to a Release." If you do not heed this requirement, your addition of clients to the operating system release will fail. Refer to "Adding Clients to Network Databases" for more information.

Creating a Default Client Set for OS Clients

A client set specifies general attributes of the operating system environment shared by all OS clients linked to a common kernel.

NOTE: If a default client set already exists, you do not need to create one.

To create a client set, follow this path through **sysadm** and respond to the prompts:

```
Client -> OS Client -> Defaults -> Create
```

The prompt for the client set's name follows:

```
Set Name: [unnamed]
```

You can either accept the default client set, `unnamed`, or you can create a client set to be customized. Press New Line to accept the default client set `unnamed` and its default attributes.

In this example, create a client set to be customized and name it **dgset.**

```
Set Name: [unnamed] dgset ↵
```

The name of the new default set must be unique. Change the `unnamed` default to a meaningful name.

Finally press New Line to accept the values you entered.

```
OK to perform operation?  [yes] ↵
Default set dgset has been created.
```

Listing the Contents of a Default Client Set

A client set specifies general attributes of the operating system environment shared by all OS clients linked to a common kernel.

To list the attributes of a client set, follow this path through **sysadm** and respond to the prompts:

```
Client -> OS Client -> Defaults -> List
```

The prompt for the client set's name and a confirmation of the request follows:

```
Set Name: [default] ↵
OK to perform operation? [yes] ↵
```

The attributes for the client set (in this example, `unnamed`, are listed:

```
Parameter           Value
---------           -----
client_release      PRIMARY
client_server
client_homedir      /home
client_swapsize     16
client_kernel       /srv/release/PRIMARY/root/_Kernels/dgux.diskless
client_bootstrap    /usr/stand/boot.aviion
```

Modifying a Default Client Set for OS Clients

After you create a client set to be customized, you must next modify it.

NOTE: If you accept the default client set named `unnamed`, you will not need to modify it.

To modify the default client set, follow this path through **sysadm** and respond to the prompts:

```
Client -> OS Client -> Defaults -> Modify
```

The Modify operation allows you to change one or more values in a default client set.

Enter the default set you created in the previous section, **dgset**.

```
Set Name: [default] dgset ↵
```

Press New Line to accept the default release area name or enter a secondary release area name.

```
Release Area: [PRIMARY] ↵
```

Ensure that the correct OS server host name is provided before pressing New Line.

```
Server Host Name: [mav-server] ↵
```

The next prompt asks for the name of the directory to contain home directories for users that you will add on the OS client system. The operation will add an entry for the directory to the OS client's file system table **/etc/fstab**. This directory must already exist on the OS server and be exportable. The operation will create a mount point for the directory relative to the OS client. This directory should be the same as the parent directory specified when adding a user on the OS client. If you do not specify a directory, the operation will not add a home directory entry to the OS client's **/etc/fstab** file.

```
Home Directory: [/home] ↵
```

Accept the default or enter the size of **/srv/swap** you recorded in your planning worksheet for logical disks.

```
Swap Size (in megabytes): (4-128) [16] ↵
```

Accept the default pathname of the kernel file for this release. A link to this file will be made for each OS client.

```
Kernel Pathname: [/srv/release/PRIMARY/root/_Kernels/dgux.diskless] ↵
```

The bootstrap file contains the bootstrap needed by each OS client to boot a kernel. Accept the default pathname.

```
Bootstrap File: [/usr/stand/boot.aviion] ↵
```

Finally, press New Line to accept the values that you have entered.

```
OK to perform operation? [yes] ↵
Defaults for set dgset have been modified.
```

After you answer all questions, the default set for diskless OS clients will be modified for your particular set of OS clients sharing the same release.

Selecting a Default Client Set

A default set must be selected before it can be used. To select a default client set, follow this path through **sysadm**:

```
Client -> OS Client -> Defaults -> Select
```

You select the desired default set as follows:

```
Set Name: [system] ? ♪

Enter the name of the default set to Select.
This must be a set that already exists.

Choices are

     1  system
     2  dgset
     3  unnamed

Enter a number, a name, the initial part of a name, <NL> to take
the default, ? for help, ^ to return to the previous query, < to restart
the operation, or q to quit.
Set Name: [system] dgset ♪
OK to perform operation? [yes] ♪
Default set dgset has been selected.
```

Typing ? gives a list of existing default sets. Now you can add an OS client to the default set.

Adding an OS Client to the Default Client Set

After you create and select a default client set and add entries for an OS client in the hosts and ethers databases, to link each OS client to that client set, follow this path through **sysadm** and respond to the prompts:

```
Client -> OS Client -> Add
```

The first prompt asks for the host name for the OS client being added to the default set.

```
Client Host Name: mad-max ♪
```

The remaining prompts are identical to those answered to modify a default client set. Refer to the section "Modifying a Default Client Set for OS Clients" for the dialogues.

Listing an Added OS Client

To verify the successful addition of an OS client, follow this path through **sysadm** and respond to the prompts:

```
Client -> OS Client -> List
```

Specify a client host name or "all." If you specify all clients, a list of name and release areas is displayed.

```
Client Host Name: all ↵
OK to perform operation?  [yes] ↵

Client Name              Release Area
-----------              ------------
matilda                  PRIMARY
mad-max                  PRIMARY
client_dgux4.32          dgux4.32
```

Booting the OS Client Kernel

You must now boot the OS client kernel that you just built at the OS client computer itself. You cannot boot the OS client kernel at the OS server computer.

Figure 8-2 shows a typical boot command and booting messages at an OS client.

```
SCM> b inen() -3 ↵

Booting inen()
Local Ethernet address is 08:00:1C:1F:03:77
Local Internet address is 128.222.14.32 or 80DE0E20 hex
Trying server at 128.222.14.31 or 80DE0E1F hex for TFTP transfer
00166464

DG/UX Bootstrap Release 5.4

Boot:  inen (0)
Local Ethernet address is 08:00:1C:1F:03:77
Local Internet address is 123.228.3.19
Broadcasting request for a boot server ...
Host name: max-server
Using max-server:/srv/release/PRIMARY/root/mad-max as root
Loading /dgux ..............................................
DG/UX System Release 5.4, Version generic
Using 16 Megabytes of physical memory
Found 1 processor(s)
Processor 1 is running
Configuring devices ...................................


    Checking local file systems ...
    Current date and time is Tue Apr 23 14:30:24 EDT 1991
    Checking system files .....
    Enabling automatically pushed STREAMS modules .....
    Linking short names for /dev device nodes ...
    Loading terminal controllers ....
    Starting disk update daemons .....
    Mounting local file systems ...........
    Checking for packages that have not been set up ...
    Starting miscellaneous daemons ...
    Starting TCP/IP network interfaces .........
    Starting system logging daemon ....
    Starting NIS services ......
    Starting NFS lock services ......

    NOTE:  Pausing for 15 seconds to allow remote systems to
           reclaim NFS locks.

    Starting batch services ....
    Starting line printer scheduler ....
    Saving ex(1) and vi(1) temporary files ....
    Starting NFS services .....
    Starting TCP/IP daemons ........
    Mounting NFS file systems .....

    NOTE:  See /etc/log/init.log for a verbose description of the
           system initialization process.
```

Figure 8-2 *Typical Messages for OS Client Booting to Run Level 3*

In this example, **mad-max** is the OS client and **max-server** is the OS server.

Logging In to the DG/UX System at the OS Client

The login prompt appears after the DG/UX system has finished booting. Therefore, OS clients should log in as **sysadm** to access the **sysadm** utilities for procedures to set up software packages and mount file systems before their computers are fully operational.

If the OS server have installed the X Window System package, you will see the following login box displayed on the screen (see Figure 8-3).

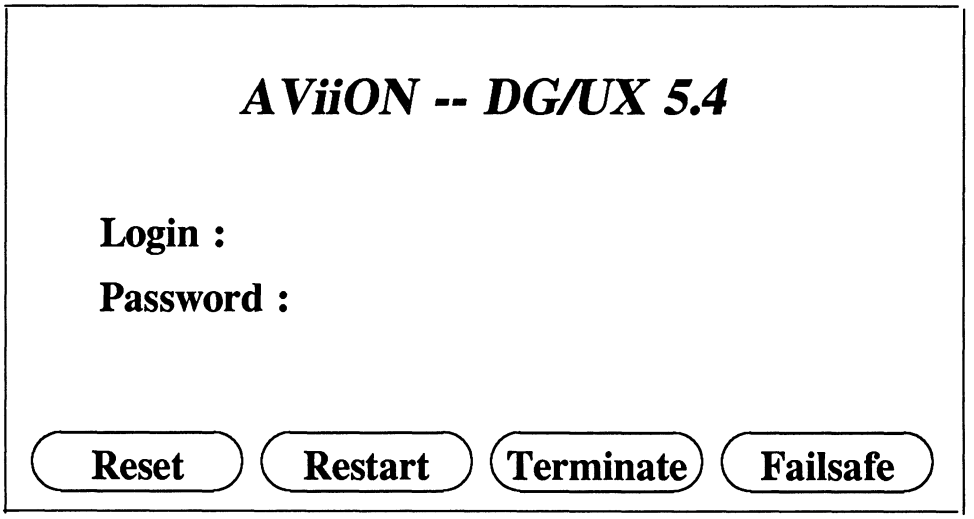


Figure 8-3 *Login Screen for a Graphics Monitor*

The four buttons along the bottom of the screen are provided as aids to controlling the X server, which is the program that controls the display of information to the screen.

Reset Resets the X server. The X server is not terminated.

Restart Terminates the X server, then restarts it.

Terminate Terminates the X server, giving control to a single VT100 terminal emulation screen.

Failsafe Is a toggle button, which limits startup to a single xterm window.

NOTES: If you choose to exit the X Window environment and continue the procedures in a single VT100 terminal screen environment, move the cursor to the "Terminate" button and click.

To return to the X Window environment, type **xdm** at the login prompt. The X Window environment will be restored.

You will be presented with a single window and a login prompt. Log in as **sysadm**.

Setting Up Packages at the OS Client

Before the OS client computer is operational, the DG/UX packages must be first set up.

Perform these procedures at the OS client computer.

Follow this path through **sysadm**:

```
Software -> Package -> Set up
```

Figure 8-4 provides a typical dialogue for setting up all DG/UX software packages for an OS client.

```
Package Name(s):  [all] ? ↵

Select 'all' to set up package(s) in the root file systems of all
clients attached to the selected release area.  Select individual
clients if you want to set up the package only for this client.

Select nothing or 'none' to set up the package only in your own root
file system and not in the root file system of any clients.  Typically,
clients set up their own packages.

Choices are

1 all
2 X11
3 X11.lg
4 dgux
5 aview
6 nfs
7 onc
8 tcpip

Package Name(s) [all] ↵
OK to perform operation? [yes] ↵
```

Figure 8-4 *Typical Entire Software Package Setup Dialogue*

The choices listed will depend on the packages that were loaded and set up on the OS server.

The next stage of initialization is **dgux** package setup, which takes about three minutes. The messages produced from this process follow:

```
Setting up dgux in mad-max root.

Setting up DG/UX...

Initializing DG/UX prototype files.....................

Creating DG/UX run level links.....................

Initializing sysadm(1M).............

Continuing setup of DG/UX...........

NOTE:  See /srv/release/PRIMARY/root/mad-max
       for a detailed account of the root setup of DG/UX.
```

NOTE: In all package setup example messages, "**mad-max**," is the OS client host name.

Setting Up X11

The following messages appear for **X11**.

```
Setting up X11 in mad-max root.
Package X11 has been successfully set up in mad-max root.
Package setup for X11 is complete.
```

Setting Up AView

The following messages appear for **AView**.

```
Setting up aview in mad-max root.
Package aview has been successfully set up in mad-max root.
Package setup for aview is complete.
```

Setting Up NFS

The following messages appear for NFS.

```
Setting up nfs in mad-max root.

    Setting up NFS in mad-max root.....

    Creating NFS run level links.....

    Initializing NFS prototype files.....

    NOTE:  See /srv/release/PRIMARY/root/mad-max/var/setup.d
           /log/nfs.root for a detailed account of the root
           setup of NFS.

Package nfs has been successfully set up in mad-max root.
Package setup for nfs is complete.
```

Setting Up ONC

In setting up ONC, you will need to supply one value: the name of the NIS domain. Check your Client Network Planning Worksheet that you completed in Appendix E for this information.

```
Setting up onc in mad-max root.

    Setting up ONC in mad-max root.....

    Creating ONC run level links.....

    Initializing ONC prototype files.....

    Enter the NIS Domain name []: work-net ↵
    [work-net] Correct? [yes]:

    NOTE:  This host will first run as an NIS client.

    NOTE:  See /srv/release/PRIMARY/root/mad-max/var/setup.d
           /log/onc.root for a detailed account of the root
           setup of ONC.

Package onc has been successfully set up in mad-max root.
Package setup for onc is complete.
```

NOTE: If you are upgrading, the NIS domain name will be supplied by default.

Setting Up TCP/IP

Setup of TCP/IP requires more advance planning than any of the other packages. You must supply configuration information to these prompts. Check your OS Server Network Planning Worksheets that you completed in Appendix E for this information. Also, refer to *Managing TCP/IP on the DG/UX™ System* for information that will help to set up the TCP/IP package.

```
Setup package tcpip in mad-max root

    Setting up tcpip...

    The following queries refer to the primary network interface.

    Enter host name: mad-max ↵
    [mad-max] Correct? [yes] ↵
    Enter host Internet address: 128.222.8.60 ↵
    [128.222.8.60] Correct? [yes] ↵
    Is your local network subnetted? [no] yes ↵
    Enter the network mask: 0xffffff00 ↵
    [0xffffff00] Correct? [yes] ↵
    Enter the controller device name: inen0
    [inen0] Correct? [yes]

    NOTE:  Using inen0 as the primary network interface controller.

    Package setup for has been successfully set up in mad-max root.

    NOTE:  See /var/setup.d/log/tcpip.root file for a
           verbose description of the package setup for root.

    Package setup for tcpip is complete.
```

The setup procedures for TCP/IP and other packages have completed.

The superuser prompt (#) for input is displayed on your screen for continued system administration work.

If you are ready to start your normal work, you must have a user account. Refer to Chapter 4 for information on adding user accounts, which includes the assignment of a password.

Adding an OS Client with a Local Root and Swap Logical Disk Configuration

This type of OS client has an attached physical disk that can provide its own local **root** and **swap** logical disk OS resources (**/** file system and swap space), while receiving its **usr** logical disk OS resources (**/usr** file system) from the OS server via a LAN.

You will perform these steps:

- Complete the Planning Worksheet
- Build a temporary OS client kernel
- Build an OS client kernel with local **root** and **swap** logical disk resources
- Load the **root** logical disk on the local physical disk
- Set up the software packages
- Clean up at the OS server computer

Completing the Planning Worksheet

Before you set up this configuration, complete the following worksheet to reflect your particular arrangement.

Network Planning Worksheet for the OS Client with Local Root and Swap Logical Disks Configuration

Parameter Type	Example Value	Actual Value
OS client disk device name	**sd(insc(0),3,0)**	
OS client host name	**mav_client**	
OS server host name	**mav_server**	
OS server's Internet address	**128.222.3.120**	
OS client's Internet address	**128.222.3.86**	
OS client's Internet address (hexadecimal equivalent)	**80DE0356**	
NIS domain name	**my_domain**	
Do you subnet?	**yes**	
Network mask	**0xffffff00**	

Building a Temporary OS Client Kernel

1. At the OS server, follow this path through **sysadm** to specify the OS client host name and the OS client internet address in the network database.

   ```
   Networking -> TCP/IP -> Hosts -> Add
   ```

   Respond to the prompts using the information recorded on your planning worksheet.

2. At the OS server, follow this path through **sysadm** to specify the OS client host name and the OS client Ethernet address in the network database.

   ```
   Networking -> TCP/IP - > Ethers -> Add
   ```

   Respond to the prompts using the information recorded on your planning worksheet.

3. Follow this path through **sysadm** to create a temporary system file.

```
System -> Kernel - > Build
```

   At the prompt for the system file name, enter **temp** and press New Line.

4. Edit the **system.temp** file.

5. When asked whether you want to build for this host or for an OS client of this host, answer **OS Client.**

6. Press New Line when prompted to use the editor **/usr/bin/vi** to edit the new system file.

7. Edit the "Typical Hardware Devices for an OS Client" section, and insert the name of the local disk device. An example follows:

   sd(insc(0),3,0)

8. Complete the kernel build and configuration by exiting the editor and pressing New Line to affirm a continuation of the build process. The new kernel will be located at **/srv/release/PRIMARY/root/_Kernels/dgux.temp.**

Adding an OS Client to a Release

In this section, you will create, modify, select, and add an OS client to a default set. When prompted for the kernel pathname, make sure that it is **/srv/release/PRIMARY/root/_Kernels/dgux.temp**. Refer to the section "Adding an OS Client to a Release" in this manual for this information.

Building a Kernel with Local Root and Swap Logical Disk Resources

1. At the OS server, follow this path through **sysadm** to create a system file named **system.local_root_swap.**

```
System -> Kernel - > Build
```

2. At the prompt for the system file name, enter **local_root_swap** and press New Line.

3. Edit the **system.local_root_swap** file.

4. When asked whether you want to build for this host or for an OS client of this host, answer **OS Client.**

5. Press New Line when prompted to use the editor **/usr/bin/vi** to edit the new system file.

6. Edit the "OS Client Configuration Variables" section of **system.local_root_swap.** Insert a pound sign (#) before the ROOTFSTYPE and SWAPDEVTYPE variables. At the end of the list, add the NETSTART variable and its assigned value.

```
NETBOOTDEV          "inen()"
# ROOTFSTYPE         NETWORK_ROOT
# SWAPDEVTYPE        NETWORK_ROOT
NETSTART           REAL_NET
```

7. In the "Typical Hardware Devices for an OS Client" section, insert the name of the local disk device. An example follows:

   sd(insc(0),3,0)

8. Complete the kernel build and configuration by exiting the editor and pressing New Line to affirm the continuation of the build process. The new kernel will be located at **/srv/release/PRIMARY/root/_Kernels/dgux.local_root_swap.**

9. Answer **no** to the query asking to link the new kernel to which OS clients.

10. Copy the **dgux.local_root_swap** kernel to **/usr/stand** using this command.

    # cp /srv/release/PRIMARY/root/_Kernels/dgux.local_root_swap /usr/stand/dgux.local_root_swap ↵

11. To export the **/usr** file system and temporarily give root permission to the OS client, enter this command.

    # exportfs -iv -o root=mav_client /usr ↵

NOTE: Do not edit the **/etc/exports** file.

You have completed the procedures at the OS server computer.

Loading the Root Logical Disk on the Local Physical Disk

Perform these procedures at the OS client computer:

1. Boot the OS client to init level S (single-user mode) using this command.

   SCM> **b inen(0) -S ↵**

2. From the Bourne shell, set and export the **TERM** variable:

   # TERM=vt100 ↵
   # export TERM ↵

3. Invoke stand-among **diskman** as follows:

   # diskman ↵

   From the "Diskman Main Menu," perform these steps. Refer to the Logical Disk Planning Worksheet that you completed in Appendix E for this information.

- Prepare the physical disk **sd(insc(0),3,0)**, if necessary.
- Create the **root** logical disk and file system.
- Create the **swap** logical disk, but do not create a file system on the logical disk.
- Exit **diskman**.

4. At the shell prompt (#), type these commands to mount the local root file system at a temporary mount point:

```
# mkdir /mnt ↵
# mkdir /mnt/tmp_root ↵
# mount /dev/dsk/root /mnt/tmp_root ↵
```

5. Type these commands to load the local root file system:

```
# cd /usr/root.proto ↵
# tar -cf - . | (cd /mnt/tmp_root ; tar -xf - ) ↵
```

6. Type these commands to copy the **dgux.local_root_swap** kernel into the local root:

```
# cp /usr/stand/dgux.local_root_swap /mnt/tmp_root/dgux.local_root_swap ↵
```

7. Type these commands to create an **/etc/hosts** file:

```
# cd /mnt/tmp_root/etc ↵
# cp hosts.proto hosts ↵
```

8. To add an entry for the OS server host name and Internet address in the **/mnt/tmp_root/etc/hosts** file, edit the file and enter:

```
128.222.3.120    mav_server
```

9. From the shell, type these commands to create an **/etc/fstab** file:

```
# cd /mnt/tmp_root/etc ↵
# cp fstab.proto fstab ↵
```

10. To specify that **/usr** will be mounted remotely, first type a pound sign (#) to comment out the following line from **/etc/fstab**:

```
# /dev/dsk/usr    /usr    dg/ux rw 1 0
```

Insert this line:

```
mav_server:/usr /usr    nfs bg,rw,hard 0 0
```

You have loaded the **root** logical disk on the local physical disk.

Setting Up Packages on the Local Disk's / File System

1. At the OS client computer, from the shell type this command to halt the system:

   # halt -q ↵

2. At the SCM prompt, type this command to boot the installer kernel from the local disk device:

   SCM> **b sd(insc(0),3,0)root:/dgux.local_root_swap -i** ↵

   This leads directly to package setup.

3. Perform package setup. Refer to the planning worksheet that you completed in this chapter for the values that you will need to supply during setup.

4. After you complete package setup, you will be asked if you want to build a kernel; answer **"no"**. Otherwise, you would build a custom kernel that would be suitable for a diskless OS client configuration.

5. Log in as **sysadm.**

6. Change run levels using this command:

   # init 3 ↵

7. Using this command, link the **dgux.local_root_swap** file to **/dgux** so that the correct kernel is subsequently rebooted at the client host.

   # admkernel -o link local_root_swap ↵

Your system is now operational.

Cleaning Up at the OS Server

Perform these steps at the OS server computer to clean up.

1. From **sysadm**, follow this path to delete OS client **mav_client.**

   ```
   System -> OS Client -> Delete
   ```

2. From the shell prompt, enter these commands to link the OS client's Internet address to the **/tftpboot** directory.

   # cd /tftpboot ↵
   # ln -s /usr/stand/boot.aviion 80DE0356 ↵
   # initrarp ↵

   where:

80DE0356 is the hexadecimal equivalent for the OS client's Internet address **128.222.3.86**. Check your Planning Worksheet for this information. This link is needed for maintaining OS client address entries that are resolved and used each time an OS client boots. This OS client (with remote **root** and **swap**) will fail to boot if this entry is not present on the OS server.

Your system is now operational.

Adding an OS Client with a Local Swap Logical Disk Configuration

This type of OS client has an attached physical disk that can provide its own local **swap** logical disk OS resources, while receiving its **root** and **usr** logical disk OS resources (**/** and **/usr** file systems) from the OS server via an Ethernet LAN.

You will perform these steps:

- Collect configuration information.
- Build a temporary diskless OS client kernel.
- Build an OS client kernel with local **swap** logical disk resources.

Collecting Configuration Information

Before you perform these procedures, make sure that you know the names of the OS server, OS client, and the DG/UX common device specification of the disk device that is locally attached to the OS client.

Building a Temporary Diskless OS Client Kernel

1. Follow this path through **sysadm** to create a temporary system file.

   ```
   System -> Kernel - > Build
   ```

   At the prompt for the system file name, enter **temp** and press New Line.

2. Specify a kernel build for the OS client **mad-max**. Then edit the **system.temp** file.

3. When asked whether you want to build for this host or for an OS client of this host, answer **OS Client.**

4. Press New Line when prompted to use the editor **/usr/bin/vi** to edit the new system file.

5. Edit the "Typical Hardware Devices for an OS Client" section, and insert the name of the local disk device. An example follows:

   sd(insc(0),*)

   The asterisk (*) is a pattern-matching metacharacter. Any SCSI disk device is matched by this pattern.

6. Complete the kernel build and configuration by exiting the editor and pressing New Line to affirm the continuation of the build process. The new kernel will be located at **/srv/release/PRIMARY/root/_Kernels/dgux.temp.**

Building a Kernel with Local Swap Logical Disk Resources

1. Boot the OS client to a run level of 3 using this command:

   ```
   SCM> b  inen(0) -3 ↵
   ```

2. Invoke stand-alone **diskman**:

   ```
   # diskman ↵
   ```

   From the "Diskman Main Menu," perform these steps. Refer to the Logical Disk Planning Worksheet that you completed in Appendix E for this information.

   a. Prepare the physical disk **sd(insc(0),3,0)**, if necessary.

   b. Create the **swap** logical disk, but do not create a file system on the logical disk.

   c. Exit **diskman**.

3. Edit the **/etc/fstab** file by typing a pound sign (#) to comment out the first line:

   ```
   # mav_server:/srv/swap/mav_client      swap    swap sw x 0
   ```

   Insert the following line:

   ```
   /dev/dsk/swap  swap_area  swap  sw 0 0
   ```

4. Follow this path through **sysadm** to create a system file named **system.local_swap**.

   ```
   System -> Kernel - > Build
   ```

5. When asked what OS client you want to build the kernel for, answer "**this host**".

6. Edit the "OS Client Configuration Variables" section of **system.local_swap**. At the end of the list, add the SWAPDEVTYPE variable and its assigned value.

   ```
   NETBOOTDEV          "inen()"
   ROOTFSTYPE          NETWORK_ROOT
   SWAPDEVTYPE         LOCAL_SWAP
   ```

7. In the "Typical Hardware Devices for an OS Client" section, insert the name of the local disk device. An example follows:

   ```
   sd(insc(0),*)
   ```

   The asterisk (*) is a pattern-matching metacharacter. Any SCSI disk device is matched by this pattern.

8. Complete the kernel build and configuration by exiting the editor and pressing New Line to affirm the continuation of the build process. The new kernel will be located at **/srv/release/PRIMARY/root/_Kernels/dgux.local_swap**.

9. From the shell type this command to halt the system.

```
# halt -q ↵
```

10. At the SCM prompt, type this command to boot the **dgux.local_swap** kernel from the local disk to an init level of 3:

```
SCM> b sd(insc(0),3,0)root:/dgux.local_swap -3 ↵
```

11. Using this command, link the **dgux.local_swap** file to **/dgux** so that the correct kernel is subsequently rebooted at the client host.

```
# admkernel -o link local_swap ↵
```

Your system is now operational.

End of Chapter

Chapter 9
Adding Secondary Operating System Releases

In addition to providing the 5.4 primary release of the DG/UX system to its OS clients over a LAN, an OS server can also provide multiple secondary releases to its OS clients. These types of operating systems can be installed in secondary release areas of your DG/UX system:

- Earlier versions of the DG/UX system (such as 4.32) on a 5.4 OS server system.
- Foreign release (such as SunOS).

  Moreover, the OS clients do not have to be the same hardware architecture as each other or as the OS server. Even though the DG/UX system software is the most common type of release, it does not necessarily have to be an operating system intended for Data General AViiON system hardware. To install a foreign OS in the secondary release area of a 5.4 release of the DG/UX system, you will need to consult the documentation and release notice of the foreign release.
- 5.4 version of the DG/UX system on a pre-5.4 (such as 4.32) OS server system.

This chapter focuses on procedures to install either the 4.32, 4.33, or 4.34 release of the DG/UX system as a secondary release on an AViiON computer running the 5.4 release.

If you are currently running an earlier revision of the DG/UX system (such as 4.32) as your primary release and you have no intentions of upgrading, you may instead choose to install the 5.4 release as a secondary release. Instructions are provided later in this chapter for setting up this configuration.

The number of secondary releases that can be supported is restricted only by disk space and system performance.

Installing DG/UX 4.32 as a Secondary Release on a DG/UX 5.4 Primary Release System

You will perform these steps:

- Create a secondary release area
- Load the DG/UX 4.32 release into the secondary release area
- Add OS clients of the secondary release
- Set up the 4.32 packages
- Mount local and remote file systems
- Build a kernel for a 4.32 diskless client
- Boot the 4.32 release at the OS client

Creating a Secondary Release Area

1. Before you create a secondary release area, you must create the required logical disks to accommodate the software. Refer back to Chapter 2 for information on logical disk planning and Chapter 3 for procedures to create the logical disks and mount them on the DG/UX directory structure before returning here.

   Recommended sizes for the required and optional logical disks may vary between releases 4.32 and 5.4. Table 9-1 shows the default sizes and mount point directories for the **root_dgux432, srv_swap, usr_dgux432, usr_opt_aview_dgux432** and **usr_opt_X11_dgux432** logical disks for the 4.32 release of DG/UX.

Table 9-1 Default Sizes and Mount Points for 4.32 DG/UX Logical Disks

Default Logical Disk Name	Mount Point Directory	Size in 512-Byte Blocks
srv_swap	**/srv/swap**	(50,000 * number-of-clients) + 17%
root_dgux432	**/srv/release/dgux_432/root**	(40000 * number-of-clients) + 10%
usr_dgux432	**/srv/release/dgux_432/usr**	160000 + 10%
usr_opt_aview_dgux432	**/srv/release/dgux_432/usr/opt/aview**	8000 + 10%
usr_opt_X11_dgux432	**/srv/release/dgux_432/usr/opt/X11**	105000 + 10%

This operation creates the directory structure to hold a secondary software release. To create a release, you will supply a release area name as well as path names identifying the locations of the **/usr** file system, swap area shared by OS clients, root areas, and any other shared software. This operation creates the release area directory structure, but the release area will remain empty until you explicitly load it.

2. Follow this path through **sysadm** to create a secondary release area.

```
Software -> Release Area -> Create
```

3. You are first prompted to type a unique name for the release you are adding.

```
Release Area Name: dgux_432 ↵
```

4. A default pathname for the OS client parent directory is the location for the root directories of the OS clients of this release. When you add an OS client to this release, the prototype host-dependent (**/**) directory structure is copied for the OS client. If the root parent directory does not already exist, it is created for you. Press New Line to accept the default.

```
Client Root Parent Directory: [/srv/release/dgux_432/root] ↵
```

5. Next, you are asked for the path name of the host-independent (**/usr**) directory for this release. Only one host-independent directory per release is created because all OS clients of a release share the **/usr** directory. If the directory does not exist, this operation creates it. Press New Line to accept the default.

```
/usr Directory: [/srv/release/dgux_432/usr] ↵
```

6. The share directory is any directory that contains software that the OS clients of this release will share. If the directory does not exist, the operation will create it. Accept the defaults for the share and swap directories. Refer to Figure 2-3 for a view of the **/srv/release/dgux_432** file system.

```
Share Directory: [/srv/share] ↵
Swap Directory: [/srv/swap] ↵
OK to perform operation?  [yes] ↵
Release dgux_432 has been added.  You
may now load software into
this release area using the Package management operations.
```

Loading the DG/UX 4.32 Release into the Secondary Release Area

After you create the secondary release area for the 4.32 DG/UX release, you can perform the same procedures used for installing the 5.4 release of the DG/UX system. Obtain the 4.32 release tape before you start these procedures.

Go to Chapter 7 for procedures on loading software into the secondary release area. Since you are loading a 4.32 release, the appearance of the prompts may be different from those shown in Chapter 7.

Adding OS Clients of the Secondary Release

Go to Chapter 8 for procedures to adding OS clients of the secondary release.

Setting Up the 4.32 Packages

Go to Chapter 7 for procedures to set up 4.32 packages for each OS client of the secondary release.

Mounting Local and Remote File Systems

Go to Chapter 3 for procedures to mount local and remote file systems.

Building a Kernel for a 4.32 Diskless Client

Perform these steps to build a kernel for a 4.32 diskless client.

1. To build a kernel for a diskless client attached to secondary DG/UX release 4.32, 4.33, or 4.34 on an OS server that is running the 5.4 release, boot the OS client to run level 1 as follows:

   # b inen() -1 ↵

2. Log in as **sysadm** and build a new kernel using the following command:

   # sysadm newdgux ↵

At the prompt for the system (file) name, accept the default file name extension `aviion` by pressing New Line. The system file name produced is **system.aviion**.

3. You are notified that the file does not exist, so are asked if you want to create the file. Accept the `yes` default by pressing New Line.

4. Press New Line when prompted to use the editor **vi** to edit the new system file.

5. Update the appropriate section that reflects your OS client's devices. Search for the header that identifies the AViiON workstation model for whom you are building a kernel. Insert a pound sign (#) before the device names that do not exist in your configuration. A typical example follows:

```
#### Typical AViiON 200 or 300 series workstation configuration:

# Note that your system an have a second duart() or an lp()
# controller, but not both.

    kbd()          # -- keyboard
    grfx()         # -- graphics display
#    sd(insc(),*)   # -- all SCSI disks on integrated SCSI adapter
#    st(insc(),*)   # -- all SCSI tapes on integrated SCSI adapter
    inen()         # -- integrated Ethernet controller
    duart()        # -- integrated Duart terminal line controller
#    duart(1)       # -- second Duart (if present in system)
#    lp()           # -- integrated line printer controller (if present)
```

Comment out (insert # symbols) for all devices for the remaining workstation and server configuration templates in the system file.

6. Locate the tunable configuration parameters section of the system file and change the value of the NODE variable to reflect the name of your host. An example follows:

```
NODE                "gyramax"
```

   where **gyramax** is the host name of the OS client.

7. Locate another tunable configuration parameters section of the system file and remove the leading # symbols. The result follows:

```
DUMP                "inen()"
PERCENTNFS          100
NETBOOTDEV          "inen()"
ROOTFSTYPE          NETWORK_ROOT
SWAPDEVTYPE         NETWORK_SWAP
```

Check the appropriate DG/UX release notice for other configuration parameters that may require tuning.

8. Complete the kernel configuration and build by exiting the editor and pressing New Line to affirm the continuation of the configuration process. The new kernel will be located at **/usr/src/uts/aviion/Build/system.aviion**.

9. Install the new kernel by answering the following queries as shown:

```
Install the New Kernel? [no] yes ↵
For a Diskless Client of this Host? [no] ↵
Kernel Pathname? [/dgux.aviion] ↵
The new kernel has been copied to /dgux.aviion.
Link /dgux to the New Kernel? [yes] ↵

The new kernel will not take effect until you shutdown and reboot.
To do this, quit from sysadm, and say:

        cd /
        /etc/shutdown
```

```
/etc/halt -q
```

10. Shut down the system by typing these commands:

```
# cd / ↵
# /etc/shutdown ↵
# /etc/halt -q ↵
```

Control goes to the SCM (System Control Monitor). After the SCM prompt is displayed, you know the DG/UX system has shut down.

Booting the 4.32 Release at the OS Client

From the SCM, boot the new 4.32 release as follows to a run level of 3:

```
SCM> b inen() -3 ↵
```

Booting messages will quickly scroll up the screen. Refer to Chapter 12 for a view of typical booting messages.

You will then log in to the DG/UX system. Refer to Chapter 12 for these procedures.

Your OS client is now running the 4.32 DG/UX release from a secondary release area of an OS server running a 5.4 DG/UX release.

Installing DG/UX 5.4 as a Secondary Release on a DG/UX 4.32 Primary Release System

If you are currently running an earlier revision of the DG/UX system (such as 4.32) as your primary release and you have no intentions of upgrading the primary release to 5.4, you may instead choose to install the 5.4 release as a secondary release. Instructions are provided for setting up this configuration.

You will perform these steps:

- Install patch 4.32 at the OS server
- Create the secondary release area and load the 5.4 release of the DG/UX system into the secondary release area
- Write-enable the **/usr**, **/usr/opt/X11**, and **/usr/opt/aview** file systems
- Boot the OS client to run level **i**
- Set up packages
- Build and boot a custom kernel

You will be using the DG/UX 4.32 installation procedures to install DG/UX 5.4 as a secondary release. Refer to *Installing and Managing the DG/UX™* for exact procedures.

Installing the Patch

Install the 4.32 patch to the 4.32 release at the OS server using these The patch retrofits the 4.32 release with commands that were upgraded in the 5.4 release. The affected commands are: the **loadpackage** option to **sysadm**(1), **tar**(1), **xdrtoc**(1M), and **rpc.lockd**(1M). Also, a kernel-related NFS locking feature has been modified. Read the DG/UX 5.4 release notice.

1. Log in as the superuser on your system as follows:

```
$ su ↵
```

2. Type these commands to load the patch files and to upgrade the **tar**(1) command.

```
# cd /usr ↵
# mt -f /dev/rmt/0n fsf 4 ↵
# mv /bin/tar /bin/432 ↵
# /bin/432_tar -xvf /dev/rmt/0 ↵
```

The 4.32 DG/UX **tar** command is renamed as **432_tar**.

Creating and Loading a Secondary Release Area

The procedures in this section assume the following values:

Secondary release area name	**dgux5.4**
OS server host name	**server_dgux5.4**
OS client host name	**client_dgux5.4**

1. Create the logical disks and file systems required for the 5.4 secondary release.

CAUTION: Add an extra 10% extra disk blocks to each required and optional logical disk that you create.

Consult the Logical Disk Planning Worksheet you completed in Appendix E and refer to Chapter 3 for the actual procedures to create logical disks and file systems.

2. Create a release area for the DG/UX 5.4 release using these procedures:

```
# sysadm addrelease ↵
Release Area? dgux5.4 ↵
Usr Directory? [/srv/release/dgux5.4/usr] ↵
Share Directory? [/srv/share] ↵
Client Root Parent Directory? [/srv/release/dgux5.4/root] ↵
Swap Parent Directory? [/srv/swap] ↵

Release dgux5.4 has been added.  You may now use loadpackage.
```

3. Loading the entire Client-Server AViiON System Package release 5.4 tape requires between 45 minutes to 1 hour.

CAUTION: Do not load any prep images (file names ending in **:prep**) from the 5.4 release tape. Loading these images will cause the package setup procedure to fail.

4. Load the 5.4 release of the DG/UX system into the secondary release area just created.

```
# sysadm loadpackage ↵
Release Area? [PRIMARY] dgux5.4 ↵
Tape Drive? [0] ↵
Is the tape mounted and ready? [yes] ↵
.
.
.
loadpackage is finished
#
```

You have loaded the 5.4 release of DG/UX into the secondary release area **/srv/release/dgux5.4.**

Providing an OS Client with a Kernel and Adding an OS Client to the Release

These procedures add one OS client to the secondary release area. However, you may choose to add multiple OS clients.

1. From the shell, copy the common kernel **dgux.installer** to the secondary release area's root directory.

```
# cd /srv/release/dgux5.4/root ↵
# mkdir _Kernels ↵
# cp dgux5.4/usr/stand/dgux.installer.diskless _Kernels/dgux.diskless ↵
```

2. Create a new default client set for this release area. A typical dialogue using a default set follows.

```
# sysadm clientdefaults ↵
Default Set Name: [unnamed] dgset_5.4 ↵
Release Area? [PRIMARY] dgux5.4 ↵
Default Swap Size? [16m] ↵
Default Home Directory? [/home] ↵
Default Kernel? [/srv/release/dgux5.4/root/_Kernels/dgux.diskless] ↵
Default Bootstrap File? [/srv/release/dgux5.4/usr/stand/boot.aviion] ↵
Defaults for set dgset_5.4 have been assigned.
```

3. Add an OS client to the secondary release area. A typical dialogue for modifying a default set follows.

```
# sysadm addclient ↵
Server's host name on client's network? [server] server_dgux5.4 ↵
Client Host Name? client_dgux5.4 ↵
Defaults Set Name? [none] dgset_5.4 ↵
Use ALL defaults from dgset_5.4? [yes] ↵
Creating the root.
Creating the swap file.
Creating the /etc/fstab.
Creating the /etc/hosts.
Creating the /tcpip/params.
Creating the /etc/nfs.params.
Creating the kernel link.
Creating the bootstrap link.
Client client_dgux5.4 has been added.
Do you wish to add another client? [yes] no ↵
```

Write-enabling the /usr, /usr/opt/X11, and /usr/opt/aview File Systems

1. From the shell, type this command to make the **/usr** file system unavailable for mounting over the network to OS clients.

```
# exportfs -uv /srv/release/dgux5.4/usr ↵
```

2. Edit the **/etc/exports** file by commenting out the first line and inserting the following line:

```
# /srv/release/dgux5.4/usr
```

/srv/release/dgux5.4/usr -root=client_dgux5.4

3. From the shell, use this command to re-export the **/usr** file system:

   # exportfs -v /srv/release/dgux5.4/usr ↵

4. Give the **/usr** file system write permission by editing the OS client's **/srv/release/dgux5.4/root/client_dgux5.4/etc/fstab** file. The first line shows the current read-only status; the second, the updated write permission.

```
server_dgux4.32:srv/release/dgux5.4/usr nfs ro x 0

server_dgux4.32:srv/release/dgux5.4/usr nfs rw x 0
```

5. Repeat steps 1, 2, and 3 for the **/usr/opt/X11** and **/usr/opt/aview** file systems.

6. Entries for **/usr/opt/X11** and **/usr/opt/aview** do not exist in the OS client's **srv/release/dgux5.4/root/client_dgux5.4/etc/fstab** file. Insert these lines:

   server_dgux5.4:/srv/release/dgux_5.4/usr/opt/X11 /usr/opt/X11 nfs rw x 0
   server_dgux5.4:/srv/release/dgux_5.4/usr/opt/aview /usr/opt/aview nfs rw x 0

7. Boot the OS client to a run level of **i** by entering the command:

   SCM> **b inen() -i ↵**

8. Refer to Chapter 7 for instructions on setting up the DG/UX 5.4 software packages on a DG/UX 4.32 system as a secondary release.

9. Refer to Chapter 11 for instructions on building on building a custom kernel for the OS client.

10. Refer to Chapter 12 for instructions on booting and logging in to the DG/UX system.

You have finished procedures for adding an OS client to a secondary release of the 5.4 DG/UX release.

End of Chapter

Chapter 10
Adding Physical Devices

You can add these physical devices to your hardware configuration at any time:

- Winchester disk drive
- CD-ROM (Compact Disk-Read-Only Memory)
- Multiple-read/write magneto-optical disk
- Diskette drive
- Tape drive

The first four physical devices are considered disk devices; the last one is a tape device. It is assumed that you have completed hardware installation of the device using the appropriate hardware documentation.

This chapter covers these topics:

- Identifying physical devices
- Formatting disk devices
- Special considerations for devices
- Removing removable media from disk drives
- Locating device file names after the system is booted

Identifying Devices

To format a disk device or configure a device in the kernel through the system file, you need the name of the device using the DG/UX common device specification format. See Appendix D for information on device naming. Examples of device names follow:

st(insc(0),5) Tape device at SCSI ID 5 on the first (0) integrated SCSI adapter (**insc**).

sd(insc(0),6,0) Disk device at logical unit number 0 on SCSI ID 6 on the first (0) integrated SCSI adapter (**insc**).

sd(insc(0),3,6) Seventh (6) diskette drive at SCSI ID of 3 on the first (0) integrated SCSI adapter.

sd(cisc(0),2,4) Fifth (4) magneto-optical device at SCSI ID of 2 on the first (0) Ciprico SCSI adapter.

Formatting Disk Devices

This section gives procedures to format the disk devices to receive data. Only readable and writable devices can be formatted; for example, a CD-ROM device is not writable, so cannot be formatted. Furthermore, any removable medium must be properly inserted in the drive and the drive door closed. Also, a diskette must be write-enabled before its surface can be prepared.

Formatting includes these operations:

- Installing a disk label.
- Creating system areas.
- Installing a bootstrap.
- Performing surface analysis.

Follow this path through **sysadm**:

```
Device -> Disk -> Physical Disk -> Manage
```

Figure 10-1 shows the Diskman Main Menu:

```
=======================================================

              Diskman Main Menu

        1.  Physical Disk Management Menu

        2.  Logical Disk Management Menu

        3.  Disk Mirror Management Menu

        4.  File System Management Menu

        5.  System Installation Menu

=======================================================

Enter a ? or <number>? for help, ^ to return to previous menu, or q to quit
Enter choice: [5]
```

Figure 10-1 *Diskman Main Menu*

- Select option 1, Physical Disk Management Menu.
- Select option 5, Format a Physical Disk.

Figure 10-2 shows the Physical Disk Formatting Menu.

```
=========================================================

              Physical Disk Formatting Menu


        1.  Install a Disk Label on a Physical Disk

        2.  Create DG/UX System Areas on Physical Disk

        3.  Install a Bootstrap on a Physical Disk

        4.  Perform Surface Analysis on a Physical Disk

        5.  All Formatting Steps

=========================================================

Enter ? or <number>? for help, ^ to return to previous menu, or q to quit.
Enter choice: [5]
```

Figure 10-2 *Physical Disk Formatting Menu*

In most cases, you can select option 5, All Formatting Steps. In some circumstances, however, you may not want to create system areas and to perform surface analysis. Therefore, you may select each formatting step explicitly from this menu. Typical dialogues are given for each selection in this menu, including all formatting steps.

NOTE: Before you make a selection, make sure that all device power cables are connected as described in your hardware documentation. If a device is disconnected, power down the system before you connect the device. Then reboot the system using the procedures in Chapter 12.

Installing a Disk Label on a Physical Disk

Choose option 1, Install a Disk Label on a Physical Disk, to install a particular disk label on a physical disk. A disk label contains the disk layout (such as tracks per cylinder, bytes per sector, and so on). All disks except SCSI disks and HADA disk arrays, which have generic labels, must contain this information so that the system can access them. A typical dialogue follows. Read the explanation following the dialogue before answering any questions.

```
=====================================================

Install a Disk Label on a Physical Disk

=====================================================

Enter the physical disk specification in DG/UX common format: sd(cisc(0),0,0) ↵

Physical disk sd(cisc(0),0,0) does not have a disk label.
Do you want to install one? [yes] ↵
A disk label has been installed on the physical disk sd(insc(0),0,0).

Press New Line to continue.
```

If the disk is not a SCSI disk or HADA disk array device, you see the following display:

```
=============================================================

              Physical Disk Types

1. Model 6442: full-height ESDI, 327 MB
2. Model 6555: full-height ESDI, 648 MB
3. Model 6661: half-height ESDI, 330 MB
4. Model 6541 or 6542 SMD, 1066 MB
5. None of the above

Enter the type of the disk that is to be labeled:

=============================================================
```

If you select one of the numbers above, the disk label will be installed. If you select option 5, None of the above, the following prompts appear:

```
Enter the total cylinders per drive:
Enter the OS visible cylinders per drive:
Enter the tracks per cylinder:
Enter the sectors per track:
Enter the bytes per logical sector:
Enter the bytes in mfg defect information:
Enter the bytes per unformatted sector:
Enter mfg defect information start sector:
Enter the number of relocation areas:
Enter the sectors per relocation area:
Enter the interleave:
Enter the head skew:
Enter the cylinder skew:
Enter the head group skew:
Enter the spares per track:
Enter the bytes per data preamble:
Enter the bytes per id preamble:
Enter the base head for volume:
Enter the bytes in gap 1:
Enter the bytes in gap 2:
Does the drive use SMD extended addressing?
Does the drive have a final short sector?
```

Refer to your hardware documentation for information on answering the preceding questions. If the disk label installation was successful, the following typical message appears:

```
A disk label has been installed on the physical disk sd(insc(),5,0)
```

Creating DG/UX System Areas on a Physical Disk

The operating system needs system areas to describe the logical disks on a physical disk. The system areas include a Primary System Area (PSA), a Logical Disk Piece Table that describes the logical disks that are on the physical disk, and a place to store the bootstrap. Another system area is the Bad Block Table which the system uses to remap bad blocks on the disk medium.

NOTE: If you are formatting a diskette or an optical disk, and you want to create only one file on the medium, to conserve space, you may not want to create system areas. In the case of the diskette, the diskette is so small that the system areas would occupy a relatively large amount of space. Bypass this step and for more information go to the section on special considerations for devices.

From the Physical Disk Formatting menu, choose option 2, Create DG/UX System Areas on Physical Disk, to create system areas on a physical disk.

In response to the prompts, press New Line to accept the defaults.

```
=========================================================

Create DG/UX System Areas on Physical Disk

=========================================================

Enter the physical disk specification in DG/UX common format: [sd(cisc(0),0,0)] ↵

Physical disk sd(cisc(0),0,0) does not have DG/UX system areas.
Do you want to create them? [yes] ↵

CAUTION: This operation will destroy any data on the physical
         disk sd(cisc(0),0,0).

Do you want to continue? [yes] ↵

The physical disk sd(cisc(0),0,0) is 366592 blocks in size.

Enter the number of blocks to allocate for the remap area: [189] ↵
DG/UX system areas have been created on the physical disk sd(cisc(0),0,0).

Press New Line to continue.
```

When you create system areas on a disk, **diskman** calculates the size of the remap area based on the overall size of the disk.

Installing a Bootstrap on a Physical Disk

You need to install a bootstrap only if the disk contains bootable software. You have to allocate space for the bootstrap, regardless of whether or not you actually install one.

The **diskman** program has access to the low-level bootstrap program that is used to boot the DG/UX system image. During formatting, this program is written to disk by **diskman.** Choose option 3, Install a Bootstrap on a Physical Disk, and press New Line to accept the default responses to the prompts.

```
=========================================================

Install a Bootstrap on a Physical Disk

=========================================================

Enter the physical disk specification in DG/UX common format: [sd(cisc(0),0,0)] ↵

Physical disk sd(cisc(0),0,0) does not have an up-to-date bootstrap.
Do you want to install one? [yes] ↵
Enter the pathname of the bootstrap image file: [/usr/stand/boot.aviion] ↵
A bootstrap has been installed on the physical disk sd(cisc(0),0,0).

Press New Line to continue.
```

Performing Surface Analysis on a Physical Disk

Surface analysis checks the unallocated portion of a disk's surface for flaws, and upon finding any, builds a bad block table that identifies flawed areas, rendering them usable. Surface analysis takes about 20 minutes per 100 Mbytes, depending on your physical disk model and CPU. An average is about one hour per physical disk. Data General disk devices do not require surface analysis. Therefore, to save time, you may choose not to perform surface analysis.

If this disk is on a controller that performs hardware bad block remapping, you will be informed and asked whether surface analysis is still desired. If it is still desired, select option 4, Perform Surface Analysis on a Physical Disk. The resulting dialogue should look like this:

```
=======================================================

Perform Surface Analysis on a Physical Disk

=======================================================

Enter the physical disk specification in DG/UX common format: [sd(cisc(0),0,0)] ↵

CAUTION:  Surface analysis will destroy any data on the unallocated
          regions of the physical disk sd(cisc(0),0,0).

Do you want to want to run surface analysis? [yes] ↵

You have the option of running either all three test patterns or a single test
pattern.

Do you want to run all the test patterns? [yes] ↵
Beginning surface analysis ...


Surface analysis finished.
0 blocks were found and remapped.


Press New Line to continue.
```

Generally, it is unnecessary to run surface analysis.

Performing All Formatting Steps

Rather than selecting each formatting step separately, you can perform all formatting steps sequentially in an automated fashion.

The following dialogue is typical for previously formatted physical disks for which you want to reformat some aspects.

Choose option 5 to perform all formatting steps.

```
=====================================================

All Formatting Steps

=====================================================

Enter the physical disk specification in DG/UX common format: sd(cisc(0),0,0) ↵
The physical disk must be deregistered for this operation.

Do you want to deregister it? [yes]  ↵
Physical disk sd(cisc(0),0,0) has been deregistered.

Physical disk sd(cisc(0),0,0) already has a disk label.
Do you want to re-install one? [no] yes ↵
A disk label has been re-installed on the physical disk sd(cisc(0),0,0).


Physical disk sd(cisc(0),0,0) already has DG/UX system areas.
Do you want to re-create them? [no] yes ↵

CAUTION:  This operation will destroy any data on the physical
          disk sd(cisc(0),0,0).

Do you want to continue? [no] yes ↵


The physical disk sd(cisc(0),0,0) is 366592 blocks in size.

Enter the number of blocks to allocate for the remap area: [189]
DG/UX system areas have been re-created on the physical disk sd(cisc(0),0,0).


Physical disk sd(cisc(0),0,0) does not have an up-to-date bootstrap.
Do you want to install one? [yes] ↵
Enter the pathname of the bootstrap image file: [/usr/stand/boot.aviion] ↵
A bootstrap has been installed on the physical disk sd(cisc(0),0,0).


CAUTION:  Surface analysis will destroy any data on the unallocated
          regions of the physical disk sd(cisc(0),0,0).

Do you want to run surface analysis? [yes] ↵

You have the option of running either all three test patterns or a single test
pattern.

Do you want to run all the test patterns? [yes] no ↵
Beginning surface analysis ...


Surface analysis finished.
0 blocks were found and remapped.


Press New Line to continue.
```

Special Consideration for Devices

This section contains information on the requirements and uses for each device that you install:

- Tape drive
- CD-ROM (Compact Disk-Read-Only Memory)
- Multiple-read/write magneto-optical disk drive
- Diskette drive

Tape Drive

A tape drive requires no formatting. It is assumed that is has been correctly installed using the appropriate hardware documentation.

When you add a new tape drive to your hardware configuration, it must be recognized in the kernel. Refer to Chapter 12 for information on ensuring that the newly added tape drive is listed in the system file.

Using CD-ROM, Magneto-Optical, and Diskette Devices

For CD-ROM, magneto-optical, and diskette devices, make sure you have the proper terminator on the last device in a chain of SCSI devices. Also, excessive cable length could result in problems when trying to access the last device on the chain.

If you have multiple CD-ROM devices, each must have a unique SCSI ID. Multiple magneto-optical or diskette devices, on the other hand, may be clustered so that as many as four share the same SCSI ID. If you put multiple units on the same SCSI ID, you will need to use device specifications that include third arguments (for logical unit numbers). For example, the following specifications represent three 5.25-inch diskette devices at SCSI ID 3:

```
sd(insc(0),3,0)
sd(insc(0),3,1)
sd(insc(0),3,2)
```

Using the CD-ROM Device

To create multiple file systems, refer to Chapter 3 for instructions on creating logical disks, registering the physical disk, and mounting file systems. You must specify the device file name and a **cdrom** file type. To learn the device file name, refer to the section on locating a device's short name after the system is booted.

Before removing a compact disk from the device, first unmount the file system. If you had originally registered the device (as you do with disks that contain named logical disks), you also need to deregister the device. Refer to a later section for instructions on removing removable media from a drive.

When you add a new CD-ROM device to your hardware configuration, it must be recognized in the kernel. Refer to Chapter 12 for information on ensuring that the newly added CD-ROM device is listed in the system file.

Using the Magneto-Optical Device

Decide how many file systems you want on the device. If you want to create multiple file systems, refer to Chapter 3 for instructions on creating logical disks, registering the physical disk, and mounting file systems.

If you want to create only one file system on the disk, you do not need to create system areas. (System areas are needed to manage the location of multiple logical disks on a physical disk.) If you plan only one file system on the disk, to conserve disk space, do not create system areas.

Use the following command format to make the entire disk one file system:

mkfs *device-file-name*

To learn the device file name, refer to the section on locating a device's short name after the system is booted.

For example, to create a file system for the device at **/dev/pdsk/1**, use this command line:

```
# mkfs /dev/pdsk/1 ↵
```

You will also need to mount the file system. Refer to Chapter 3 for instructions on mounting file systems. You will need to specify the device file name and a **dg/ux** file type.

Before removing an optical disk from the device, first unmount the file system. If you had originally registered the device (as you do with disks that contain named logical disks), you also need to deregister the device. Refer to a later section for instructions on removing removable media from a drive.

You may use a magneto-optical device as a tape as well as a file system. You can write to the disk using **tar**(1) or **cpio**(1) just as you can to tape.

You cannot use **sysadm** to create a backup on an optical disk. To make a backup on an optical disk, use the **dump2**(1M) command. If you do this, you are limited to one disk.

When you add a new magneto-optical device to your hardware configuration, it must be recognized in the kernel. Refer to Chapter 12 for information on ensuring that the newly added magneto-optical device is listed in the system file.

Using the Diskette Device

The 3.5-inch diskette device supports 720 Kbytes and 1.44 Mbytes formats. The 5.25-inch diskette device supports 360 Kbytes, 720 Kbytes, and 1.2 Mbytes formats. Use **diskman** to format diskettes or obtain pre-formatted diskettes from your Data General representative.

Before removing a diskette from the drive, first unmount the file system. If you had originally registered the device (as you do with disks that contain named logical disks), you also need to deregister the device. Refer to a later section for instructions on removing removable media from a drive.

Removing a diskette that is mounted and/or registered results in an error message such as:

```
From System:
The file system on device device-name sealed,
      Status nnnnnnn
        Run fsck to restore
```

or an error message such as:

```
File system is no longer fault tolerant
```

where *nnnnnn* is a status number.

If either kind of error occurs, you must re-insert the diskette and run **fsck**(1M) to recover. Refer to *Managing the DG/UX™ System* for information on using **fsck**(1M).

When you add a new diskette device to your hardware configuration, it must be recognized in the kernel. Refer to Chapter 12 for information on ensuring that the newly added diskette device is listed in the system file.

Using a Diskette as a File System

Decide how many file systems you want on the diskette. If you want to create multiple file systems, refer to Chapter 3 for instructions on creating logical disks, registering the physical disk, and mounting file systems.

If you want to create only one file system on the disk, you do not need to create system areas. (System areas are needed to manage the location of multiple logical disks on a physical disk.) If you plan only one file system on the disk, to conserve disk space, do not create system areas.

Use the following command format to make the entire diskette one file system:

mkfs *device-file-name*

To learn the device file name, refer to the section on locating a device's short name after the system is booted.

For example, to create a file system for the device at **/dev/pdsk/1**, use this command line:

```
# mkfs /dev/pdsk/1 ↵
```

You will also need to mount the file system. Refer to Chapter 3 for instructions on mounting file systems. You will need to specify the device file name and a **dos** file type.

Using the Diskette Device as a Tape

Instead of using the diskette device as a file system, you may use it as a tape. Like a tape, you can write to the diskette using **tar**, **cpio**, **dump**, **dump2**, or **dd**. Refer to the device as **/dev/rpdsk** or **/dev/pdsk**. Before you can use the device this way, you must unmount it (if mounted) and deregister it (if registered). Refer to a later section for instructions on removing removable media from a drive.

Because diskettes do not have tape marks (as found on magnetic tape), you cannot use multiple diskettes with **tar**, **dump**, **dump2**, or **dd**. The **cpio** command, on the other hand, allows you to use multiple diskettes when archiving files.

You cannot use **sysadm** to dump to a diskette device. You may dump to a diskette from the shell using the **dump2**(1M) command, but you are limited to one diskette.

If you boot the system while a diskette with a **tar** or **cpio** format file is in the diskette device, you may see an error having to do with the physical file table. Ignore this error. If the diskette device is registered as a file system when you try to write to it as a file (with **tar** or **cpio**), you will receive the error message, `conflict on open`. Deregister the device.

Removing Media from Drives

Before you remove an optical disk, compact disk, or diskette from a drive, you should first make the file system on the device inaccessible to users. You might send a broadcast message using the **wall**(1M) command to all logged in users to inform them. Then, perform these procedures:

- Unmount the file system
- Deregister the device

Unmounting the File System

This operation unmounts a mounted file system. Unmounting a file system detaches it from the file system hierarchy, making it inaccessible to users. You need to know the file system on the device medium that you want to remove from the drive.

Follow this path through **sysadm**:

```
File System -> Local Filesys -> Unmount
```

A typical dialogue follows:

```
File System(s) to Unmount: /accounts
OK to perform operation? [yes] ↵

/accounts is unmounted.
```

If you receive the message, `Device Busy`, you cannot unmount the file system. This message suggests that a user's working directory is in the file system you are trying to unmount. Or, a user is executing a program or accessing data from a file located in the file system you are trying to unmount. You can find out the identity of the offending user by issuing this command:

fuser -Fu *file-system*

You can inform the user to exit the file system. With no users on the file system, you should be able to unmount it.

Deregistering the Medium

Use this option to deregister a registered physical disk, thus making any logical disks on it inaccessible.

- From the Diskman Main Menu, select option 1.
- From the Physical Disk Menu, select option 1.
- From the Physical Disk Registration Menu, select option 2 to deregister a physical disk.

The dialogue follows:

```
=======================================================

Deregister a Physical Disk

=======================================================

Enter the physical disk specification in DG/UX
common format: [sd(insc(0),0,0)] ↵

Physical Disk sd(insc(0),0,0) is deregistered.

Press New Line to continue. ↵
```

You can now remove the optical disk, compact disk, or diskette from the drive.

Locating Device File Names after the System is Booted

For every device on your system, there is a short name created at boot time that you can use to refer to the device. You can find a table showing the long name/short name pairs in the file **/etc/devlinktab**. For example, a SCSI tape device has this specification in DG/UX common format:

```
st(insc(0),4,0)
```

With this name in mind, you can locate the device's entry (which has a very similar long name) in the **devlinktab** file. The example below shows the device's entry (as well as the comment lines from the file that have the table headers).

```
# directory      short      long
#
/dev/rmt         0          st(insc@7(FFF8A000),4,0)
```

From this entry, you know that the short name for **st(insc(0),4,0)** is **/dev/rmt/0**. The device is a tape device, so it has a no-rewind version too, which is **/dev/rmt/0n**.

You use a device's short name when you need to write to it (like a tape) or mount it as a file system.

For more information on device naming, see Appendix D.

End of Chapter

Chapter 11
Building Kernels

A kernel is an executable program that provides operating system services to all other programs running on the system. The kernel runs directly on the hardware, managing access to physical devices as well as user requests and application programs.

If you just finished installing your basic DG/UX system (using procedures in *Installing the DG/UX™ System*), you have already built and booted a kernel for an OS server or a stand-alone system. By default, this kernel is named **/dgux.aviion**. Over time, you may need to rebuild and boot subsequent kernels for these purposes:

- Adding OS clients
- Adding physical devices or device drivers
- Adding software packages that require special variable tuning
- Improving system performance by tuning special variables

This chapter covers these topics:

- Kernel building
- Checking the kernel

Building a Kernel

To build a kernel, follow this path through **sysadm**:

```
System -> Kernel
```

The Kernel Menu offers three options, which are:

Auto Configure Builds a kernel that includes default variable values and recognition of all hardware devices installed at standard locations. A kernel built using this option is referred to as an "auto-configured" kernel.

Build Builds a kernel that is the same as the one built with Auto Configure except that you can customize the tunable variable values and add entries for non-standard devices in the system file, which is the basis of the kernel. A kernel built using this option is referred to as a "custom" kernel.

Reboot Shuts down the system completely (except for the hardware itself) and restarts the operating system. To activate a particular kernel, you must reboot the system.

Building a custom kernel means that you directly edit the system file to reflect your hardware and software configuration. Each time you build a new kernel (custom or auto-configured) for your computer (you create a new system file), it is probed for the existence of attached devices at standard locations. A complete list of devices at standard locations is provided in **/usr/etc/probedevtab**. The probe produces a list of names of located devices and inserts the list at the top of the system file, which you can edit.

If you have a device that is jumpered at a nonstandard location, it will not be located by the probe (the device is not listed in the **probedevtab** file). See **probedev**(1M) for more information on the device probe process.

Also, you can tune the performance of your system by setting variables. Refer to *Managing the DG/UX™ System* for a complete list of tunable variables.

This chapter illustrates custom kernel building for this host and the OS client of this host. You will build a kernel for "this host" in most situations except when the OS server must build an OS client's first-time kernel. After an OS client is operational, however, it can build its own kernel, if necessary.

Building a Custom Kernel for This Host

From the Kernel Build Menu, select the "Build" option.

You are prompted for the system configuration file name.

```
System configuration file name: [aviion] ↵
```

The name of the system file distinguishes this host's system file (and derived kernel) from all other system files and kernels. Pressing New Line accepts the default filename extension, which is **aviion**, to yield **system.aviion**. If the file name does not exist, a new one is created. The

new file contains a list of attached devices that result from the device probe and a series of concatenated prototype system file fragments, named **system.*package*.proto**, where *package* can be **nfs**, **tcpip**, or any other software package name. These files are located in **/usr/src/uts/aviion/cf**.

If that system file already exists, it will not be overwritten; rather, it is opened for editing. For instance, if you added a device to your hardware configuration since the system file was last generated, you must add the device name explicitly to the list. The system file is located in **/usr/src/uts/aviion/Build**, which is a symbolic link to **/var/Build**.

If you develop a special configuration file-naming convention, specify the desired name instead of accepting the default.

The next prompt follows:

```
[system.aviion] Correct? [yes] ↵
```

Verify the selected system file name.

You are next asked if you are building a kernel for this host or an OS client of this host:

```
Build for this host or for OS client(s) of this host: [this host] ↵
```

The host name is often used to name a kernel for this host. However, if you build more than one kernel, you will need to develop a convention to distinguish them. Since you are building a custom kernel for `this host`, press New Line to accept the default.

The final prompt allows you to specify an editor to use for editing the system file.

```
Editor: [/usr/bin/vi] ↵
```

Finally, you use the **vi** editor to edit the system file. (Refer to Appendix A for a summary of **vi** commands.) Alternatively, you can specify the editor of choice if you know its location on your system.

Editing the System Configuration File for This Host

The system file is lengthy; it spans several screens. Configuration variables that do not apply to your system are ignored (or commented out). A line that is commented out contains a # in the first column. You are advised to read the contents of the system file before you edit it.

The two sections that you will be most concerned with are:

- Automatically configured hardware devices
- General configuration variables

Figure 11-1 shows an excerpt from a system file named **system.aviion**.

```
#--------------------------------------------------------------
# Automatically Configured Hardware Devices:
#
# These hardware devices were found on the system by probedev(1M).
#
#
    kbd()             ## Workstation keyboard
    grfx()            ## Workstation graphics display
    lp()              ## Integrated parallel line printer controller
    duart(0)          ## Dual-line terminal controller (number 0)
    inen()            ## Integrated Ethernet controller
    sd(insc(),0)      ## SCSI disk 0 on Integrated SCSI adapter
    sd(insc(),1)      ## SCSI disk 1 on Integrated SCSI adapter
    st(insc(),4)      ## SCSI tape 4 on Integrated SCSI adapter
    st(insc(),6)      ## SCSI tape 6 on Integrated SCSI adapter

#--------------------------------------------------------------
# General Configuration Variables:
#
#
# The NODE variable controls your nodename for uname(1) and uucp(1).
#

  NODE                "aviion"
```

Figure 11-1 *Excerpts from System File for This Host*

Automatically Configured Hardware Devices

Each configured device is specified using the DG/UX common device specification format with an associated brief description. The # symbol signifies a comment, which is ignored. For more information on the device specification format, refer to Appendix D.

NOTE: If you can anticipate the addition of more devices of the same type to your hardware configuration in the future, rather than rebuilding the kernel to acknowledge the new device, you can edit your system file in advance. An example follows:

```
sd(insc(),*)        ## All SCSI disks on first Integrated SCSI adapter
```

The asterisk (*) is a pattern-matching metacharacter that recognizes all SCSI IDs for disk devices attached to the first integrated SCSI adapter.

General Configuration Variables

NODE refers to the host name of your computer. Your computer's host name, for example **"aviion,"** is automatically supplied. It is used by the **uname**(1) command to report the name and other attributes of the current system. Also, the **uucp**(1) command uses the node name for performing file transfers on UNIX systems. It is also presented as part of the login banner message when you log in to your system. If you set up the TCP/IP package, the node name also corresponds with the host name that you supply during TCP/IP setup. The node name is restricted to a maximum of 31 characters.

Previous releases of the DG/UX system required that you explicitly set the **DUMP** variable in the system file. Starting with release 5.4, the **DUMP** variable is set through a new command named **dg_sysctl**(1M) that you type at the shell. Refer to Chapter 12 and the **dg_sysctl**(1M) manual page for more information.

The master files located in the **/usr/etc/master.d** directory contain a complete list of general configuration variables and default values. You can override a given default by setting the variable to the desired value in the system file. Read about the tunable variables in *Managing the DG/UX™ System*. You are accepting the defaults by not changing any values assigned to variables in the system file.

The system file contains a series of concatenated software package prototype system file fragments. If you have just loaded a new software package on your system, check its release notice for information on possible variable tuning. If you need to tune a variable, either enter or modify it in the appropriate location in the system file.

Building the Kernel for This Host

You must link the new kernel (named **dgux.aviion** or **dgux.goose**, for example) to **/dgux** before you can use it. If you do not link the new kernel to **/dgux**, the existing kernel (if one exists) remains linked to **/dgux**.

```
Link the new kernel to /dgux? [yes] ↵
```

Accept the `yes` default to link the new kernel to **/dgux**. The kernel build resumes. The system file is then configured (a C language source file is generated) and a new kernel is produced. See the **config**(1M) manual page for more information on kernel building.

```
Continue with the build?  [yes] ↵
Configuring system...
Building kernel...
Successfully built dgux.aviion.
Linked /dgux.  You must reboot in order for this
kernel to take effect.

Enter a number, a name, ? or <number>? for help, <NL> to redisplay menu,
^ to return to previous menu, or q to quit:
```

A common source of many kernel build failures is the inadvertent absence of a comment symbol (#) which is used to flag descriptive notes to be ignored. Be sure you comment out all text that is to be ignored. Check your spelling and verify the device names specified in DG/UX common device specification format.

After the new kernel is built, the bootable kernel file is in the **/** directory as **dgux.***name* (**dgux.aviion** and **dgux.goose** as examples) and is also linked to **dgux**. The new kernel will take effect only by booting it.

CAUTION: Do not try to boot the OS client kernel from the OS server. Only the OS client can boot its kernel.

Go to Chapter 12 for information on booting a kernel.

Building a First-Time Custom Kernel for an OS Client

The OS server should build an OS client its first-time kernel only to establish an OS client's basic operation. After an OS client is operational, however, it can build its own kernel, if necessary. Refer to Chapter 8 for information on building a first-time kernel for an OS client.

Checking the Kernel

You may have multiple kernels resulting from several kernel-building sessions. However, only one kernel (linked to **/dgux**) can be booted at a time. You can find out which kernel is linked to **/dgux** by using the following shell command:

```
# ls -i /dgux* ↵
4051 /dgux*              4049 /dgux.installer*   4058 /dgux.starter*
4088 /dgux.gyramax       4051 /dgux.goose
```

The example output shows that the inode numbers match for **/dgux** and **/dgux.goose**. Therefore, **dgux.goose** is the name of your current system file.

You can also check the current values for variables that are set in your kernel. Checking these values may be helpful in determining whether or not you need to build a new kernel. For example, if you add a SCSI tape device to your configuration, you must build a new kernel only if the kernel does not already include an entry for the device. Use the **sysdef**(1M) shell command to find out current kernel values. An excerpt of typical output follows.

```
# sysdef ↵
# Configured devices
#
kbd()
grfx()
inen()
duart()

# Configuration variables
#
NODE "gyramax"
NETBOOTDEV "inen()"
ROOTFSTYPE &init_mount_network_root
SWAPDEVTYPE &init_start_network_swap
NFS
```

The list does not include a SCSI tape device; therefore, to include the SCSI device, you must create a new kernel.

You can also peruse the system files on which kernels are based in this location: **/usr/src/uts/aviion/Build**.

Where To Go Next

After you build a new kernel, you must boot it in order to make the new kernel take effect. Refer to Chapter 12 for this information.

End of Chapter

Chapter 12
Booting and Logging In to the DG/UX System

The primary reasons for booting the DG/UX system are to gain access to a new kernel that has just been built and to reinitialize the system following a panic or failure. Refer to *Managing the DG/UX™ System* for information on failure detection and recovery.

These topics are covered:

- Using **sysadm** to reboot
- Using the System Control Monitor (SCM) to boot
- Automatic rebooting
- DG/UX run levels
- Shutting down the system
- Booting messages
- Logging in to the DG/UX system

Regardless of the method used to boot the DG/UX system, the basic boot command is the same. It identifies a boot file located on a local device (disk or tape) or the local area network (LAN) device. See "Specifying a Boot Path" later in this chapter for boot path syntax.

Whenever you reboot your computer, any running processes are killed. If the OS server reboots, attached OS clients may continue to operate normally or they may reboot automatically. OS client reboots, however, do not affect the OS server.

If multiple users are logged in to the system, before you reboot, send a broadcast message to warn that you are rebooting the system and give a grace period. See the **wall**(1M) manual page for information on sending a broadcast message to all users logged in the system. Make sure all users are logged out before proceeding. An example of such a warning follows:

```
# /etc/wall ↵
Five minutes until the system will be rebooted. Please log off.
<Ctrl-D>
```

A system reboot displays messages that describe the system initialization stages followed by a login prompt.

Using Sysadm

Follow this path through **sysadm**:

```
System -> Kernel -> Reboot
```

In the following example, the kernel **/dgux** on the **root** logical disk located on **sd(insc(0),0,0)** is booted automatically to a run level of 3. However, you could specify another run level (refer to "DG/UX Run Levels" for a list of options).

```
Boot path: [sd(insc(0),0,0)root:/dgux -3] ↵
All currently running processes will be killed.
Are you sure you want to reboot the system? [yes] ↵
```

The default boot path is pre-set at the factory or can be reset through the **dg_sysctl**(1M) command or the SCM Configuration Menu. You can override the default to establish the correct boot path for your system. Refer to "Specifying a Boot Path" for the exact syntax. For example, a boot path of **inen() −3**, which specifies a boot to run level 3 over the network, could be established for an OS client. You are informed that all currently running processes will be killed when you reboot. Press New Line to reboot the system.

Booting from the SCM

A system that has been shut down will operate at the SCM level. An OS client will boot its kernel from the SCM. From the SCM prompt, you issue a boot command using this syntax:

SCM> **b** *boot-path*

where:

b stands for boot.

boot-path identifies the boot file located on a boot device. Refer to "Specifying a Boot Path" in this chapter for the specific syntax.

Example:

 # b sd(insc(0),0,0)root:/dgux −3 ↵

This command boots **/dgux** contained in the logical disk **root**. The boot device is a SCSI disk attached to the first integrated SCSI adapter whose SCSI ID is 0. The system will boot to a run level of 3, which enables multi-user mode. Refer to the section on "DG/UX Run Levels" in this chapter for more information on run levels.

Rather than typing this lengthy command each time you want to reboot the system, you can establish a shortcut using instructions given in the next section.

Establishing a Default Path Using the SCM Menus

You can establish a shortcut for booting your DG/UX system by setting a default boot path for the kernel. The shortcut allows you to type only the **b** alias in lieu of a longhand boot command from the SCM prompt. See *Using the AViiON® System Control Monitor (SCM)* for more information.

1. Type the **f** (format) command at the SCM prompt. You will see the display of the SCM Configuration Menu.

```
SCM> f↵

View or Change System Configuration

1. Change boot parameters
2. Change console parameters
3. Change mouse parameters
4. Change printer parameters
5. View memory configuration
6. Change testing parameters
7. Return to previous screen

Enter choice(s) ->
```

2. Type the option **1**, followed by New Line, to change boot parameters. The following menu will be displayed.

```
Change Boot Parameters

1 Change system boot path
2 Change diagnostics boot path
3 Change data transfer mode [BLOCK]
4 Return to previous screen

Enter choice(s) ->
```

3. Type the option **1** again, followed by New Line. The system displays the current boot path. Enter **y** followed by New Line to change the path.

```
System boot path = [sd(insc(0),0,0)root:/dgux.installer]

Do you want to modify the boot path? [N]
```

4. Enter the new boot path at the prompt. Two examples of boot devices are given. The first one is for a system having a SCSI boot disk. Your system will boot to a run level of 3 automatically. The second is for a system having an ESDI boot disk.

```
Enter new system boot path -> sd(insc(0),0,0)root:/dgux -3 ↵
Enter new system boot path -> cied()root:/dgux ↵
```

   At the next prompt, press new Line to confirm the path that you specified. Examples follow:

```
System boot path = [sd(insc(0),0,0)root:/dgux -3]
System boot path = [cied( ),0)root:/dgux]

Do you want to modify the system boot path? [no] yes ↵
```

If you want to reboot immediately, answer **yes** to the next question. Otherwise, accept the N default, and return to the SCM prompt.

```
Do you want to boot? [N] ↵
```

5. The Change Boot Parameters menu is displayed. Choose option 3.

   The View or Change System Configuration menu is displayed. Choose option 7 to go to the SCM prompt.

From this point on, with the system boot path set correctly, you can type the single-letter alias **b** followed by a New Line at the SCM prompt to boot the kernel.

Setting the Boot Path With the dg_sysctl(1M) Command

You can use the **dg_sysctl**(1M) command from the shell to set your system's booting behavior following a panic situation. You can establish whether or not the DG/UX system automatically reboots and you can identify the device to which a system dump is directed. The syntax follows:

dg_sysctl [**-t**] [**-r** *reboot-state*] [**-b** "*boot-path*"] [**-d** *autodump-state*] [**-f** "*dump-device*"]

where:

-t makes a temporary change. It does not persist following the system reboot. The default specifies a permanent change.

-r sets the system's reboot behavior. If *reboot-state* is **auto**, then the system will automatically reboot after a panic. If *reboot-state* is **halt**, the system will not automatically reboot after a panic. The default is **halt**.

-b sets the system's boot path. The boot path must conform to the SCM boot syntax. *boot-path* is the path to use when the system is rebooted. Be sure to surround the boot path with double quotation marks (" "). Refer to "Specifying a Boot Path" for exact syntax.

-d sets the system's *auto-dump* behavior. If *autodump-state* is **auto**, then the system will attempt to dump to *dump-device* after a panic. A tape must be present in the drive at all times in the event of a dump. If *autodump-state* is **skip**, then the system will not attempt to dump to *dump-device* after a panic. If *autodump-state* is **ask**, then the system will ask if you wish to take a system dump after a panic. The default is **ask**.

-f sets the system's *dump-device* to be used during a panic. The default value for the DUMP variable is set in the **/usr/etc/master.d/dgux** file and can be reset in the system configuration file. For an AViiON 4000 OS server, for example, the default boot tape device is

st(insc(0),4,0). For an OS client, a dump is submitted over the network **inen()** to the OS server's dump device.

dump-device is the device expressed in the DG/UX common device specification format, to which a panic dump is written.

Example 1:

```
# dg_sysctl -r auto -b "/dgux -3" -d auto -f  "st(insc(0),4)" ↵
```

This command enables the auto-reboot-on-panic functionality. It reboots the kernel located at **/dgux** to a run level of 3. It also enables the auto-dump-on-panic functionality, and dumps to **st(insc(0),4)**, which is a SCSI tape device having an SCSI ID of 4 that is attached to the first (0) integrated SCSI controller.

Example 2:

```
# dg_sysctl ↵
```

With no arguments, this command reports the current values for the arguments, which include whether or not the boot path is permanently changed, whether the system will automatically autoboot following a panic or halt and query you about taking a dump, the boot path, and the dump device.

Make sure you have an appropriate tape inserted in the drive of the dump device. If the system panics, a dump will be written to the specified device automatically.

Rebooting With a Specific Boot Path (reboot Command)

You can use the **reboot**(1M) command from the shell to explicitly shut down the system before rebooting the kernel. The shutdown procedure is normally logged to **/etc/wtmp**. Refer to "Shutting Down Your System" in this chapter for details for what occurs during a shutdown. The syntax follows:

reboot [*options*] [*boot-path*]

options can be specified to alter the default behavior of the command, which is to reboot from the boot path used when the system was last booted. Refer to the **reboot**(1M) manual page for information on options.

boot-path is the path to use when the system is rebooted. By default, this is the boot path used when the system was last booted. The current boot path can be changed with the **dg_sysctl**(1M) command. If no *boot-path* is specified, the default boot path is assumed. The default boot path could have been set at the factory or you could have reset through the **dg_sysctl**(1M) command or the SCM Configuration Menu. Refer to "Specifying a Boot Path" in this chapter for information on boot path syntax.

Example:

reboot

The **reboot** command with no options assumes default conditions which is the boot path used when the system was last booted. The default boot path could have been set at the factory or you could have reset through the **dg_sysctl**(1M) command or the SCM Configuration Menu.

Specifying a Boot Path

You can boot from either a disk or tape device or a local area network (LAN). Each specification is covered in the next two sections.

Disk and Tape Device

The syntax for booting a file located on a disk or tape device follows:

device-specification logical-disk:/directory/file [*−run-level*]

NOTE: Do not space between any of these fields except between *file* and [*−run-level*].

where:

device-specification, expressed in DG/UX common device specification format, identifies the boot device. Refer to Appendix E for information on device naming. You can specify devices using the short or long form.

logical-disk contains the executable kernel image.

/directory/file is the pathname of the executable kernel image on the logical disk.

−run-level specifies the run level to which your system boots automatically. You should set your run level in the **/etc/inittab** file and use the boot path to override the pre-established default. Refer to the section "DG/UX Run Levels" for information on the run levels and see *Managing the DG/UX™ System* for information on editing the **inittab** file.

Refer to the section "DG/UX Run Levels" in this manual for more information.

Example 1:

sd(insc(),0)root:/dgux −3 ↵

This boot path specifies the **/dgux** kernel contained on the logical disk **root**. The boot device is a SCSI disk attached to the first integrated SCSI adapter whose SCSI ID is 0. The system will boot to a run level of 3, which enables multi-user mode.

Example 2:

sd@7(FFF8A000),0)root:/dgux -1 ↵

This boot path specifies the **/dgux** kernel contained on the logical disk **root.** The boot device is a SCSI disk attached to the integrated SCSI adapter whose device code is 7, nonstandard address is FFF8A000, and SCSI ID is 0. The system will boot to a run level of 1, which is administrative mode.

Example 3:

b ↵

This boot command automatically boots the file identified in the autoboot path that you set through the SCM Configuration Menus.

Example 4:

st(insc(0),4,0) ↵

This boot path specifies the first logical file (0) on a 150-megabyte QIC tape whose SCSI ID is 4.

Local Area Network

An OS client will boot its kernel via a network.

The syntax for booting a file over a LAN follows:

network-controller-device OS-client-Internet-address:/directory/file

NOTE: Do not space between any of these fields.

where:

network-controller-device is the name of the device used to connect the OS client to the local area network. Valid controllers are expressed in the form: *controller-typecontroller-num* where *controller-type* can be **inen** (integrated Ethernet controller), **hken** (Interphase VME Ethernet controller, or **dgen** (Data General second generation integrated Ethernet controller). *controller-num* for controller type **inen** can be only **0**; valid controller numbers for **hken** are **0-7**; for **dgen, 0-1.** Valid examples are **inen0, hken3,** or **dgen0.** Consult your network administrator.

OS-server-Internet-address, is the Internet address of the OS server. Consult with the network system administrator. An example of an Internet address is **128.223.2.1.** In this example, **128.223** refers to the network number, **2** refers to the subnet number, and **1** refers to the host number. The dots are field separators.

/directory/file identifies the kernel image to boot.

Example:

inen()128.223.2.1:/srv/release/PRIMARY/root/bob/dgux ↵

The OS client boots the file located in **/srv/release/PRIMARY/dgux/bob/dgux** over the network controller **inen()** by way of the OS server's Internet address. In this example, **bob** is the OS client host name.

DG/UX Run Levels

The DG/UX system can operate in one of several run levels. Table 12-1 lists them.

Table 12-1 DG/UX Run Levels

Run Level	Description
Q or **q**	Consult the **/etc/inittab** file and re-execute the appropriate commands that correspond with the current run level.
S or **s**	Single user mode. The system default file systems (**/swap**, **/**, and **/usr**) are mounted. No processes are running except those of the system administrator who is logged in as **root**.
i	Installation mode. All local file systems are mounted and essential processes are running. The **installman**(1M) command is invoked to perform installation tasks. Refer to the **installman**(1M) manual page for more information.
1	Administrative mode. This mode is used to install and remove software, and to perform administrative tasks, such as checking file systems and doing backups. System processes are running, and all file systems are mounted. Only **sysadm** and **root** are recognized login names at this run level.
2	Multi-user mode. This is the mode with the most service for those who are not operating in a network environment and who are not running the DG/UX X Window System software, Release 4. All local file systems are mounted.
3	Multi-user mode. This is the mode required to run DG/UX X Window System software, Release 4. It is also the mode with remote file system sharing (NFS) and network services.
4	User-defined level. Used mainly for applications.
5	Stops the system and goes to the SCM. This state is functionally equivalent to bringing the system to state S and issuing the **halt**(1M) command. Refer to the **halt**(1M) manual page for more information.
6	Stops the system and reboots the default boot path. This state is functionally equivalent to bringing the system to state S and issuing the **reboot**(1M) command. Refer to the **reboot**(1M) manual page for more information.

OS client-server configurations operate at run level 3 since network services are required to support OS clients.

Normally, you will establish a run level when you boot your kernel from the SCM, by way of an autoboot path that you establish through the SCM Menus, or a **sysadm** menu selection. However, you can explicitly change run levels from the shell using this command format:

init *run-level*

Shutting Down Your DG/UX System

Before you perform a shutdown, notify all system users of your intention and that they must log off. You may choose to send users a broadcast message to warn of the shutdown and to give a grace period. See the **wall**(1m) manual page. Make sure all users are logged off before proceeding.

From the system console at the shell prompt, type these commands:

```
# cd / ↵
# shutdown -g0 -y ↵
```

The first command changes the current directory to **/** (root). The second command specifies a grace period of 0 seconds between sending an automatic message to users logged in warning them of the shutdown and the beginning of the shutdown. Also, it includes an affirmative response (yes) to start the shutdown. Otherwise, a confirmation request would be explicitly issued, requiring a response. You will see the following messages:

```
Shutdown started.          Wed June 16 11:08:57 DST 1991
.
.
.
Shutdown is complete.
```

Your system has been shut down.

You next type:

```
# halt ↵
```

Control goes to the SCM. After the SCM prompt is displayed, you will know that the DG/UX system has shut down.

Booting Messages

When you boot your system, messages are displayed that describe the stages of system initialization. The exact text of the messages will depend on whether you are booting an OS server (or stand-alone system) or OS client and the run level to which you are booting.

An example of booting the system following the **sysadm** route follows:

```
System -> Kernel -> Reboot
```

A typical dialogue follows:

```
Boot path: [sd(insc(0),0,0)root:/dgux -3] ↵
All currently running processes will be killed.
Are you sure you want to reboot the system? [yes] ↵
```

Figure 12-1 shows typical rebooting messages.

```
Booting sd(insc(0),0,0)root:/dgux -3
DG/UX Bootstrap Release 5.4
Loading image ..................................................
DG/UX System Release 5.4, Version generic
Using 16 Megabytes of physical memory
Found 2 processor(s)
Configuring devices ..............................

    Checking local file systems ...
    Current date and time is Tue Apr 23 14:30:24 EDT 1991
    Checking system files .....
    Enabling automatically pushed STREAMS modules .....
    Linking short names for /dev device nodes ...
    Loading terminal controllers ....
    Starting disk update daemons .....
    Mounting local file systems ..........
    Checking for packages that have not been set up ...
    Starting miscellaneous daemons ...
    Starting TCP/IP network interfaces .........
    Starting system logging daemon ....
    Starting NIS services ......
    Starting NFS lock services ......

    NOTE:  Pausing for 15 seconds to allow remote systems to
           reclaim NFS locks.

    Starting batch services ....
    Starting line printer scheduler ....
    Saving ex(1) and vi(1) temporary files ....
    Starting NFS services .....
    Starting TCP/IP daemons ........
    Mounting NFS file systems .....

    NOTE:  See /etc/log/init.log for a verbose description of the
           system initialization process.
```

Figure 12-1 *Typical Messages for OS Server Booting to Run Level 3*

Figure 12-2 shows a typical boot command and booting messages at an OS client. In this example, bob is the name for the OS client and steve is the name for the OS server.

```
SCM> b inen() -3 ↵

Booting inen()
Local Ethernet address is 08:00:1C:1F:03:77
Local Internet address is 128.222.14.32 or 80DE0E20 hex
Trying server at 128.222.14.31 or 80DE0E1F hex for TFTP transfer
00166464

DG/UX Bootstrap Release 5.4

Boot:  inen (0)
Local Ethernet address is 08:00:1C:1F:03:77
Local Internet address is 123.228.3.19
Broadcasting request for a boot server ...
Host name: bob
Using steve:/srv/release/PRIMARY/root/bob as root
Loading /dgux ........................................
DG/UX System Release 5.4, Version generic
Using 16 Megabytes of physical memory
Found 1 processor(s)
Processor 1 is running
Configuring devices ..................................


    Checking local file systems ...
    Current date and time is Tue Apr 23 14:30:24 EDT 1991
    Checking system files .....
    Enabling automatically pushed STREAMS modules .....
    Linking short names for /dev device nodes ...
    Loading terminal controllers ....
    Starting disk update daemons .....
    Mounting local file systems ...........
    Checking for packages that have not been set up ...
    Starting miscellaneous daemons ...
    Starting TCP/IP network interfaces .........
    Starting system logging daemon ....
    Starting NIS services ......
    Starting NFS lock services ......

    NOTE:  Pausing for 15 seconds to allow remote systems to
           reclaim NFS locks.

    Starting batch services ....
    Starting line printer scheduler ....
    Saving ex(1) and vi(1) temporary files ....
    Starting NFS services .....
    Starting TCP/IP daemons ........
    Mounting NFS file systems .....

    NOTE:  See /etc/log/init.log for a verbose description of the
           system initialization process.
```

Figure 12-2 *Typical Messages for OS Client Booting to Run Level 3*

Logging In to the DG/UX System

The login prompt appears after the DG/UX system has finished booting. The user login that you enter will depend on what you are doing. OS clients booting their first-time kernels need to set up software packages and mount file systems before their computers are operational. Therefore, OS clients should log in as **sysadm** to gain access to **sysadm**, which offers a menu-driven set of system administration procedures.

Those who do not need to perform continued setup activities, can log in with their user logins to begin real work. Make sure that you have been assigned a user login and password (see Chapter 4 for details).

The type of login prompt you see will depend on whether you are using a graphics monitor or an alphanumeric display terminal. Choose the appropriate section for continued procedures.

Logging in with a Graphics Monitor

If your system console is a graphics monitor and you have installed the X Window System package, you will see the following login box displayed on the screen (see Figure 12-3).

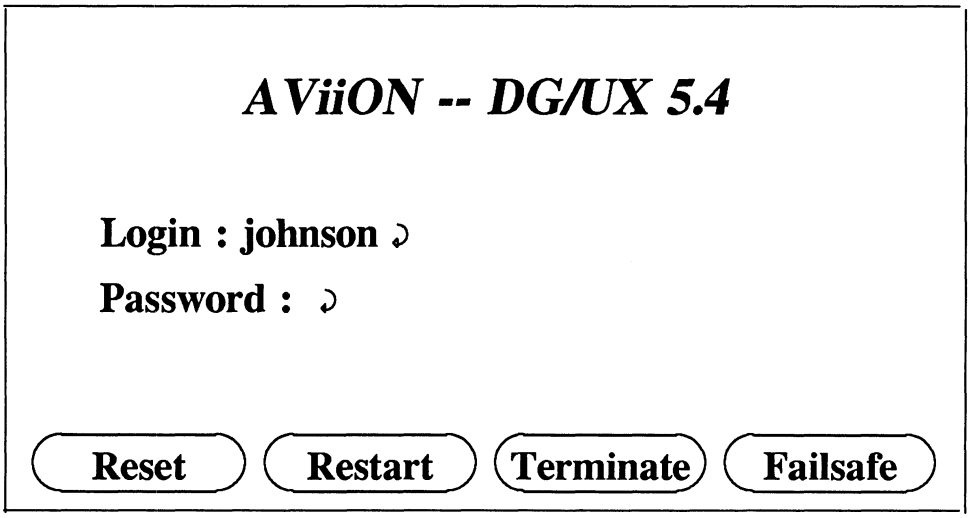


Figure 12-3 *Login Screen for a Graphics Monitor*

The four buttons along the bottom of the screen are provided as aids to controlling the X server, which is the program that controls the display of information to the screen.

Reset Resets the X server. The X server is not terminated.

Restart Terminates the X server, then restarts it.

Terminate Terminates the X server, giving control to a single VT100 terminal emulation screen.

Failsafe Is a toggle button, which limits startup to a single xterm window.

NOTES: If you choose to exit the X Window environment and continue the procedures in a single VT100 terminal screen environment, move the cursor to the "Terminate" button and click.

To return to the X Window environment, type **xdm** at the login prompt. The X Window environment will be restored.

You will be presented with a single window and a login prompt.

Logging in with an Alphanumeric Display Terminal

Figure 12-4 shows the login display for an alphanumeric terminal.

```
bob
DG/UX Release 5.4 AViiON
Console login: johnson ↵
Password:  ↵
Copyright (c) Data General Corporation, 1984-1991
All Rights Reserved

============================================================
#                                                          #
#                       WARNING                            #
#                                                          #
#                                                          #
# ACCESS TO AND USE OF THIS SYSTEM IS RESTRICTED TO        #
#               AUTHORIZED INDIVIDUALS                     #
#                                                          #
#          Data General AViiON DG/UX System                #
#                                                          #
============================================================

#
```

Figure 12-4 *Alphanumeric Display Terminal as a System Console*

In this example, **bob** is the host name, which was specified during TCP/IP setup.

The login banner appears, followed by the # shell prompt. You can access the many commands and utilities offered by DG/UX to do your job.

Where To Go Next

OS clients booting their first-time kernels need to set up software packages and mount file systems before their computers are operational. Package setup procedures are provided in Chapter 7. Procedures to add local and remote file systems are given in Chapter 3.

If your computer is fully operational, begin to use the DG/UX system facilities. Refer to the "Related Documents" section at the end of this manual for a list of manuals.

End of Chapter

Appendix A
Editing with vi

Vi operates in two modes: input mode and command mode. In input mode, anything you type (except for the **Esc** character) goes into the file as text. In command mode, any keys you type are commands that tell **vi** things like where to move the cursor, what text to delete, what text to move, and so on.

When you first invoke **vi**, it is in command mode. Table A-1 shows the commands you can use in command mode. These commands are case sensitive.

Table A-1 Summary of vi Commands

Command	Description
<Ctrl-F>	Move forward one screen.
<Ctrl-B>	Move backward one screen.
G	Move to the bottom of the file.
j	Move the cursor down one line.
5j	Move the cursor down five lines.
k	Move the cursor up one line.
/Tunable ↵	Move to the next occurrence of **Tunable**.*
?NODE ↵	Move to the previous occurrence of **NODE**.
l	Move ahead one character.
h	Move back one character.
w	Move ahead one word.
b	Move back one word.
dw	Delete to the end of this word.**
dd	Delete this line.
8dd	Delete this line and the next 7.
u	Undo my last change.
i	Enter input mode (described below), inserting text at the cursor.
I	Enter input mode, inserting text at the beginning of the line.
o	Enter input mode, opening a new line below the current position.
<Ctrl-R>	Redraw the screen.
:w ↵	Write (save) my changes to the file.
:w! ↵	Force changes to the file, overriding permissions.***
:q ↵	Exit **vi** (only works if you have not changed the file).
:q! ↵	Exit **vi** without writing my changes.
:wq ↵	Writes the changes to the file and quits the editor.

* searches in **vi** wrap; that is, if **vi** has not found the requested string when it reaches the end of the file (or the beginning, if you search with **?**), **vi** will then search from the top (or the bottom) of the file back to the cursor position.

** signifies that words are marked by punctuation as well as spaces.

*** means that the **w!** forced write works only if you are the superuser or the owner of the file.

In **vi** input mode, anything you type goes into the file as text until you press the **Esc** key, which returns you to editing mode. For more information on **vi**, see *Using the DG/UX™ Editors* or the **vi**(1) manual page.

End of Appendix

Appendix B
Displaying the Layout of Your Physical Disks

By displaying the disk's layout of your physical disk periodically as you create logical disks, you can determine how to allocate remaining disk resources. Use of the **diskman** option "Display a Physical Disk's Layout" can be particularly helpful for creating multiple logical disk pieces on the same logical disk or across multiple physical disks.

Procedures for accessing this option follow.

1. From the Diskman Main Menu, select option 1, Physical Disk Management Menu.

2. Select option 3, Display a Physical Disk's Layout.

3. When prompted, enter the name of the disk device whose layout you want to examine.

Figure B-1 shows an example of the appearance of a typical preloaded physical disk.

```
System areas on physical disk sd(insc(0),0,0):

       Area Name              LD Piece      Physical Disk     Size
                               Number        Address
                                             of Area

      Primary System Area      ......              0          8
    System Bootstrap Area      ......              8        500
    Secondary System Area      ......            508          8
  Primary Bad Block Table      ......            516          5
        Primary LDP Table      ......            521          9
     Bad Block Remap Area      ......            530        315
Secondary Bad Block Table      ......            845          5
      Secondary LDP Table      ......            850          9
                     swap      1 of 1          40859      50000
                     root      1 of 1          90859      40000
                      usr      1 of 1         130859     240000
            usr_opt_aview      1 of 1         370859      10000
              usr_opt_X11      1 of 1         380000     105000
             (Free Space)      ......         485000     650922

Total physical disk size: 1135922 blocks.
Unallocated space: 650922 blocks.

Do you want to switch to another physical disk instead? [no] ↵

Press New Line to continue.  ↵
```

Figure B-1 *Typical Physical Disk Layout for Preloaded System*

The first column "Area Name" identifies both physical disk attributes (created during physical disk preparation) and logical disks that are either already preloaded on your boot disk device or created when you installed your system using the release tape. In this example, logical disks—**swap, root, usr, usr_opt_aview,** and **usr_opt_X11**—are listed. The "Free Space" entry in this column identifies the address and the amount of remaining space (in blocks) that is contiguous to the last logical disk.

The second column lists the the number of the piece and the number of pieces that compose the logical disk. Most logical disks will be arranged in one piece (1 of 1). However, creating a logical disk in pieces may be advantageous if you have limited resources (see Chapter 2).

The third column lists the beginning address of an area. Each area and logical disk are arranged contiguously on the physical disk; however, a contiguous arrangement isn't required. The final column lists the size (in blocks) of each area and logical disk. In this example, the free space and unallocated space both report the same amount of available blocks that remains for your use. However, if your physical disk contained multiple areas of noncontiguous free space, multiple "Free Space" reports would be given. The "Unallocated Space" is all free space reported on the physical disk.

As another example, Figure B-2 shows the customized arrangement of logical disks on a physical disk.

```
System areas on physical disk sd(insc(0),0,0):


       Area Name              LD Piece      Physical Disk     Size
                               Number        Address
                                             of Area

      Primary System Area      ......              0          8
    System Bootstrap Area      ......              8        500
    Secondary System Area      ......            508          8
  Primary Bad Block Table      ......            516          5
        Primary LDP Table      ......            521          9
     Bad Block Remap Area      ......            530        315
Secondary Bad Block Table      ......            845          5
      Secondary LDP Table      ......            850          9
                     swap      1 of 1          40859      50000
                     root      1 of 1          90859      40000
                      usr      1 of 1         130859     240000
            usr_opt_aview      1 of 1         370859      10000
              usr_opt_X11      1 of 1         380000     105000
             (Free Space)      1 of 1         485000      29659
                 accounts      1 of 1         534659     100000
                 fredware      1 of 2         634659      35000
             (Free Space)      1 of 1         669659      60000
              spreadsheet      1 of 1         675659      90000
                 fredware      2 of 2         765659      37000
                    tools      1 of 1         802659     180000
             (Free Space)      ......         982659     153263

Total physical disk size: 1135922 blocks.
Unallocated space: 262922 blocks.
Do you want to switch to another physical disk instead? [no] ↵

Press New Line to continue.  ↵
```

Figure B-2 *Typical Customized Physical Disk Layout*

Notice that in this example, the three instances of "Free Space," which are noncontiguously arranged over the physical disk. The "Unallocated Space" reports the total of the "Free Space" reports across the physical disk. **fredware** is an example of a logical disk arranged in two pieces.

Using **diskman** to display the layout can be particularly useful when determining logical disk resources that span physical disks. As an example, let's say you need to create a logical disk for another package **brand_x**. This product's release notice specifies that 620,000 blocks are needed. If you wanted to create a logical disk for **brand_x** on **sd(insc(0),0,0)** shown in the preceding figure, you can see that there are only 262,922 blocks of available free space —

that's a shortage of 357,078 blocks on **sd(insc(0),0,0)**. Perhaps, you could find sufficient contiguous space for **brand_x** on another physical disks. You can check the free space on remaining physical disk devices in your configuration using the same procedure. Table B-1 shows a strategy you might use to arrange the logical disk pieces across multiple physical disks.

Table B-1 Arranging Logical Disk Pieces on Multiple Physical Disks

Logical Disk Piece	Physical Disk	Size (in Blocks)
1 of 6	**sd(insc(0),0,0)**	20,000
2 of 6	**sd(insc(0),0,0)**	60,000
3 of 6	**sd(insc(0),0,0)**	153,263
4 of 6	**sd(insc(0),1,0)**	128,912
5 of 6	**sd(insc(0),1,0)**	128,912
6 of 6	**sd(insc(0),2,0)**	128,913
TOTAL		620,000

End of Appendix

Appendix C
Determining tty Lines for Asynchronous Ports

You can add terminals and printers to your DG/UX system all at once or one at a time. If you add them all at once, you assign the same characteristics to all asynchronous ports on your computer. In other words, the RS-232/422 ports on the computer unit, the ports on any Systech asynchronous controllers (VAC/16 controllers), and the ports on the cluster controllers for any Systech asynchronous distributed host adapters (VDA/128 and VDA/255 host adapters) would all have the same characteristics. If you add terminals or printers one at a time, you may assign different characteristics to different ports. See Chapters 5 and 6 for information on adding terminals and printers through **sysadm**.

NOTE: In this appendix the expression "terminal line controller" will be used to refer to the RS-232/422 interfaces for the ports on the computer unit, the VAC/16 controller, and the VDA/128 and VDA/255 host adapters. Furthermore, the term "ports" on a VDA host adapter actually refers to the ports on the cluster controllers which are connected to the host adapter.

To add a terminal or printer to your DG/UX system, you need the following information:

- The tty line number that DG/UX assigned to the terminal line controller port where the terminal or printer is connected. A terminal may be connected either directly to the port or indirectly through a modem and dial-up telephone line to a modem connected to the port.
- The type of terminal or printer (for example, VT100 terminal or PostScript printer).

In this appendix you will obtain this information by performing the following tasks:

- Determine the order in which the terminal line controllers are listed in the system configuration file.
- Use the AViiON System Diagnostics to obtain information about terminal line controllers.
- Determine the tty line number for each device on a terminal line controller.

Using Worksheets

The following worksheets are referenced throughout this appendix:

- Terminal Line Controllers Worksheet
- Device Worksheet for RS/232/422 Ports on the Computer Unit
- Device Worksheet for a VAC/16 Controller
- Device Worksheet for a VDA Host Adapter

Go to Appendix E for these worksheets. Each Device Worksheet for a VDA Host Adapter contains space for recording 32 ports. So, if you have a VDA host adapter, you will need two additional sheets for a VDA/128 host adapter and six additional sheets for a VDA/255 host adapter.

Preceding each worksheet is a sample worksheet that has been filled out for an AViiON 5000 computer with three terminal line controllers: RS-232/422 terminal/modem port on the computer unit, one VAC/16 controller, and one VDA/128 host adapter. The host adapter has one 8-line cluster controller and seven 16-line cluster controllers, providing ports for a maximum of 120 serial devices and one parallel printer. These terminal line controllers have the device names given below, and these device names are listed in your DG/UX system configuration file in the relative order shown below:

duart() terminal/modem port on computer unit
syac() VAC/16
syac(1) VDA/128

Determining the Order of DG/UX Device Names for Terminal Line Controllers

You will need to know the relative order of the terminal line controller device names in your DG/UX system's configuration file.

Change to the current directory of the system configuration file and use the **more**(1) command to view its contents. An example follows:

```
# cd /usr/src/uts/aviion/Build ♪
# more system.aviion ♪
```

NOTE: You may have assigned a different name to your system configuration file. View the appropriate one. Refer to Chapter 11 for information on building kernels and editing system configuration files.

Figure C-1 shows an excerpt from a system file named **system.aviion.**

```
#-----------------------------------------------------------------
# Automatically Configured Hardware Devices:
#
# These hardware devices were found on the system by probedev(1M).
#
#
    kbd()               ## Workstation keyboard
    grfx()              ## Workstation graphics display
    lp()                ## Integrated parallel line printer controller
    duart(0)            ## Dual-line terminal controller (number 0)
    inen()              ## Integrated Ethernet controller
    syac()              ## Systech terminal line controller
    syac(1)             ## second Systech terminal line controller
    sd(insc(),0)        ## SCSI disk 0 on Integrated SCSI adapter
    sd(insc(),1)        ## SCSI disk 1 on Integrated SCSI adapter
    st(insc(),4)        ## SCSI tape 4 on Integrated SCSI adapter
    st(insc(),6)        ## SCSI tape 6 on Integrated SCSI adapter
```

Figure C-1 *Arrangement of Automatically Configured Devices in the System Configuration File*

Viewing the system file, you can see the relative order of the terminal line controllers. The ordering follows:

duart(0) position 1
syac() position 2
syac(1) position 3

The order in which terminal line controllers are configured and listed in the system configuration file is important because it affects the assignment of tty line numbers.

On the Terminal Line Controllers Worksheet in Appendix E, record the relative order of your terminal line controller devices.

Using the AViiON System Diagnostics to Get Information About Terminal Line Controllers

This section describes how to use your AViiON System Diagnostics to help you get the following information:

- Board type (VAC/16, VDA/128, or VDA/255) for the **syac** terminal line controller device name.
- Cluster controller type (8-line, 16-line) for each controller connected to a VDA host adapter.
- VAC/16 or cluster controller port to which a specific terminal or printer is connected.

NOTE: The person installing your computer hardware should have recorded this information on Device Worksheets that were supplied in the manual *Setting Up and Installing VMEbus Options in AViiON® Systems*. If these worksheets are available from the hardware installer and they have all the information listed above, then go to the section "Determining the tty Line for Terminal Line Controller Ports" later in this appendix. If these worksheets are not available or they are missing information, continue with this section.

If your AViiON computer was preloaded with the DG/UX system software, the system diagnostics is also on the boot disk. You boot the diagnostics as a stand-alone program. If your computer came with the DG/UX system on a release tape, you can boot the diagnostics from the AViiON System Diagnostics cartridge tape.

To boot the system diagnostics, you must first shut down the DG/UX system. If the SCM prompt appears on the screen, the DG/UX system is shut down and you should go to the appropriate section below for booting the diagnostics from disk or tape. If the DG/UX system is running, shut it down as described below.

Shutting Down Your DG/UX System

Before you perform a shutdown, notify all system users of your intention and that they must log out. You may send users a broadcast message to warn of the shutdown and give a grace period. See the **wall**(1m) manual page. Make sure all users are logged out before proceeding.

From the system console at the shell prompt, type these commands:

```
# cd /↵
# shutdown -g0 -y ↵
```

The first command changes the current directory to **/** (root). The second command specifies a grace period of 0 seconds between sending an automatic message to users logged in warning them of the shutdown and the beginning of the shutdown. Also, it includes an affirmative response (yes) to start the shutdown. Otherwise, a confirmation request would be explicitly issued, requiring a response. You will see the following messages:

```
Shutdown started.          Wed June 16 11:08:57 EST 1991
.
.
.
Shutdown is complete.
```

Your system has been shut down.

You next type:

 # halt -q ↵

This command halts the processors, by-passing a confirmation request. After the SCM prompt is displayed, you will know that the DG/UX system has shut down.

Go to the appropriate section that follows for booting the diagnostics from disk or tape.

Booting System Diagnostics from Disk

To boot the system diagnostics from disk, use the following command syntax:

b *disk-name***usr:/stand/diags**

where:

b stands for boot.

disk-name is the DG/UX common device specification for the boot disk. Table C-1 contains a list of default boot disk (and tape) devices. Select one that is appropriate. Also, you may want to refer to Appendix E for information on device names.

usr:/stand/diags specifies logical disk **usr** which contains the bootable diagnostics program that is located in the file named **/stand/diags**.

CAUTION: If the **usr** logical disk were created in pieces, you cannot boot system diagnostics from disk. You must, instead, boot from the release tape; those instructions are given in the next section.

A typical command for booting system diagnostics from disk follows:

SCM> **b sd(insc(0),0,0)usr:/stand/diags** ↵

Table C-1 Default Boot Tape and Disk Devices

AViiON Series Model	Boot Tape	Description	Boot Disk	Description
100, 200, 300, 400, 3000 and 4000	**st(insc(0),4)**	Tape device at SCSI ID 4 on the first (0) integrated SCSI adapter (**insc**).	**sd(insc(0),0,0)**	Disk device at SCSI ID 0 on the first (0) integrated SCSI adapter (**insc**).
5000, 6000, 7000, and 8000	**st(cisc(0),4)**	Tape device at SCSI ID 4 on the first (0) Ciprico SCSI adapter (**cisc**).	**sd(cisc(0),0,0)**	Disk device at SCSI ID 0 on the first (0) Ciprico SCSI adapter (**cisc**).
7000 and 8000*	**st(hada(0),4)**	Tape device at SCSI ID 4 on the first (0) High-Availability Disk Array Subsystem (**hada**).	**sd(hada(0),0,0)**	Disk device at SCSI ID 0 on the first (0) High-Availability Disk Array Subsystem (**hada**).
7000 and 8000*	**st(hada(0),4)**	Tape device at SCSI ID 4 on the first (0) High-Availability Disk Array Subsystem (**hada**).	**da(hada(0),6)**	Disk array at unit number 6 on the first (0) High-Availability Disk Array Subsystem (**hada**).
530 and 4600	**st(ncsc(0),4)**	Tape device at SCSI ID 4 on the first (0) NCR integrated SCSI adapter (**ncsc**).	**sd(ncsc(0),0,0)**	Disk device at SCSI ID 0 on the first (0) NCR integrated SCSI adapter (**ncsc**).

* means that if you have a 7000 or 8000 AViiON series model, and if you have the High Availability Disk Array Subsystem, you may have these default boot tape and disk devices.

Go to the section "Determining the Terminal Line Controller Type and Cluster Controller Type" below.

Booting System Diagnostics from Tape

To boot the system diagnostics from tape, use the following command syntax:

b *tape-name*

where:

b stands for boot.

tape-name is the DG/UX common device specification for the boot disk. Table C-1 contains a list of default boot tape (and disk) devices. Select one that is appropriate. Also, you may want to refer to Appendix E for information on device names.

A typical command for booting system diagnostics from tape follows:

```
SCM> b st(insc(0),4) ↵
```

Go to the section "Determining the Terminal Line Controller Type and Cluster Controller Type" below.

Determining the Terminal Line Controller Type and Cluster Controller Type

Figure C-2 shows the first screen of the System Diagnostics:

```
System Diagnostics
Revision xx.xx mm/dd/yy hh:mm:ss

Initializing Operating System for System Diagnostics

   Sizing host adapters on VME Bus:

   Checking for VME ESDI Boards at addresses:
            ffffef00 (hex)      Not found
            fffff100 (hex)      Not found
            fffffb00 (hex)      Not found
            fffffd00 (hex)      Not found

Checking for VME SCSI Boards at addresses:
            fffff300 (hex)      Found
            fffff500 (hex)      Not found
            fffff700 (hex)      Not found
            fffff900 (hex)      Not found

Checking for VME LAN Boards at addresses:
            ffff4000 (hex)      Not found
            ffff5000 (hex)      Not found

Press New Line to Proceed ♪

Checking for VME Synchronous I/O Boards at addresses:
            55b00000 (hex)      Not found
            55b10000 (hex)      Not found
            55b20000 (hex)      Not found
            55b30000 (hex)      Not found
            55b40000 (hex)      Not found
            55b50000 (hex)      Not found
            55b60000 (hex)      Not found
            55b70000 (hex)      Not found

Checking for VME Asynchronous I/O Boards at addresses:
            60000000 (hex)      Found
            60020000 (hex)      Found
            60040000 (hex)      Not found
            60060000 (hex)      Not found
            60080000 (hex)      Not found

Press New Line to Proceed  ♪
```

Figure C-2 *First Screen of System Diagnostics*

Information about the licensing of the diagnostics appears first. When you are ready to continue, proceed as follows:

1. Press New Line to clear the screen.

   The system diagnostics program looks for (sizes) components in your computer, and displays information about each component that it finds, as shown in the sample screen that follows.

   Press New Line to go to the next screen.

2. Next, the diagnostics initializes the components that it found and displays the beginning of the sample screen below.

   Press New Line to accept the default responses to the prompts as shown in the next screen.

```
16384 Kbytes system memory
15109 Kbytes memory available for test
PROM revision xx.xx
Dual CPU System (Motorola 88100 CPU Rev x)
2 Instruction Caches (Motorola 88200 CMMU Rev x)
2 Data Caches (Motorola 88200 CMMU Rev x)
Initializing Virtual Console
Initializing Real Time Clock
Initializing VME Async I/O Board 0
Initializing VME Async I/O Board 1
Initializing VME SCSI Board 0
Initializing Parallel Printer
Initializing Duart 0
Initializing Duart 1

Run with instruction caches on (Y/N) [Y]? ↵
Enable Parity Checking for instructions (Y/N) [Y]?  ↵
Run with data caches on (Y/N) [Y]? ↵
Enable Parity Checking for data (Y/N) [Y]? ↵
Current time is 16:15 Monday, April 1, 1991. Is this correct (Y/N) [Y] ↵
```

3. Next, the system diagnostics lists the peripheral devices connected to initialized controllers, as shown in the sample screen that follows.

```
Sizing Peripherals....

VME SCSI Board 0:
            Unit 0:  Microp 1578-15 UPDG02 Disk Drive found
            Unit 1:  Microp 1578-15 UPDG02 Disk Drive found
            Unit 3:  TEAC 5.25 Floppy (LUN 2) Disk Drive found
            Unit 4:  Archive Viper 150 21247-045 Tape Drive found

VME Async Board 0:
            16-line VME Async Board
            Model                  = HPS-6236
            Firmware P/N           = 90-070408-8-01A

VME Async Board 1:
            128-line VME Host Adapter 0
            Model                  = HPS-6945
            Firmware P/N           = 90-070154-6-05A

            Sizing Cluster Controller Network
            ---------------------------------
            Net ID = 01(hex): HPS-7082-020 (Ready)
            Net ID = 02(hex): HPS-7088-020 (Ready)
            Net ID = 03(hex): HPS-7088-020 (Ready)
            Net ID = 04(hex): HPS-7088-020 (Ready)
            Net ID = 05(hex): HPS-7088-020 (Ready)
            Net ID = 06(hex): HPS-7088-020 (Ready)
            Net ID = 07(hex): HPS-7088-020 (Ready)
            Net ID = 08(hex): HPS-7088-020 (Ready)

Press New Line to Proceed
```

On the Terminal Line Controllers Worksheet in Appendix E, record each VME async board type listed under "Sizing Peripherals...." The names that the diagnostics system uses for the different terminal line controller boards are as follows:

16-Line Async Board	VAC/16
128-line VME Host Adapter	VDA/128
255-line VME Host Adapter	VDA/255

For example, using the sample screen above, you would write "VDA/16" in the Board or Port Type column for Board No. 0, and "VAC/128" in the Board or Port Type column for Board No. 1.

On the Terminal Line Controllers Worksheet in Appendix E, record the number of lines for each cluster controller address (Net ID) listed under a VME host adapter. The model numbers for the 8-line and 16-line cluster controllers are as follows:

HPS-7082	8-line cluster controller
HPS-7088	16-line cluster controller

For example, using the sample screen above, for Board 0, under "Cluster Controllers" you would write "8" in the "No. Lines" column for Address 01, and write "16" in the "No. Lines" column for Addresses 02 and 08.

When you have finished recording this information, press New Line. If more sizing information appears, repeat these procedures appropriately. Continue this process of viewing and recording sizing information until the Main Menu appears.

Before continuing, you should transfer information from the Terminal Line Controllers Worksheet to the tty worksheets as described below. If you do not do this, you will have difficulty determining the tty line assigned to each port on these terminal line controllers.

VAC/16

For each VAC/16 controller in your computer, record the board number and its device name on a Device Worksheet for a VAC/16 Controller.

VDA Host Adapter

For each VDA host adapter in your computer, record its board number, device name, and device type (VDA/128 or VDA/255) on a Device Worksheet for a VDA Host Adapter. For each cluster controller connected to the host adapter, record the cluster address. If the cluster controller is an 8-line controller, draw a vertical arrow from the cluster address you entered down to the dashed line. This indicates that only nine ports are available on this controller. If the controller is a 16-line controller, draw the vertical line through the dashed line all the way down to the bottom of the column to indicate that 16 ports are available.

You now have enough information to determine which tty line your DG/UX system assigns to a specific port on any of the terminal line controllers in your computer. In addition, if you know what type of terminal or printer is connected to each terminal line controller port, you can add these devices to your DG/UX system without having to complete the following section. In this case, you should record the device type for each port on the appropriate tty worksheet.

Next, exit to the SCM by selecting option 4 on the Main Menu, and go to the "Determining the tty Line for Each Terminal Line Controller Port" section later in this appendix. If you do not know which type of terminal or printer is connected to each terminal line controller port, continue on to the next section.

Determining the Port Where a Specific Device Is Connected

While you do not need to know the port to which a specific device is connected to determine the tty line for that port, you will need this information when you add a terminal or printer to your DG/UX system, or if you need to troubleshoot problems with a terminal line controller. This section will help you obtain that information.

Figure C-3 shows the System Diagnostics' Main Menu that appears on the system console.

```
System Diagnostics
Revision: xx.xx

Data General Corporation
Proprietary Use Only


          Main Menu

1.  Run Acceptance test
2.  View Tools Menu
3.  Display help screen
4.  Exit to SCM

Enter choice [1]:
```

1. At the Main Menu, select **2** from the View Tools Menu and press New Line.

   The Tools Menu shown below appears.

```
          Tools Menu

1.  Format diskettes
2.  Run tape adjustment utility
3.  View Graphics Tools Menu
4.  Test network connection (TDR)
5.  Run keyboard test
6.  Run mouse test
7.  View Terminal Test Menu
8.  Display help screen
9.  Return to main menu

Enter choice [9]:
```

2. Select **7** from the View Terminal Test menu and press New Line.

   The Terminal Test Menu shown below appears.

```
                    Terminal Test Menu

          1.  Start scrolling characters set test
          2.  Start lines of characters test
          3.  Start keyboard echo test
          4.  Start port ID message test
          5.  Auto port identification
          6.  Terminate a test
          7.  Show executing tests
          8.  Display help screen
          9.  Return to Tools menu

          Enter choice [9]:
```

3. Select **4** from the Start port ID message test and then press New Line.

   The system displays the following prompt:

   ```
   Board number (0,1, [ALL])?  ↵

   Running selftest on VME Host Adapter 0 (approximately 30 seconds),
    please wait....
   ```

4. Press New Line.

   This message occurs the first time you select a host adapter for testing. On each terminal or printer connected to a port on a VAC/16 controller or VDA host adapter in your computer, you will see a port ID message similar to the one below. This message lists the board number, cluster address (VDA host adapter only), and the port number for the device displaying the message.

   ```
   128-line VME Host Adapter 0, Cluster address: 01, port: 0
   ```

NOTE: If you have an 8-line cluster controller without a parallel printer connected to port 8, or the printer is not on line and ready, a message appears telling you this. If such a message appears, simply press the Esc key to skip the test on that port.

5. Look at the message displayed on each device for which you do not know the port number, and determine the board number, cluster address, and port number for the device. On the appropriate Device Worksheet under the specified board number, cluster address (if applicable), and port number, record the type of device (for example, D460 terminal or Model 6640 parallel laser printer) displaying the message. Also record a description that locates the device (for example, office 3B, connector #1356). Worksheets are located in Appendix E.

   While the port ID messages are being displayed, the Terminal Test Menu appears on the system console screen.

6. After you finish recording the information for each terminal or printer, select **6** from the Terminate a test menu and then press New Line.

7. The following prompt is displayed:

```
Board number (0,1, [ALL])? ↵
```

8. Press New Line to accept the default response to the next prompt: From the Terminal Test Menu, press New Line again to select the Return to Tools menu.

9. From the Tools Menu, press New Line to select Return to main menu.

10. At the Main Menu, select **4** to exit to the SCM followed by New Line.

Determining the tty Line for Terminal Line Controller Ports

This section discusses how the DG/UX system allocates tty lines to asynchronous terminal line controller ports. With this information, you can determine the tty lines that your DG/UX system assigns to each such port on your computer.

How the DG/UX System Allocates tty Lines

When you installed the DG/UX system, it automatically assigned a specific tty line to each port on each terminal line controller in your computer. Table C-2 lists the number of tty lines that DG/UX allocates to each type of terminal line controller.

Table C-2 Number of tty Lines Allocated to Terminal Line Controllers

Terminal Line Controller	Lines Allocated
RS-232/422 ports on computer unit	1
VAC/16 controller	16
VDA/128 or VDA/255 host adapter	256

Notice that the DG/UX system allocates 256 tty lines to a VDA/128 host adapter. Since a VDA/128 host adapter has only 128 ports, this means that only the first 128 tty lines are actually assigned to specific ports on a VDA/128; the remaining 128 tty lines are unused.

The DG/UX system assigns a specific tty line to each port sequentially in the order in which the names of the terminal line controllers are listed in your system's configuration file. It starts with **tty00**. For each subsequent port, the numerical portion of the tty name is increased by one.

Let's look at an example of how the DG/UX system assigns tty lines. The system configuration file lists the following terminal line controllers in the order shown below:

duart() ## integrated Duart terminal line controller
syac() ## first Systech terminal line controller
syac(1) ## second Systech terminal line controller

Further suppose that **syac()** is a VAC/16 controller and **syac(1)** is a VDA/128 host adapter. For this configuration, the DG/UX system assigns tty lines as follows:

tty00 to the RS-232/422 port on the duart.
tty01—tty16 to the VAC/16 controller.
tty17—tty272 to the VDA/128 host adapter.

If the **duart** was listed after the two **syac** devices, the tty line assignment would be as follows:

tty00—tty15 to the VAC/16 controller.
tty16—tty271 to the VDA/128 host adapter.
tty272 to the RS-232/422 port on the duart.

The DG/UX system assigns each of the 16 tty lines that it allocates to a VAC/16 host adapter to its 16 ports in sequential order. In other words, in the example above where **tty01** through **tty16** are allocated to the VAC/16 controller, **tty01** is assigned to port 0, **tty02** is assigned to port 1, **tty03** is assigned to port 2, and so on.

Since devices connect to a VDA host adapter through ports on cluster controllers, the DG/UX system allocates specific tty lines to those ports. Table C-3 shows how the DG/UX system allocates 16 tty lines to each cluster controller address, 01 through 10 hexadecimal (16 decimal). This means that DG/UX allocates 16 tty lines to an 8-line cluster controller with one of these addresses. Since an 8-line cluster controller has eight asynchronous ports (ports 0 through 7) and one parallel printer port (port 8), the last seven tty lines allocated to this controller's address are unused. The DG/UX system assigns the 16 tty lines that it allocates to a cluster controller address as follows: the tty line with the lowest number (call it *n*) to port 0; the next tty line with the next highest number (*n*+1) to port 1; the one with the next highest number (*n*+2) to port 2, and so on.

Table C-3 tty Lines Allocated to Cluster Controller Addresses

Cluster Controller Address	tty Lines Allocated
01	**tty**(*n*) through **tty**(n+15)
02	**tty**(*n* + 16) through **tty**(*n* + 31)
03	**tty**(*n* + 32) through **tty**(*n* + 47)
04	**tty**(*n* + 48) through **tty**(*n* + 63)
05	**tty**(*n* + 64) through **tty**(*n* + 79)
06	**tty**(*n* + 80) through **tty**(*n* + 95)
07	**tty**(*n* + 96) through **tty**(*n* + 111)
08	**tty**(*n* + 112) through **tty**(*n* + 127)
09	**tty**(*n* + 128) through **tty**(*n* + 143)
0A	**tty**(*n* + 144) through **tty**(*n* + 159)
0B	**tty**(*n* + 160) through **tty**(*n* + 175)
0C	**tty**(*n* + 176) through **tty**(*n* + 191)
0D	**tty**(*n* + 192) through **tty**(*n* + 207)
0E	**tty**(*n* + 108) through **tty**(*n* + 223)
0F	**tty**(*n* + 224) through **tty**(*n* + 239)
10	**tty**(*n* + 240) through **tty**(*n* + 255)

NOTES: The cluster controller address is also called the node address. For more information on these addresses, refer to the *HPS Downloadable Cluster Controller Installation Guide*.

The last tty line, **tty**(*n*+255), allocated to the cluster controller with address 10 is not used. If this cluster controller is a 16-line controller, this tty line is assigned port 15, so cannot be used. If it is an 8-line cluster box, this tty line is one of the seven unused tty lines allocated to the controller.

Continuing with this example, let's assume that the VDA/128 host adapter with one 8-line cluster controller and one 16-line cluster controller is allocated tty lines **tty16** through **tty270**. If the 8-line cluster controller has address 01 and the 16-line controller has address 02, then the tty lines for the ports on the controllers are as follows.

Examples:

8-line cluster controller (address 01):

tty16 through **tty24** for ports 0 through 8 (port 8 is the parallel printer port).
tty25 through **tty31** are unused.

16-line cluster controller (address 02):

tty32 through **tty47** for ports 0 through 15.
tty48 through **tty271** are unused.

You should now have enough information on your Terminal Line Controller Worksheet and your tty worksheets to determine the specific tty line that your DG/UX system assigned to each port on a terminal line controller. Using these worksheets, proceed as follows:

1. Using the Terminal Line Controller Worksheet that you completed in Appendix E, find the device name with Configuration File Position 1, then get the tty worksheet that has that device name written on it.

2. Determine the tty line or range of tty lines assigned to this device name using the procedure below for your type of computer: computer with a nongraphics system console, or AViiON 400 series workstation.

   For a computer with a nongraphics system console, if the device name is **duart()** or **duart(1)**, write "00" in the tty Line column on the tty worksheet for that device name. If the device name is **syac()**, **syac(1)**, **syac(2)**, **syac(3)**, or **syac(4)**, use the formula in the Table C-3 for the board type of that device name to calculate the range of tty lines assigned to that board (where n = 00), and record this range on the tty worksheet for the board.

   For an AViiON 400 series workstation, if the device name is **duart()** or **duart(1)**, write "01" in the tty Line column on the tty worksheet for that device name. If the device name is **syac()** or **syac(1)**, use the formula given in the previous table for the board type of that device name to calculate the range of tty lines assigned to that board (where n = 01), and record this range on the tty worksheet for the board.

3. Using the Terminal Line Controller Worksheet that you completed in Appendix E, find the device name with previous higher Configuration File Position, and get the tty worksheet that has that device name written on it.

4. Determine the tty line or range of tty lines assigned to this device as described below. Use the following value for n:

   $$n = 1 + [\ \textit{highest tty line number you calculated for next device name}]$$

   If the device name is **duart()** or **duart(1)**, write the value for n in the tty Line column on the tty worksheet for that device name. If the device name is **syac()**, **syac(1)**, **syac(2)**, **syac(3)**, or **syac(4)**, use the formula in the next table for the board type of that device name to calculate the range of tty lines assigned to that board, and record this range on the tty worksheet for the board.

Table C-4 Lines Allocated to Systech Terminal Line Controllers and Cluster Controllers

Board Type	Range of tty Lines Allocated
VAC/16	**tty**(*n*) through **tty**(*n*+15)
VDA/128 or VDA/255	**tty**(*n*) through **tty**(*n* + 255)
Cluster controller with address:	
01	**tty**(*n*) through **tty**(*n* + *15*)
02	**tty**(*n* + 16) through **tty**(*n* + 31)
03	**tty**(*n* + 32) through **tty**(*n* + 47)
04	**tty**(*n* + 48) through **tty**(*n* + 63)
05	**tty**(*n* + 64) through **tty**(*n* + 79)
06	**tty**(*n* + 80) through **tty**(*n* + 95)
07	**tty**(*n* + 96) through **tty**(*n* + 111)
08	**tty**(*n* + 112) through **tty**(*n* + 127)
09	**tty**(*n* + 128) through **tty**(*n* + 143)
0A	**tty**(*n* + 144) through **tty**(*n* + 159)
0B	**tty**(*n* + 160) through **tty**(*n* + 175)
0C	**tty**(*n* + 176) through **tty**(*n* + 191)
0D	**tty**(*n* + 192) through **tty**(*n* + 207)
0E	**tty**(*n* + 108) through **tty**(*n* + 223)
0F	**tty**(*n* + 224) through **tty**(*n* + 239)
10	**tty**(*n* + 240) through **tty**(*n* + 255)

NOTE: *n* = the lowest tty line number assigned to the board type.

5. Repeat Steps 3 and 4 to determine the tty line or range of tty lines assigned to any other terminal line controllers in your computer.

   If your computer contains a VAC/16 controller, continue to Step 6; otherwise, go to Step 7.

6. On the tty worksheet for each VAC/16 controller, write the appropriate tty line in the tty Line column for each port number. The lowest numbered tty line allocated to the controller is the one for port 0, the next higher numbered tty line allocated is the one for port 1, and so on.

7. If your computer contains a VDA host adapter, continue to Step 7; if not, then you have finished determining the specific tty line that DG/UX assigns to each terminal line controller port in your system.

8. On the tty worksheet for each VDA host adapter, write the appropriate tty line number in the tty Line column for each port number for each cluster address. The lowest numbered tty line allocated to the controller is the one for port 0 on the cluster controller with address 01, and the next higher numbered tty line allocated is the one for port 1 on the same cluster controller, and so on. Since DG/UX assigns 16 tty lines to each cluster controller regardless of the number of ports it has, the seven highest tty lines assigned to an 8-line cluster controller are not used. To determine the tty line for port 0 on successive cluster controllers, use the formulas in the preceding table.

You have finished determining the specific tty line that DG/UX assigns to each terminal line controller port in your system.

End of Appendix

Appendix D
Device Naming

Each DG/UX system device is known by a unique name which is constructed using the DG/UX common device specification format (described in a later section). Categories of devices follow:

- SCSI disk and tape devices, and disk arrays
- SCSI and disk array device controllers and adapters
- Local area network (LAN) controllers
- ESDI and SMD disk controllers
- Synchronous and asynchronous line controllers
- Other devices (workstation keyboard, graphics display, parallel line printer)

You need to know device names under these circumstances:

- To specify boot tape and disk devices when booting stand-alone **diskman**.
- To recognize and specify disk and tape devices when performing operations through **diskman**.
- To recognize devices that are configured into the kernel and listed in the system configuration file.
- To specify nonstandard devices to be probed and configured. Refer to the section titled "Standard and Nonstandard Devices" for a definition of the terms standard and nonstandard.
- To recognize device nodes that are created in the **/dev** directory.

Two forms of the DG/UX common device specification format are available for specifying devices: short and long.

Usually, you will use the short form to specify devices. However, each short form is internally represented by its long form. Nonstandard devices and device nodes are exclusively represented in the long form. The syntax of each form is presented in a later section.

Standard and Nonstandard Devices

A standard device is one you purchased from Data General for use with AViiON computers with its device ID preset through a jumper or DIP switch setting at the factory. A series of tables in this appendix list the default values for all standard devices. All preset devices are jumpered according to the information in these tables, which identify a device by pairing a device name and an adapter or controller number, with additional parameters (such as device code or unit number and adapter or controller address). The standard device settings allow you to refer to devices using the short form, which is just the device name and device ID (or unit number). A list of standard devices is given in **/usr/etc/probedevtab**.

If you must add a device supplied by Data General, but you have used all of the standard short name device settings (for example, only one integrated SCSI adapter (**insc**) is supported, and you want to add a second integrated SCSI adapter), you must determine the address for the device and specify the long form. Refer to *Programming in the DG/UX™ Kernel Environment* for information on determining the controller or adapter address.

NOTE: Be sure to keep track of the device IDs that you may have rejumpered in case you have any problems that may require help from Data General.

A device is considered to be nonstandard if it uses a driver that is not supplied by Data General. You are responsible for resetting the device's jumper or DIP switch position to establish the desired device setting. Refer to the device's hardware installation documentation for information on resetting a device's jumper or DIP switch positions. You would use the long form to specify the nonstandard device.

The DG/UX Common Device Specification Format

The DG/UX common device specification format contains a device name followed by a list of optional parameters.

device [*@device-code*]([*parameter1* [,*parameter2*[,*parameter3*]]])

device is a short mnemonic that identifies the device type such as **sd** for SCSI disk and **st** for SCSI tape devices.

device-code has two meanings. For integrated devices, it is an internal representation; for VME devices, it represents the interrupt vector.

parameters1-3, enclosed in parentheses, are specific to the device. They are covered in detail later in this appendix.

The values for each of these fields are determined by the requirements of the particular device. For example, the meanings of the *parameter* fields will be different from device to device. For example, *parameter2* for a SCSI disk is the SCSI ID; for a Systech asynchronous controller, it is the line number.

Keep in mind that numbers are represented in hexadecimal format. Valid values range from 0-9 and A-F (a-f). Table D-1 shows the different legal values for the parameters in the device specification format.

Table D-1 Parameter Meanings for the DG/UX Device Specification Format

Device	Description	Parameter1	Parameter2	Parameter3
Disk Arrays and SCSI Disk and Tape Device Names				
da	Disk array	Device specification of controlling SCSI adapter or array adapter	Unit-number	-
sd	SCSI disk	Device specification of controlling SCSI adapter or array adapter	SCSI ID	Logical-unit-number
st	SCSI tape	Device specification of controlling SCSI adapter or array adapter	SCSI ID	Logical-unit-number
SCSI and Disk Array Device Adapter Names				
cisc	Ciprico SCSI adapter	Controller's VME A16 address	-	-
hada	High-Availability Disk Array Subsystem	Controller's VME A16 address	-	-
insc	Integrated SCSI adapter	Address of adapter's memory-mapped control registers	-	-
ncsc	NCR SCSI adapter	Address of adapter's memory-mapped control registers	-	-

(continued)

NOTE: A dash (-) means that the field is not applicable, and its value can be represented as zero (0) or omitted.

Table D-1 Parameter Meanings for the DG/UX Device Specification Format

Device	Description	Parameter1	Parameter2	Parameter3
		Network Controller Names		
dgen	Data General second-generation integrated Ethernet controller	-	-	Alternate Ethernet address
hken	Interphase VME Ethernet controller	Controller's VME A16 address	Controller's VME A32 address	Alternate Ethernet address
inen	Integrated Ethernet controller	-	-	Alternate Ethernet address
vitr	VME token ring controller	Controller's A32 address	Alternate Token Ring Network address	IBM product ID (18 hex digits)
		SMD and ESDI Disk Device Names		
cied	Ciprico ESDI disk controller	Controller's VME A16 address	Unit-number	-
cimd	Ciprico SMD disk controller	Controller's VME A16 address	Unit-number	-
cird	Ciprico ESDI or SMD disk controller	Controller's VME A16 address	Unit-number	-
		Synchronous and Asynchronous Line Controller Names		
ssid	VME VSC synchronous controller	Controller's VME A32 address	Line-number or unit-number	-
syac	Systech asynchronous terminal controller	Controller's VME A32 address	Line-number	-
duart	Dual-line asynchronous terminal controller	Address of **duart**'s memory-mapped control registers	Line-number	-

(continued)

NOTE: A dash (-) means that the field is not applicable, and its value can be represented as zero (0) or omitted.

Table D-1 Parameter Meanings for the DG/UX Device Specification Format

Device	Description	Parameter1	Parameter2	Parameter3
	Other Device Names			
grfx	Workstation graphics - display	-	-	
kbd	Workstation keyboard	-	-	-
lp	Parallel line-printer port	-	-	-

(concluded)

NOTE: A dash (-) means that the field is not applicable, and its value can be represented as zero (0) or omitted.

When specifying any device, 0 refers to the first instance of a particular device type. If a particular field contains a 0, you may omit the 0. Also, you may omit commas if they are superfluous; you must, however, keep any parentheses. Spaces are not allowed between any of these fields or within a field of this format.

Disk Arrays and SCSI Disk and Tape Device Names

The format for the disk array and SCSI disk and tape devices follows:

device(*parameter1*,*parameter2*,*parameter3*)

where:

device is as a two-letter mnemonic that specifies the disk array or the standard integrated SCSI device. The valid values are:

da Disk array
sd SCSI disk
st SCSI tape

parameter1 is the controlling adapter's device specification. See the next section "SCSI and Disk Array Device Adapters," for these values.

parameter2 for the **da** device is the particular unit attached to a High-Availability Disk Array Subsystem. A single adapter supports up to 30 standard units in arrays, numbered 6-23 (hexadecimal).

parameter2 for the **sd** and **st** devices is the SCSI ID number on the disk or tape device which is jumpered at the factory. Table D-2 lists the default SCSI IDs.

Table D-2 Default SCSI ID Numbers

Drive Type	SCSI ID Number (Parameter2)
First disk	0
Second disk	1
Third disk	2
Diskette drives	3
First cartridge tape	4
Second cartridge tape	5
Third cartridge tape	6

NOTE: These default SCSI ID number assignments are conventions only; any SCSI device can be rejumpered as SCSI ID 0 through 6.

Disk devices include the following: Winchester hard disk, 3.5-inch floppy diskette, 5.25-inch floppy diskette, CD-ROM, WORM, and erasable magneto-optical disk.

parameter3 for the **sd** and **st** devices is the logical unit number of a particular device that shares the same SCSI ID. For example, any SCSI ID of (such as 3) supports up to 8 unit numbers. A single SCSI ID can manage multiple devices through logical unit numbers, 0-7.

Refer to the next section for examples of disk array and SCSI disk and tape device specifications.

SCSI and Disk Array Device Adapter Names

The format for SCSI and disk array device adapters follows:

device[@*device-code*](*parameter1*)

where:

device is a mnemonic that specifies the standard SCSI and disk array device adapters. The valid values are:

cisc Ciprico SCSI adapter
hada High-Availability Disk Array Subsystem
insc Integrated SCSI adapter
ncsc Integrated NCR SCSI interface adapter

device-code has two meanings. For integrated devices, it is an internal representation; for VME devices, it is the interrupt vector.

parameter1 is the adapter number (when using the short form) or the adapter's address (when using the long form). Table D-3 lists the device codes, valid adapter numbers supported (short form), and adapter addresses (long form).

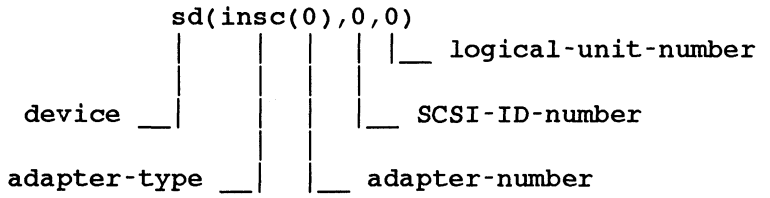
Table D-3 SCSI and Disk Array Specifications

Adapter Number (Short Form) (Parameter1)	Adapter Address (Long Form) (Device Code & Parameter1)
Ciprico SCSI Adapter (cisc)	
cisc(0)	cisc@28(FFFFF300)
cisc(1)	cisc@29(FFFFF500)
cisc(2)	cisc@2A(FFFFF700)
cisc(3)	cisc@2B(FFFFF900)
cisc(4)	cisc@2C(FFFFED00)
cisc(5)	cisc@2D(FFFFD700)
cisc(6)	cisc@2E(FFFFD900)
cisc(7)	cisc@2F(FFFFDB00)
cisc(8)	cisc@20(FFFFDD00)
cisc(9)	cisc@21(FFFFDF00)
cisc(A)	cisc@22(FFFFE100)
cisc(B)	cisc@23(FFFFE300)
cisc(C)	cisc@24(FFFFE500)
cisc(D)	cisc@25(FFFFE700)
cisc(E)	cisc@26(FFFFE900)
cisc(F)	cisc@27(FFFFEB00)
High-Availability Disk Array Subsystem (hada)	
hada(0)	hada@70(FFFF1000)
hada(1)	hada@71(FFFF1400)
hada(2)	hada@72(FFFF1800)
hada(3)	hada@73(FFFF1C00)
Integrated SCSI Adapter (insc)	
insc(0)	insc@7(FFF8A000)
NCR SCSI Interface Adapter (ncsc)	
ncsc(0)	ncsc@7(FFFB0000)
ncsc(1)	ncsc@13(FFFB0080)

Adapters with values outside this range are nonstandard.

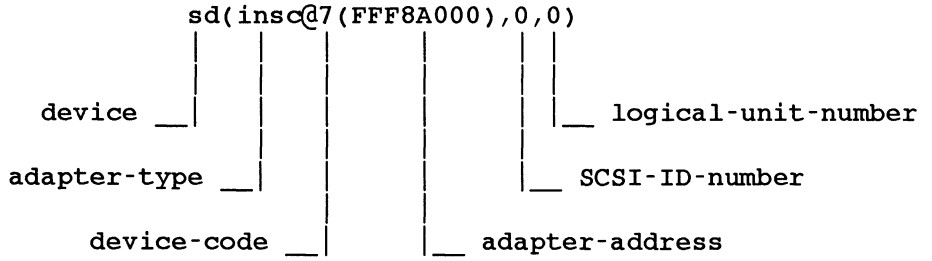
Examples 1 and 2 are short and long formats for the same device.

Example 1: SCSI Disk Device (Short Form)



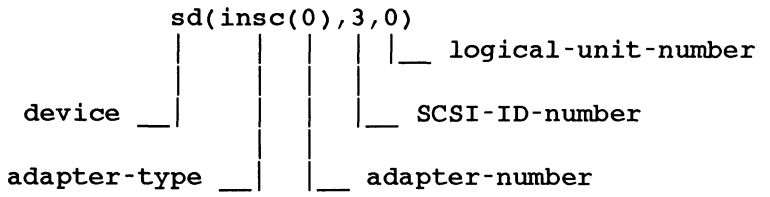
This device specification identifies a disk having the SCSI ID 0 on the first (0) integrated SCSI adapter. Since the disk is at LUN 0, it needn't be further qualified by a logical unit number. This device is specified as **sd(insc(),0)** or **sd(insc())**. Both the first adapter number, the first SCSI ID number, and the logical unit number are 0.

Example 2: SCSI Disk Device (Long Form)



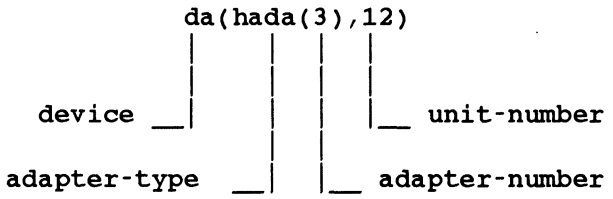
This device specification identifies a disk having the SCSI ID of 0 on the integrated SCSI adapter whose device code is 7 and adapter address is FFF8A000. Since the disk is at LUN 0, it needn't be further qualified by a logical unit number. This device could also be specified as **sd(insc())**.

Example 3: SCSI Diskette Device (Short Form)



This device specification identifies the first (0) diskette drive attached to a SCSI disk controller (having the SCSI ID of 3) on the integrated SCSI adapter.

Example 4: High-Availability Disk Array Subsystem (Short Form)



This device specification identifies a disk array having the unit number 12 on the fourth (3) High-Availability Disk Array Subsystem.

More examples:

sd(insc(),*) All SCSI disks on the integrated SCSI adapter. The asterisk is a DG/UX device-naming metacharacter, matches any character. It can be used only in the system configuration file, not in a device specification for a boot disk or tape device.

st(cisc(1),4) Tape drive having the SCSI ID 4 on the second (1) Ciprico SCSI adapter.

st(hada(3),4) SCSI tape device having the SCSI ID 4 on the fourth (3) High-Availability Disk Array Subsystem adapter.

ncsc(0) Integrated NCR SCSI adapter.

Network Controller Device Names

The format for the network controller devices follows:

device [@*device-code*]([*parameter1* [,*parameter2*[,*parameter3*]]])

where:

device is a mnemonic that specifies the network controller device. The valid values are:

dgen Data General second generation integrated Ethernet controller
hken Interphase VME Ethernet controller
inen Integrated Ethernet controller
vitr VME token ring controller

device-code has two meanings. For integrated devices, it is an internal representation; for VME devices, it is the interrupt vector.

parameter1 is the controller number.

dgen Not used
hken Controller's VME A16 address
inen Not used
vitr Controller's VME A32 address

Table D-4 lists the valid adapter numbers supported (short form), the device code, and controller address (long form).

parameter2 is different for each device; they follow:

dgen Not used
hken Controller's VME A32 address
inen Not used
vitr Alternate token ring address that overrides board-set default

Table D-3 lists the valid controller numbers supported (short form), device code, and controller address (long form).

parameter3 is different for each device; they follow:

dgen Alternate Ethernet address that overrides board-set default
hken Alternate Ethernet address that overrides board-set default
inen Alternate Ethernet address that overrides board-set default
vitr IBM product ID (18 hex digits) that overrides board-set default

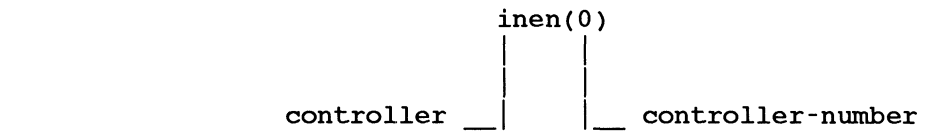
Table D-4 Network Controller Specifications

Controller Number (Short Form) (Parameter1)	Controller Address (Long Form) (Device Code, Parameter1, & Parameter2)
Data General Second Generation Integrated Ethernet Controller (dgen)	
dgen(0)	dgen@6()
dgen(1)	dgen@12()
Interphase VME Ethernet Controller (hken)	
hken(0)	hken@15(FFFF4000,55900000)
hken(1)	hken@16(FFFF5000,55980000)
hken(2)	hken@10(FFFF4200,E1100000)
hken(3)	hken@11(FFFF4400,E1180000)
hken(4)	hken@12(FFFF4600,E1200000)
hken(5)	hken@13(FFFF4800,E1280000)
hken(6)	hken@14(FFFF4A00,E1300000)
hken(7)	hken@17(FFFF4C00,E1380000)
Integrated Ethernet Controller (inen)	
inen(0)	inen@6()
VME Token Ring Controller (vitr)	
vitr(0)	vitr@40(61000000)
vitr(1)	vitr@41(61002000)
vitr(2)	vitr@42(E4004000)
vitr(3)	vitr@43(E4006000)
vitr(4)	vitr@44(E4008000)
vitr(5)	vitr@45(E400A000)
vitr(6)	vitr@46(E400C000)
vitr(7)	vitr@47(E400E000)

Controllers whose values fall outside this range are considered to be nonstandard.

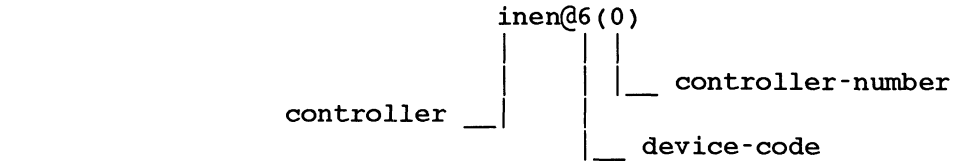
Examples 1 and 2 are short and long forms of the same device.

Example 1: Ethernet LAN Device (Short Format)



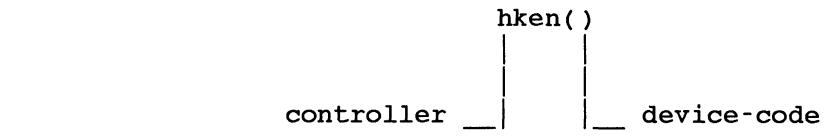
This device specification identifies an integrated Ethernet LAN.

Example 2: Ethernet LAN Device (Long Format)



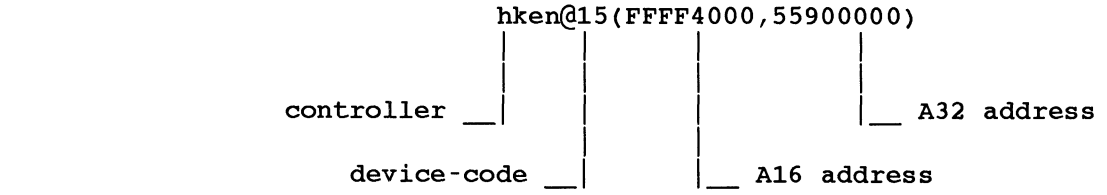
This device specification identifies an integrated Ethernet LAN whose device code is 6.

Example 3: Interphase VME Ethernet Controller (Short Form)



This device specification identifies the first interphase VME Ethernet Controller.

Example 4: Interphase VME Ethernet Controller (Long Form)



This device specification identifies an interphase VME Ethernet Controller whose device code is 15, A16-address is FFFF4000, and A32-address is 55900000.

Other examples:

hken(5) Sixth (5) Interphase VME Ethernet LAN.
vitr(7) Eighth (7) VME token ring controller.

SMD and ESDI Disk Device Names

The format for the SMD and ESDI disk devices follows:

device [@*device-code*](*parameter1*, *parameter2*)

where:

device is a mnemonic that specifies the controller device to which SMD and ESDI disk devices are attached. The valid values are:

cied Ciprico ESDI controller
cimd Ciprico SMD controller
cird Ciprico ESDI controller or Ciprico SMD controller

NOTE: The **cird** device name for the Ciprico controller is not recognized at the SCM prompt.

device-code has two meanings. For integrated devices, it is an internal representation; for VME devices, it represents the interrupt vector.

parameter1 is the device controller to which a disk device is attached (when using the short form) or the controller's VME A16 address (when using the long form). You can attach ESDI devices to a **cied** controller, or SMD devices to **cimd** devices. The name **cird** is a synonym for either the **cied** or **cimd** controller.

Table D-5 lists the valid adapter numbers supported (short form) and the corresponding device code and controller address (long form).

parameter2 is the unit number of the particular disk device attached to the controller.

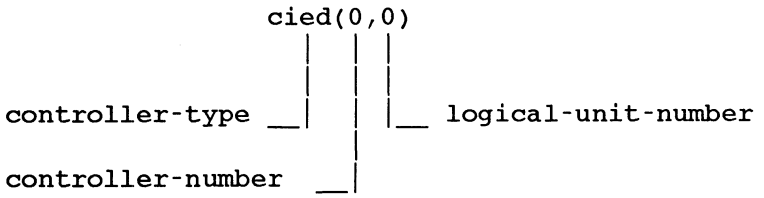
Table D-5 SMD and ESDI Disk Device Controller Device Specifications

Controller Number (Short Form) (Parameter1)	Controller Address (Long Form) (Device Code & Parameter1)
Ciprico ESDI Controller (cied)	
cied(0)	cied@18(FFFFEF00)
cied(1)	cied@19(FFFFF100)
cied(2)	cied@1A(FFFFFB00)
cied(3)	cied@1B(FFFFFD00)
Ciprico SMD Controller (cimd)	
cimd(0)	cimd@18(FFFFEF00)
cimd(1)	cimd@19(FFFFF100)
cimd(2)	cimd@1A(FFFFFB00)
cimd(3)	cimd@1B(FFFFFD00)

Controllers with values outside this range are nonstandard.

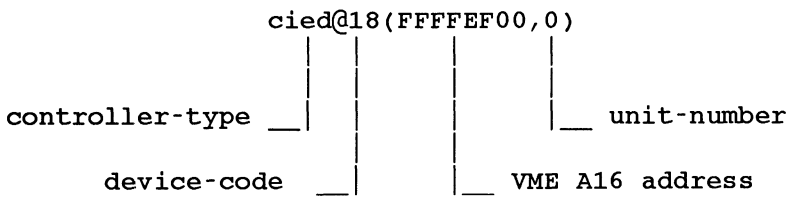
Examples 1 and 2 are short and long forms of the same device.

Example 1: ESDI Disk Device (Short Form)



This device specification identifies the first (0) disk attached to the first (0) Ciprico ESDI controller.

Example 2: ESDI Disk Device (Long Form)



This device specification identifies the first (0) disk attached to the ESDI controller whose device code is 18 and whose VME A16 address is FFFFEF00.

Other examples:

cied(0,2) Third (2) disk attached to the first (0) Ciprico Rimfire controller.
cimd() First disk attached to the first Ciprico Rimfire controller.
cird(3) First SMD or ESDI disk attached to to the fourth (3) Ciprico Rimfire controller.

Synchronous and Asynchronous Line Controller Names

The format for the synchronous and asynchronous line controllers follows:

device [@*device-code*](*parameter1*,*parameter2*)

where:

device is a mnemonic that specifies the synchronous and asynchronous line controllers. The valid values are:

syac Systech asynchronous terminal controller
ssid VME VSC synchronous controller
duart Dual-line asynchronous terminal controller

device-code has two meanings. For integrated devices, it is an internal representation; for VME devices, it represents the interrupt vector.

parameter1 is the logical number of the controller address (when using the short form) or the controller's VME A32 address (when using the long form). For the **duart** device, it refers to the address of its memory-mapped control registers (when using the long form). See Table B-6.

parameter2, for **syac**, refers to the line number, starting at 1.

parameter2, for **ssid**, is either the line number on the controller or the logical unit number. Values 0 through 3 refer to line numbers; all other values refer to logical unit numbers.

parameter2 for **duart** refers to the line number (either 0 or 1).

Table D-6 lists the valid controller numbers supported (short form) and the corresponding device code and controller address (long form).

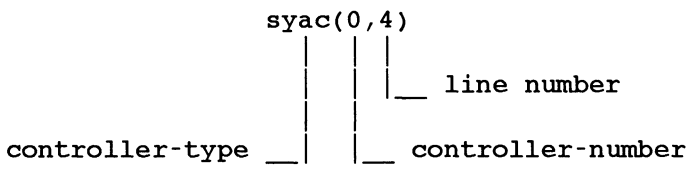
Table D-6 Synchronous and Asynchronous Line Controller Device Specifications

Controller Number (Short Form) (Parameter1)	Controller Address (Long Form) (Device Code & Parameter1)
VME VSC Synchronous Controller (ssid)	
ssid(0)	ssid@50(55B00000)
ssid(1)	ssid@51(55B10000)
ssid(2)	ssid@52(55B20000)
ssid(3)	ssid@53(55B30000)
ssid(4)	ssid@54(55B40000)
ssid(5)	ssid@55(55B50000)
ssid(6)	ssid@56(55B60000)
ssid(7)	ssid@57(55B70000)
ssid(8)	ssid@58(E2080000)
ssid(9)	ssid@59(E2090000)
ssid(A)	ssid@5A(E20A0000)
ssid(B)	ssid@5B(E20B0000)
ssid(C)	ssid@5C(E20C0000)
ssid(D)	ssid@5D(E20D0000)
ssid(E)	ssid@5E(E20E0000)
ssid(F)	ssid@5F(E20F0000)
Systech Asynchronous Terminal Controller (syac)	
syac(0)	syac@60(60000000)
syac(1)	syac@61(60020000)
syac(2)	syac@62(60040000)
syac(3)	syac@63(60060000)
syac(4)	syac@64(60080000)
syac(5)	syac@65(E30A0000)
syac(6)	syac@66(E30C0000)
syac(7)	syac@67(E30E0000)
syac(8)	syac@68(E3100000)
syac(9)	syac@69(E3120000)
syac(A)	syac@6A(E3140000)
syac(B)	syac@6B(E3160000)
syac(C)	syac@6C(E3180000)
syac(D)	syac@6D(E31A0000)
syac(E)	syac@6E(E31C0000)
syac(F)	syac@6F(E31E0000)
Dual-Line Asynchronous Terminal Controller (duart)	
duart(0)	duart@4(FFF82000)
duart(1)	duart@10(FFF82000)

Controllers whose values fall outside this range are considered to be nonstandard.

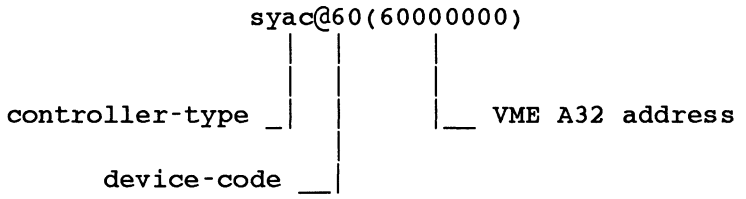
Examples 1 and 2 are short and long forms of the same device.

Example 1: Asynchronous Terminal Controller Device (Short Form)



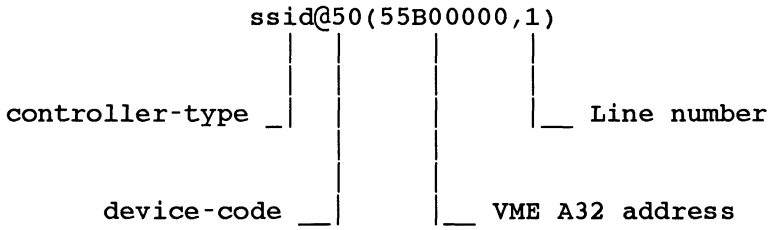
This device specification identifies the first (0) **syac** controller on the fourth (4) line.

Example 2: Asynchronous Terminal Controller Device (Long Form)



This device specification identifies the **syac** controller whose device code is 60 and whose VME A32 address is 60000000.

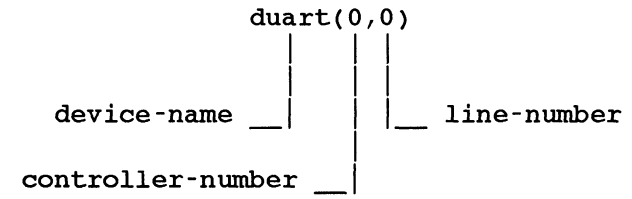
Example 3: VME VSC Synchronous Controller Device (Long Form)



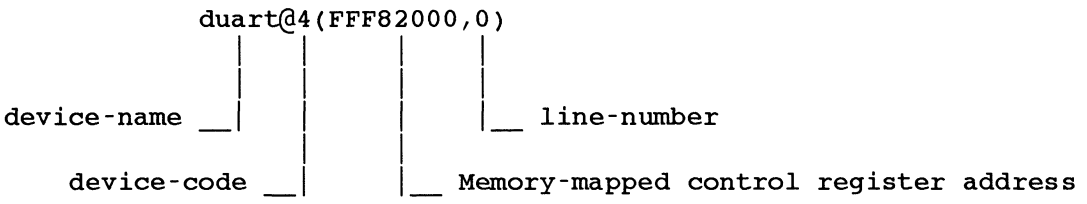
This device specification identifies the **ssid** controller whose device code is 50, VME-A32 address is 55B00000, and line number is 1.

Examples 4 and 5 are short and long forms of the same device.

Example 4: Duart (Short Form)



Example 5: Duart (Long Form)



This device specification identifies the first (0) device attached to the **duart** whose device code is 4 and memory-mapped control register address is FFF82000.

This device specification identifies the first (0) device attached to the first (0) **duart**. This device could also be specified as **duart()**.

Other Device Names

The format for the other devices follows:

device [@*device-code*]()

where:

device is a mnemonic that specifies the particular device. The valid values are:

kbd Workstation keyboard
grfx Graphics display
lp Parallel line printer

device-code has two meanings. For integrated devices, it is an internal representation; for VME devices, it represents the interrupt vector.

Table D-7 lists the valid controller numbers supported (short form) and the corresponding device code and controller address (long form).

Table D-7 Other Device Controller Device Specifications

Controller Number (Short Form) (Parameter1)	Controller Address (Long Form) (Parameter1)
Graphics Display, Keyboard, and Line Printer (grfx, kbd, lp)	
grfx(0)	grfx@9()
kbd(0)	kbd@3()
lp(0)	lp@5()

Controllers with values outside this range are nonstandard.

Device Nodes

The following sections describe device nodes for physical and logical disks, tapes, terminals, and all other devices supported by the DG/UX system. Device nodes are automatically created in the **/dev** directory each time you boot the kernel.

A device node is a special file that provides a handle to a particular device so that programs can access it. Device nodes come in two forms:

- Character mode (for character-at-a-time access)
- Block mode (for buffered access)

See the **mknod**(1M) manual page in the *System Manager's Reference for the DG/UX™ System.*

Physical Disk Nodes

Each physical disk is accessible in both block and character mode. Block nodes are in **/dev/pdsk**. Character nodes are in **/dev/rpdsk**. The following are example entries created in **/dev**:

/dev/pdsk/cied@18(FFFFEE00,0)	Block node for **cied(0,0)**
/dev/rpdsk/cied@18(FFFFEE00,0)	Character node for **cied(0,0)**
/dev/pdsk/sd(insc@7(FFF8A000),2,0)	Block node for **sd(insc(0),2,0)**
/dev/rdsk/sd(insc@7(FFF8A000),2,0)	Character node for **sd(insc(0),2,0)**

Logical Disks

Logical disks are accessible in both block and character mode. Therefore, the kernel creates both types of nodes. Block nodes are in **/dev/dsk** and character nodes are in **/dev/rdsk**. When you use **diskman** or **sysadm** to create a logical disk, a corresponding logical disk node is created. If you create a logical disk named **comm**, the nodes in **/dev** will be:

/dev/dsk/comm
/dev/rdsk/comm

Tape Drive Nodes

Tape drives are accessible only in character mode. The character-access nodes are in **/dev/rmt**; an example is **/dev/rmt/st(cisc@28(FFFFF300,4,0)**. All possible combinations of density and rewind options are created for each unit as follows:

/dev/rmt/*tape@device-code*(*address, SCSI-ID, LUN*)[{**lmh**}][**n**]

You select one of **l**, **m**, or **h** to indicate whether you are using a low, medium, or high density storage medium. A node without a denoting letter is also created, which accesses the device's default density.

Use **n** to indicate that you do not want the device to rewind after it closes.

/dev/rmt/st(cisc@28(FFFFF300),0,0)

This is a SCSI tape with a Ciprico SCSI adapter with default device code and address. The tape is at the default SCSI ID of 4.

Terminal Nodes

Terminal line devices have nodes with names of the form **/dev/tty***number*, where *number* is a decimal number of at least two digits starting at 00. The assignment of names to your system's terminal lines depends on the order that the terminal line controller devices are configured by the kernel, which in turn depends on the order those controller devices are listed in the system configuration file. Terminal nodes are assigned in a first-come, first-served order, starting at **/dev/tty00** and working up. If you use the standard kernel autoconfiguration mechanism, all standard **duart** controllers on the system are listed in order, followed by all standard **syac** controllers on the system.

Only one asynchronous line is available on each **duart** device. On workstation systems, this is because one channel of the first **duart** is used to provide access to the mouse device, which has the special device node **/dev/mouse**. Other systems also reserve one **duart** channel for special use, in this case as the system console (**/dev/syscon**). If the system has a second **duart**, only one port is provided at the back of the system cabinet, so only one additional terminal device node is created.

A **syac** controller causes from 16 to 255 device nodes to be created, depending on how many lines are on the controller. If a cluster controller box is used, all 255 lines are created, even though only a few of them (perhaps 8 or 16) are actually hooked up.

End of Appendix

Appendix E
Planning Worksheets

This appendix contains a series of worksheets for you to use when customizing the DG/UX system. Collecting this information ahead of time speeds up the customizing process considerably. Worksheets are provided for these activities:

- Logical Disk Planning
- Printer (Local and Remote) Planning
- Client Network Planning
- OS Server Network Planning
- Tty Line Planning

Logical Disk Planning

You will need to provide this information:

- Logical disk name
- Mount point directory
- Physical disk name in DG/UX common device specification format (refer to *Installing the DG/UX™ System*)
- Physical disk capacity in Mbytes (see Table 2-1)
- Size in blocks of logical disk (or piece)

A sample worksheet is provided as a model, followed by two worksheets for your use.

Sample Logical Disk Planning Worksheet

Logical Disk Name	Mount Point Directory	Disk Name sd(insc(0),0,0) 1040 Mbytes		Disk Name sd(insc(0),3,0) 662 Mbytes		Disk Name sd(insc(0),5,0) 322 Mbytes	
		Piece #	Blocks	Piece #	Blocks	Piece #	Blocks
swap	-	1	50,000				
root	/	1	40,000				
usr	/usr	1	240,000				
usr_opt_X11	/usr/opt/X11	1	105,000				
usr_opt_aview	/usr/opt/aview	1	10,000				
home	/home	1	100,000	2	100,000		
usr_opt_pkg	/usr/opt/pkg	1	100,000	2	100,000	3	100,000
var_tmp	/var/tmp	1	50,000				
srv	/srv	1	50,000				
srv_root	/srv/release/PRIMARY	1	100,000	2	100,000		
srv_dump	/srv/dump	1	38,600				
srv_swap	/srv/swap			1 2	200,000 34,000		
root_dgux432	/srv/release/dgux_432			1	200,000		
usr_dgux432	/srv/release/dgux_432			1	240,000		
X11_dgux432	/srv/release/dgux_432			1	100,000	2	5,000
Total Used			883,600		1,074,000		6000
Total Capacity			2,129,920		1,355,776		659,456
Free Space			1,246,320		281,776		653,456

Logical Disk Layout (Part 1 of 2)

Logical Disk Name	Mount Point Directory	Disk Name _____ _____ Mbytes		Disk Name _____ _____ Mbytes		Disk Name _____ _____Mbytes	
		Piece #	Blocks	Piece#	Blocks	Piece #	Blocks
swap	-	1	50,000				
root	/	1	40,000				
usr	/usr	1	240,000				
usr_opt_X11	/usr/opt/X11	1	105,000				
usr_opt_aview	/usr/opt/aview	1	10,000				
Total Used							
Total Capacity							
Free Space							

(continued)

Logical Disk Layout (Part 2 of 2)

Logical Disk Name	Mount Point Directory	Disk Name _____ _____ Mbytes Piece # Blocks	Disk Name _____ _____ Mbytes Piece# Blocks	Disk Name _____ _____Mbytes Piece # Blocks
Total Used				
Total Capacity				
Free Space				

(concluded)

Printer Planning

Two worksheets are provided for recording information for local printers (attached directly to your AViiON computer) and remote printers (attached to another AViiON computer, but accessible via a LAN).

Local Printer Planning Worksheet (1 of 2)

Printer Name	Printer Model	Device File	Printer Type
Example: laser	*standard*	*/dev/tty2*	*unknown*

Remote Printer Planning Worksheet (2 of 2)

Printer Name (Local Name)	Remote Host	Print Name (Remote Name)
Example: draft	*gyramax*	*post*

Client Network Planning

Two worksheets are provided for recording information for clients (OS and X). Record the Internet address and Ethernet address for each client on your system in the first worksheet, and the appropriate ONC and TCP/IP parameter values in the second worksheet.

Client Network Planning Worksheet (Part 1 of 2)

Client Host Name	Internet Address	Ethernet Address
Example: mav-client	*123.227.3.14*	*08:00:1B:03:45:11*

Client Network Planning Worksheet (Part 2 of 2)

ONC and TCP/IP Parameters	Example Values	Actual Values
NIS domain name	**my_domain**	
Do you subnet?	**yes**	
Network mask	**0xffffff00**	

OS Server Network Planning

One worksheet is provided for recording the appropriate ONC and TCP/IP parameter values for OS servers.

OS Server Network Planning Worksheet

ONC and TCP/IP Parameters	Example Actual Information	Information
NIS domain name	**work-net**	
Host name	**alvin**	
Internet address	**123.227.2.14**	
Do you subnet?	**yes**	
Network mask	**0xffffff00**	
Controller device name	**inen0**	

Tty Line Planning

The following worksheets are used for planning your tty lines.

- Terminal Line Controllers Worksheet
- Device Worksheet for RS/232/422 Ports on the Computer Unit
- Device Worksheet for a VAC/16 Controller
- Device Worksheet for a VDA Host Adapter

Each Device Worksheet for a VDA Host Adapter contains space for recording 32 ports. So, if you have a VDA host adapter, you will need two additional sheets for a VDA/128 host adapter and six additional sheets for a VDA/255 host adapter.

Preceding each worksheet is a sample worksheet that has been filled out for an AViiON 5000 computer with three terminal line controllers: RS-232/422 terminal/modem port on the computer unit, one VAC/16 controller, and one VDA/128 host adapter. The host adapter has one 8-line cluster controller and seven 16-line cluster controllers, providing ports for a maximum of 120 serial devices and one parallel printer. These terminal line controllers have the device names given below, and these device names are listed in your DG/UX system configuration file in the relative order shown below:

duart() terminal/modem port on computer unit
syac() VAC/16
syac(1) VDA/128

SAMPLE Terminal Line Controllers Worksheet

Board No.	Device Name	Configuration File Position	Board or Port Type	Cluster Controllers			
				Address	No. Lines	Address	No. Lines
	duart()	*1*					
	duart(1)						
0	syac()	*2*	*VAC/16*	01		09	
				02		0A	
				03		0B	
				04		0C	
				05		0D	
				06		0E	
				07		0F	
				08		10	
1	syac(1)	*3*	*VDA/128*	01	*8*	09	
				02	*16*	0A	
				03	*16*	0B	
				04	*16*	0C	
				05	*16*	0D	
				06	*16*	0E	
				07	*16*	0F	
				08	*16*	10	
2	syac(2)			01		09	
				02		0A	
				03		0B	
				04		0C	
				05		0D	
				06		0E	
				07		0F	
				08		10	
3	syac(3)			01		09	
				02		0A	
				03		0B	
				04		0C	
				05		0D	
				06		0E	
				07		0F	
				08		10	
4	syac(4)			01		09	
				02		0A	
				03		0B	
				04		0C	
				05		0D	
				06		0E	
				07		0F	
				08		10	

Terminal Line Controllers Worksheet

Board No.	Device Name	Configuration File Position	Board or Port Type	Cluster Controllers			
				Address	No. Lines	Address	No. Lines
	duart()						
	duart(1)						
0	syac()			01		09	
				02		0A	
				03		0B	
				04		0C	
				05		0D	
				06		0E	
				07		0F	
				08		10	
1	syac(1)			01		09	
				02		0A	
				03		0B	
				04		0C	
				05		0D	
				06		0E	
				07		0F	
				08		10	
2	syac(2)			01		09	
				02		0A	
				03		0B	
				04		0C	
				05		0D	
				06		0E	
				07		0F	
				08		10	
3	syac(3)			01		09	
				02		0A	
				03		0B	
				04		0C	
				05		0D	
				06		0E	
				07		0F	
				08		10	
4	syac(4)			01		09	
				02		0A	
				03		0B	
				04		0C	
				05		0D	
				06		0E	
				07		0F	
				08		10	

SAMPLE Device Worksheet for RS-232/422 Ports On Computer Unit

Port type: *terminal/modem*		**Device name:** duart()
tty Line	**Device Type**	**Description**
00	*modem to VT100*	*lab B2, conn 2203*
Port type:		**Device name:** duart(1)
tty Line	**Device Type**	**Description**

Device Worksheet for RS-232/422 Ports On Computer Unit

Port type:		**Device name:** duart()
tty Line	**Device Type**	**Description**
Port type:		**Device name:** duart(1)
tty Line	**Device Type**	**Description**

SAMPLE Device Worksheet for a VAC/16 Controller

Board no: *0*			Device name: *syac()*				Range of tty lines: *01 - 16*
Port No.	tty Line	Device Type	Description	Port No.	tty Line	Device Type	Description
0	*01*	*4558 printer*	*officeA1 conn 1100*	8	*09*	*D216+*	*officeA10 conn 1118*
1	*02*	*Epson printer*	*officeA2 conn 1102*	9	*10*	*VT100*	*officeA11 conn 1120*
2	*03*	*D216+*	*officeA3 conn 1104*	10	*11*	*D462+*	*officeA12 conn 1122*
3	*04*	*D462+*	*officeA4 conn 1106*	11	*12*	*D462+*	*officeA14 conn 1124*
4	*05*	*D216+*	*officeA5 conn 1108*	12	*13*	*D462+*	*officeA14 conn 1124*
5	*06*	*D216+*	*officeA6 conn 1110*	13	*14*	*D462+*	*officeA18 conn 1128*
6	*07*	*D216+*	*officeA8 conn 1114*	14	*15*	*D462+*	*officeA20 conn 1130*
7	*08*	*D216+*	*officeA9 conn 1116*	15	*16*	*D462+*	*officeA21 conn 1132*

Device Worksheet for a VAC/16 Controller

Board no:			Device name:				Range of tty lines:
Port No.	tty Line	Device Type	Description	Port No.	tty Line	Device Type	Description
0				8			
1				9			
2				10			
3				11			
4				12			
5				13			
6				14			
7				15			

SAMPLE Device Worksheet for a VDA Host Adapter

Sheet *1* **of** *4*

Board type: *VDA/128* **Board no:** *1* **Device name:** *syac(1)* **Range of tty lines:** *17-271*

Cluster Address	Port No.	tty Line	Device Type	Description	Cluster Address	Port No.	tty Line	Device Type	Description
01	0	*17*	*D216+*	*office B1,conn 1200*	*02*	0	*33*	*VT100*	*office B9,conn 1216*
	1	*18*	*D216+*	*office B2,conn 1202*		1	*34*	*VT100*	*office B10,conn 1218*
	2	*19*	*D462+*	*office B3,conn 1204*		2	*35*	*D216+*	*office B11,conn 1220*
	3	*20*	*VT100*	*office B4,conn 1206*		3	*36*	*D462+*	*office B12,conn 1222*
	4	*21*	*VT100*	*office B5,conn 1208*		4	*37*	*D462+*	*office B13,conn 1224*
	5	*22*	*VT100*	*office B6,conn 1210*		5	*38*	*VT100*	*office B14,conn 1226*
	6	*23*	*D216+*	*office B7,conn 1212*		6	*39*	*D462+*	*office B15,conn 1228*
	7	*24*				7	*40*	*D462+*	*office B16,conn 1230*
↓	8	*25*	*6772ptr*	*lab B2,conn 2204*		8	*41*	*D462+*	*office B17,conn 1232*
	9					9	*42*	*D462+*	*office B18,conn 1234*
	10					10	*43*	*D462+*	*office B19,conn 1236*
	11					11	*44*	*D462+*	*office B20,conn 1238*
	12					12	*45*	*D462+*	*office B21,conn 1240*
	13					13	*46*	*D462+*	*office B22,conn 1242*
	14					14			
	15				↓	15			

Device Worksheet for a VDA Host Adapter

Sheet _____ of _____

Board type:			Board no:	Device name:	Range of tty lines:				
Cluster Address	Port No.	tty Line	Device Type	Description	Cluster Address	Port No.	tty Line	Device Type	Description
	0					0			
	1					1			
	2					2			
	3					3			
	4					4			
	5					5			
	6					6			
	7					7			
	8					8			
.....	9				.....	9			
	10					10			
	11					11			
	12					12			
	13					13			
	14					14			
	15					15			

tty Lines Worksheet

Sheet _____ of _____

tty Line	Device Type	Lineset or Model	TERM Variable or Printer Name	tty Line	Device Type	Lineset or Model	TERM Variable or Printer Name

End of Appendix

Index

Note: Boldfaced page numbers (e.g., **1-5**) indicate definitions of terms or other key information.

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

V

W

X

Y

Related Documents

The following list of related manuals gives titles of Data General manuals followed by nine-digit numbers used for ordering. You can order any of these manuals via mail or telephone (see the TIPS Order Form in the back of this manual).

For a complete list of AViiON® and DG/UX™ manuals, see the *Guide to AViiON® and DG/UX™ Documentation* (069-701085). The on-line version of this manual found in **/usr/release/doc_guide** contains the most current list.

Data General Software Manuals

User's Manuals

Learning the UNIX® Operating System
Helps beginners learn UNIX fundamentals using step-by-step tutorials. Published by O'Reilly and Associates. Available from Data General via Ordering Number — 069-701042

User's Reference for the DG/UX™ System
Contains an alphabetical listing of manual pages for commands relating to general system operation. Ordering Number — 093-701054

Using the DG/UX™ Editors
Describes the text editors **vi** and **ed**, the batch editor **sed**, and the command line editor **editread**. Ordering Number — 069-701036

Using the DG/UX™ System
Describes the DG/UX system and its major features, including the C and Bourne shells, typical user commands, the file system, and communications facilities such as **mailx**. Ordering Number — 069-701035

The Korn Shell Command and Programming Language
Provides a tutorial to the Korn shell command language and interface, explain show to create scripts using this language, and provides examples and references pages. Ordering number — 093-701105

X Window System™ User's Guide, OSF/Motif Edition
Describes window concepts, the application programs (clients) commonly distributed with X, and how you can expect programs to operate with the OSF/Motif interface. Ordering Number — 069-100229

Installation and Administration Manuals

Installing the DG/UX™ System
Describes how to install the DG/UX operating system on AViiON hosts. Ordering Number — 093-701052

Managing the DG/UX™ System
Discusses the concepts and tasks related to DG/UX system management, providing general orientation to the administrator's job as well as instructions for managing disk resources, user profiles, files systems, printers and tape drives, and other features of the system. The manual approaches system administration through the **sysadm** facility. Ordering Number — 093-701088

Managing ONC™/NFS® and Its Facilities on the DG/UX™ System
Explains how to manage and use the DG/UX ONC™/NFS® product. Contains information on the Network File System (NFS), the Network Information Service (NIS), Remote Procedure Calls (RPC), and External Data Representation (XDR). Ordering Number — 093-701049

Managing TCP/IP on the DG/UX™ System
Explains how to prepare for the installation of Data General's TCP/IP (DG/UX) package on AViiON computer systems. Tells how to tailor the software for your site, use **sysadm** and **xsysadm** to manage the package and troubleshoot system problems. Ordering Number — 093-701051

System Manager's Reference for the DG/UX™ System
Contains an alphabetical listing of manual pages for commands relating to system administration or operation. Ordering Number — 093-701050

Programming Manuals

Programmer's Reference for the DG/UX™ System, (Volume 1)
Alphabetical listing of manual pages for DG/UX programming commands and system calls. This is part of a three-volume set. Ordering Number — 093-701055

Programmer's Reference for the DG/UX™ System, (Volume 2)
Alphabetical listing of manual pages for DG/UX subroutines and libraries. This is part of a three-volume set. Ordering Number — 093-701056

Programmer's Reference for the DG/UX™ System, (Volume 3)
Alphabetical listing of manual pages for DG/UX file formats, miscellaneous features, and networking protocols. Part of a three-volume set, this volume contains the table of contents and index (**contents** (0) and **index** (0)) for man pages. Ordering Number — 093-701102

Programming in the DG/UX™ Kernel Environment
Introduces kernel-level programming on the DG/UX™ system and provides reference pages for kernel-supplied utility routines. This manual is a pre-requisite to both *UNIX System V Release 4: Programmer's Guide: STREAMS* and *Writing a Standard Device Driver for the DG/UX™ System.* Ordering Number — 093-701083

Programmer's Guide: STREAMS (UNIX System V Release 4)
Describes the STREAMS interface facility and how to use it. The STREAMS facility provides special queuing, messaging and buffering functions that simplifies addition and deletion of modules of code. Information on how STREAMS works in the DG/UX system and descriptions of important kernel-level utility routines are found in *Programming in the DG/UX™ Kernel Environment* (093-701083). Ordering Number — 093-701106

Programmer's Guide: Systems Services and Application Packaging Tools (UNIX System V Release 4)
Describes standard programming procedures and interfaces available to the C application developer in the UNIX environment. Topics include interprocess communications, memory management, file and record locking and application packaging. **Note:** Chapters 5 and 9 of this Prentice Hall manual discuss topics that do not apply to the DG/UX system. Ordering Number — 093-701103

Writing a Standard Device Driver for the DG/UX™ System
Describes how to write a device driver for a DG/UX system running on an AViiON computer. Describes the drivers written to address specific devices or adapters that manage secondary bus access to specific devices. Information on kernel-level programming in the DG/UX system and descriptions of important kernel-level utility routines are found in *Programming in the DG/UX™ Kernel Environment* (093-701083). Ordering Number — 093-701053

AViiON System Manuals

Installation/Setup Manuals

Expanding AViiON® 5000 Series Systems
Explains how to open and close the 5000 series computer, plan a configuration, install add-on boards and mass-storage drives. Includes a description of the computer sub-assemblies. Ordering Number — 014-001850

Setting Up, Starting, and Maintaining AViiON® Workstations: 100, 200, and 300 Series
Describes how to unpack, connect system components and optional devices, power up and prepare for the operating system installation. Includes electrical and environmental specifications of the workstation, including the computer unit, monitor, keyboard, and mouse. Ordering Number — 014-001886

Setting Up, Starting, Expanding and Maintaining AViiON® 400, 3000, and 4000 Series Computers
Describes how to prepare a site for installation, unpack and set up 400, 3000 and 4000 series computer systems including starting the computer, and responding to any problems encountered during powerup. Explains how to add or replace drives, memory modules, system board assembly, CPU board, power supply, fan assembly, and PROM components. Includes physical, electrical, and environmental specifications for system components. Use *Setting Up and Installing VMEbus Options in AViiON® Systems* to set up VMEbus options for this machine. Ordering Number — 014-002083

Setting Up, Starting, Expanding and Maintaining AViiON® 530 and 4600 Computers
Describes how to prepare a site for installation, unpack and set up 530 and 4600 computer systems including starting the computer, and responding to any problems encountered during powerup. Explains how to add or replace drives, memory modules, system board assembly, CPU board, power supply, fan assembly, and PROM components. Includes physical, electrical, and environmental specifications for system components. Use *Setting Up and Installing VMEbus Options in AViiON® Systems* to set up VMEbus options for this machine. Ordering Number — 014-002091

Setting Up and Starting AViiON® 5000 Series Computers
Describes how to prepare a site for installation, unpack and set up a 5000 series computer system including starting the computer, and responding to any problems encountered during powerup. Refer to *Expanding AViiON® 5000 Series Systems* for descriptions of how to install add-on boards and mass-storage devices. Includes physical, electrical, and environmental specifications for computer system components. Ordering Number — 014-001806

Starting AViiON® 6000 Series Systems
Describes the basic AViiON 6000 series computer system including hardware components, and maximum configurations. Explains how to power up the computer unit and respond to common power-up problems. Includes electrical and environmental specifications for the computer unit. Ordering Number — 014-001819

Operating the High Availability Disk Array Subsystem
Describes how to configure drive modules in HADA systems as disk arrays, disk mirrors or individual disks. Also describes how to monitor the subsystem and use the automatic recovery procedures for rebuilding a disk array or disk mirror following a drive module failure. Ordering Number — 014-002059

Using the AViiON® System Control Monitor (SCM)
Describes how to use the firmware monitor program to start software, change configuration parameters, and debug your own (non-DG) operating systems. Ordering Number — 014-001802

Setting Up and Installing VMEbus Options in AViiON® Systems
Describes how to jumper VMEbus controllers to operate in an AViiON environment. Explains how to install and remove the controller boards in the system's VMEbus card cage, and how to jumper the VMEbus printed circuit backplane (backpanel) when necessary. Also supplies instructions for connecting external devices to the controller boards and provides some some information on cluster controllers.

This is the primary manual for configuring, installing, removing, and connecting VME distributed adapters and controllers in AViiON systems with 2 VMEbus option board slots. Refer to *Expanding AViiON® 5000 Series Systems* for this information as it pertains to 5000 series systems (6 to 10 slots). Ordering Number — 014-001867

HPS Downloadable Cluster Controller Installation Guide
This is the primary manual for installing, removing, and connecting cluster controllers in conjunction with VME Distributed Adapters (VDA/128 or VDA/255) in AViiON systems. This manual can be used for both Model 20 and Model 30 cluster controllers. Ordering Number — 014-001814

TIPS ORDERING PROCEDURES

TO ORDER

1. An order can be placed with the TIPS group in two ways:
   a) MAIL ORDER – Use the order form on the opposite page and fill in all requested information. Be sure to include shipping charges and local sales tax. If applicable, write in your tax exempt number in the space provided on the order form.

      Send your order form with payment to:
      Data General Corporation
      ATTN: Educational Services/TIPS G155
      4400 Computer Drive
      Westboro, MA 01581-9973

   b) TELEPHONE – Call TIPS at (508) 870-1600 for all orders that will be charged by credit card or paid for by purchase orders over $50.00. Operators are available from 8:30 AM to 5:00 PM EST.

METHOD OF PAYMENT

2. As a customer, you have several payment options:
   a) Purchase Order – Minimum of $50. If ordering by mail, a hard copy of the purchase order must accompany order.
   b) Check or Money Order – Make payable to Data General Corporation.
   c) Credit Card – A minimum order of $20 is required for Mastercard or Visa orders.

SHIPPING

3. To determine the charge for UPS shipping and handling, check the total quantity of units in your order and refer to the following chart:

Total Quantity	Shipping & Handling Charge
1-4 Units	$5.00
5-10 Units	$8.00
11-40 Units	$10.00
41-200 Units	$30.00
Over 200 Units	$100.00

   If overnight or second day shipment is desired, this information should be indicated on the order form. A separate charge will be determined at time of shipment and added to your bill.

VOLUME DISCOUNTS

4. The TIPS discount schedule is based upon the total value of the order.

Order Amount	Discount
$1-$149.99	0%
$150-$499.99	10%
Over $500	20%

TERMS AND CONDITIONS

5. Read the TIPS terms and conditions on the reverse side of the order form carefully. These must be adhered to at all times.

DELIVERY

6. Allow at least two weeks for delivery.

RETURNS

7. Items ordered through the TIPS catalog may not be returned for credit.
8. Order discrepancies must be reported within 15 days of shipment date. Contact your TIPS Administrator at (508) 870-1600 to notify the TIPS department of any problems.

INTERNATIONAL ORDERS

9. Customers outside of the United States must obtain documentation from their local Data General Subsidiary or Representative. Any TIPS orders received by Data General U.S. Headquarters will be forwarded to the appropriate DG Subsidiary or Representative for processing.

TIPS ORDER FORM

Mail To: Data General Corporation
Attn: Educational Services/TIPS G155
4400 Computer Drive
Westboro, MA 01581 - 9973

BILL TO:	SHIP TO: (No P.O. Boxes - Complete Only If Different Address)
COMPANY NAME _____	COMPANY NAME _____
ATTN: _____	ATTN: _____
ADDRESS _____	ADDRESS (NO PO BOXES) _____
CITY _____	CITY _____
STATE _____ ZIP _____	STATE _____ ZIP _____

Priority Code _____ (See label on back of catalog)

_____ _____ _____ _____ _____
Authorized Signature of Buyer (Agrees to terms & conditions on reverse side) | Title | Date | Phone (Area Code) | Ext.

ORDER #	QTY	DESCRIPTION	UNIT PRICE	TOTAL PRICE

A SHIPPING & HANDLING

☐ UPS

	ADD
1-4 Items	$ 5.00
5-10 Items	$ 8.00
11-40 Items	$ 10.00
41-200 Items	$ 30.00
200+ Items	$100.00

Check for faster delivery

Additional charge to be determined at time of shipment and added to your bill.

☐ UPS Blue Label (2 day shipping)
☐ Red Label (overnight shipping)

B VOLUME DISCOUNTS

Order Amount	Save
$0 - $149.99	0%
$150 - $499.99	10%
Over $500.00	20%

Tax Exempt # or Sales Tax (if applicable) _____

ORDER TOTAL	
Less Discount See B	-
SUB TOTAL	
Your local* sales tax	+
Shipping and handling - See A	+
TOTAL - See C	

C PAYMENT METHOD

☐ Purchase Order Attached ($50 minimum)
P.O. number is _____. (Include hardcopy P.O.)
☐ Check or Money Order Enclosed
☐ Visa ☐ MasterCard ($20 minimum on credit cards)

Account Number _____ Expiration Date _____

_____
Authorized Signature
(Credit card orders without signature and expiration date cannot be processed.)

THANK YOU FOR YOUR ORDER

PRICES SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.
PLEASE ALLOW 2 WEEKS FOR DELIVERY.
NO REFUNDS NO RETURNS.

* Data General is required by law to collect applicable sales or use tax on all purchases shipped to states where DG maintains a place of business, which covers all 50 states. Please include your local taxes when determining the total value of your order. If you are uncertain about the correct tax amount, please call 508-870-1600.

Form 702
Rev. 8/87

DATA GENERAL CORPORATION
TECHNICAL INFORMATION AND PUBLICATIONS SERVICE
TERMS AND CONDITIONS

Data General Corporation ("DGC") provides its Technical Information and Publications Service (TIPS) solely in accordance with the following terms and conditions and more specifically to the Customer signing the Educational Services TIPS Order Form. These terms and conditions apply to all orders, telephone, telex, or mail. By accepting these products the Customer accepts and agrees to be bound by these terms and conditions.

1. CUSTOMER CERTIFICATION

Customer hereby certifies that it is the owner or lessee of the DGC equipment and/or licensee/sub-licensee of the software which is the subject matter of the publication(s) ordered hereunder.

2. TAXES

Customer shall be responsible for all taxes, including taxes paid or payable by DGC for products or services supplied under this Agreement, exclusive of taxes based on DGC's net income, unless Customer provides written proof of exemption.

3. DATA AND PROPRIETARY RIGHTS

Portions of the publications and materials supplied under this Agreement are proprietary and will be so marked. Customer shall abide by such markings. DGC retains for itself exclusively all proprietary rights (including manufacturing rights) in and to all designs, engineering details and other data pertaining to the products described in such publication. Licensed software materials are provided pursuant to the terms and conditions of the Program License Agreement (PLA) between the Customer and DGC and such PLA is made a part of and incorporated into this Agreement by reference. A copyright notice on any data by itself does not constitute or evidence a publication or public disclosure.

4. LIMITED MEDIA WARRANTY

DGC warrants the CLI Macros media, provided by DGC to the Customer under this Agreement, against physical defects for a period of ninety (90) days from the date of shipment by DGC. DGC will replace defective media at no charge to you, provided it is returned postage prepaid to DGC within the ninety (90) day warranty period. This shall be your exclusive remedy and DGC's sole obligation and liability for defective media. This limited media warranty does not apply if the media has been damaged by accident, abuse or misuse.

5. DISCLAIMER OF WARRANTY

EXCEPT FOR THE LIMITED MEDIA WARRANTY NOTED ABOVE, DGC MAKES NO WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR PARTICULAR PURPOSE ON ANY OF THE PUBLICATIONS, CLI MACROS OR MATERIALS SUPPLIED HEREUNDER.

6. LIMITATION OF LIABILITY

A. CUSTOMER AGREES THAT DGC'S LIABILITY, IF ANY, FOR DAMAGES, INCLUDING BUT NOT LIMITED TO LIABILITY ARISING OUT OF CONTRACT, NEGLIGENCE, STRICT LIABILITY IN TORT OR WARRANTY SHALL NOT EXCEED THE CHARGES PAID BY CUSTOMER FOR THE PARTICULAR PUBLICATION OR CLI MACRO INVOLVED. THIS LIMITATION OF LIABILITY SHALL NOT APPLY TO CLAIMS FOR PERSONAL INJURY CAUSED SOLELY BY DGC'S NEGLIGENCE. OTHER THAN THE CHARGES REFERENCED HEREIN, IN NO EVENT SHALL DGC BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES WHATSOEVER, INCLUDING BUT NOT LIMITED TO LOST PROFITS AND DAMAGES RESULTING FROM LOSS OF USE, OR LOST DATA, OR DELIVERY DELAYS, EVEN IF DGC HAS BEEN ADVISED, KNEW OR SHOULD HAVE KNOWN OF THE POSSIBILITY THEREOF; OR FOR ANY CLAIM BY ANY THIRD PARTY.

B. ANY ACTION AGAINST DGC MUST BE COMMENCED WITHIN ONE (1) YEAR AFTER THE CAUSE OF ACTION ACCRUES.

7. GENERAL

A valid contract binding upon DGC will come into being only at the time of DGC's acceptance of the referenced Educational Services Order Form. Such contract is governed by the laws of the Commonwealth of Massachusetts, excluding its conflict of law rules. Such contract is not assignable. These terms and conditions constitute the entire agreement between the parties with respect to the subject matter hereof and supersedes all prior oral or written communications, agreements and understandings. These terms and conditions shall prevail notwithstanding any different, conflicting or additional terms and conditions which may appear on any order submitted by Customer. DGC hereby rejects all such different, conflicting, or additional terms.

8. IMPORTANT NOTICE REGARDING AOS/VS INTERNALS SERIES (ORDER #1865 & #1875)

Customer understands that information and material presented in the AOS/VS Internals Series documents may be specific to a particular revision of the product. Consequently user programs or systems based on this information and material may be revision-locked and may not function properly with prior or future revisions of the product. Therefore, Data General makes no representations as to the utility of this information and material beyond the current revision level which is the subject of the manual. Any use thereof by you or your company is at your own risk. Data General disclaims any liability arising from any such use and I and my company (Customer) hold Data General completely harmless therefrom.

Customizing the DG/UX™ System

093-701101-00

Cut here and insert in binder spine pocket

Data General

Data General Corporation, Westboro, Massachusetts 01580