

**Getting Started
With Your
DOMAIN System**

Order No. 002348
Revision 03
Software Release 9.0

**Apollo Computer Inc.
330 Billerica Road
Chelmsford, MA 01824**

Copyright © 1985 Apollo Computer Inc.
All rights reserved. Printed in U.S.A.

First Printing: May, 1983
Latest Printing: July, 1985

This document was produced using the Interleaf document preparation system. (Interleaf is a trademark of Interleaf, Inc.)

APOLLO and DOMAIN are registered trademarks of Apollo Computer Inc.

AEGIS, DOMAIN/IX, DOMAIN/Dialogue, D3M, DPSS, DGR, GMR, and DSEE are trademarks of Apollo Computer Inc.

Apollo Computer Inc. reserves the right to make changes in specifications and other information contained in this publication without prior notice, and the reader should in all cases consult Apollo Computer Inc. to determine whether any such changes have been made.

THE TERMS AND CONDITIONS GOVERNING THE SALE OF APOLLO COMPUTER INC. HARDWARE PRODUCTS AND THE LICENSING OF APOLLO COMPUTER INC. SOFTWARE CONSIST SOLELY OF THOSE SET FORTH IN THE WRITTEN CONTRACTS BETWEEN APOLLO COMPUTER INC. AND ITS CUSTOMERS. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY APOLLO COMPUTER INC. FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY BY APOLLO COMPUTER INC. WHATSOEVER.

IN NO EVENT SHALL APOLLO COMPUTER INC. BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATING TO THIS PUBLICATION OR THE INFORMATION CONTAINED IN IT, EVEN IF APOLLO COMPUTER INC. HAS BEEN ADVISED, KNEW OR SHOULD HAVE KNOWN OF THE POSSIBILITY OF SUCH DAMAGES.

THE SOFTWARE PROGRAMS DESCRIBED IN THIS DOCUMENT ARE CONFIDENTIAL INFORMATION, AND PROPRIETARY PRODUCTS OF APOLLO COMPUTER INC. OR ITS LICENSORS.

PREFACE

Getting Started With Your DOMAIN System introduces you to the basic concepts you'll need to use the DOMAIN® system. It teaches you how to use the keyboard, manage the information displayed on your screen, and to edit text. You'll also learn how to request system services using interactive commands.

You don't have to be an experienced programmer to use this manual. We've written the material for users with little experience using other computers. We carefully define terms and try to avoid computer industry jargon.

This manual also includes examples you can try at your computer while you read. In this way, you can learn by doing.

If you are an experienced programmer, this manual provides an easy-to-read overview of the DOMAIN system. After you read it, you'll be familiar with the terminology we use throughout our documentation.

The Organization of this Manual

We've organized the information in this manual as follows:

- | | |
|-----------|---|
| Chapter 1 | Explains how to use the keyboard and display, and how to give instructions to the system. |
| Chapter 2 | Describes how to manage the information on your display. |
| Chapter 3 | Describes how the system organizes information. |
| Chapter 4 | Teaches you how to create, edit, and read text. This chapter also explains how to print, copy, and delete information stored on the system. |

Chapter 5	Explains how to use the Shell, a program that executes instructions which request system services.
Glossary	Defines terms used throughout our documentation.
Getting Started Command Summary	Summarizes commands and keys described in this manual.

Documentation Conventions

This manual uses the following symbolic conventions:

User Input	Information that you enter appears in color in this typeface .
System Response	Information that the system displays appears in this typeface .
UPPERCASE	Uppercase characters represent commands.
lowercase	Lowercase characters represent information that you supply.
< >	Angle brackets enclose the name of a key on the keyboard.
Bolded term or key	When new terms or keys are introduced, they appear in boldface .
<RET>	The <RET> symbol in command lines indicates when you press <RETURN>.
CTRL/	The notation CTRL/ followed by the name of a key indicates a control key sequence. Hold down <CTRL> while you press the key.

Related Manuals

When you complete this manual, continue with the *DOMAIN System User's Guide* (Order number 005488). It contains more advanced information about the system and describes how to perform various tasks using the system components.

The *DOMAIN System Command Reference* (Order number 002547) contains detailed descriptions of all DOMAIN system commands. The commands are arranged alphabetically for quick and easy access.

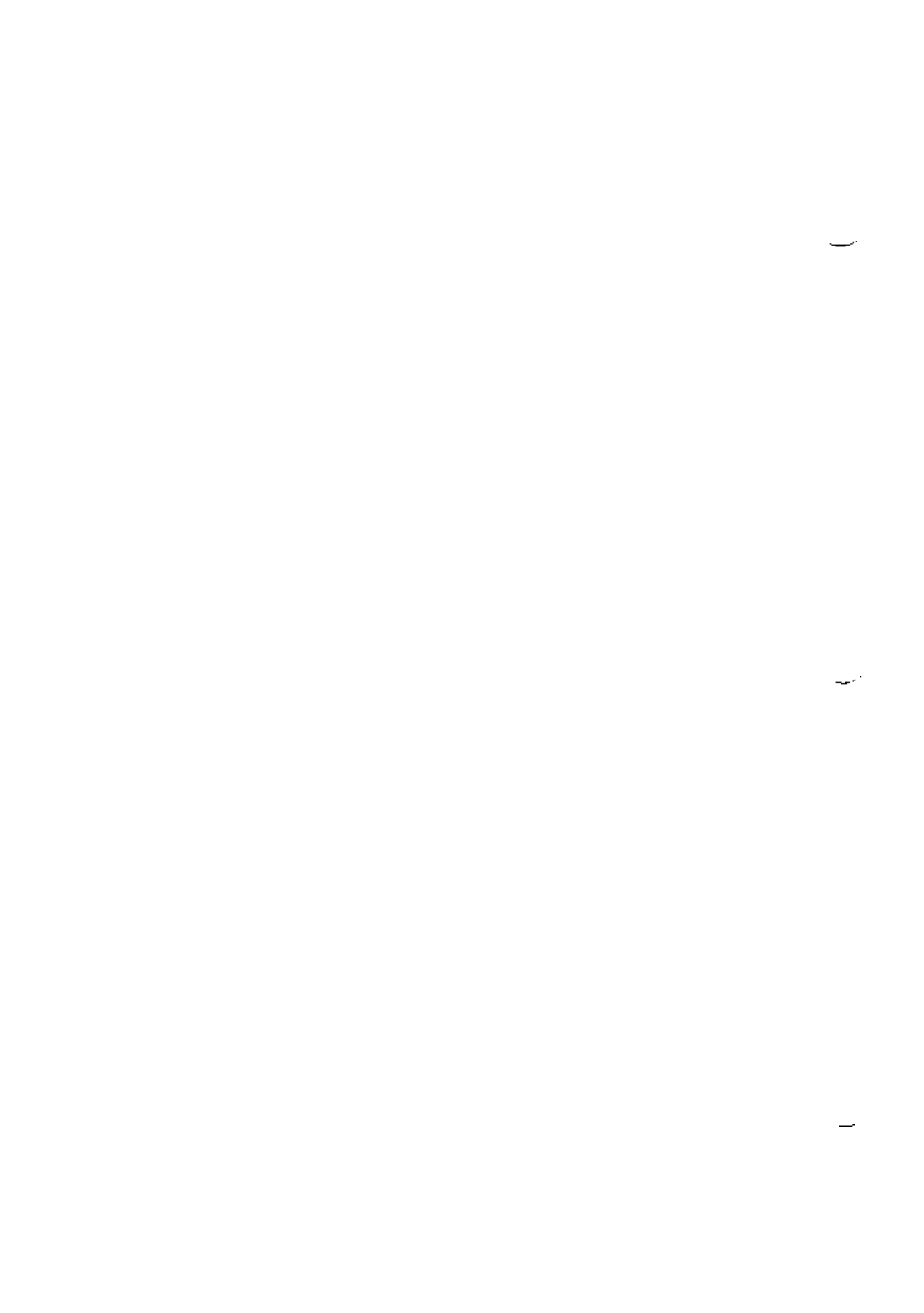
Problems, Questions, and Suggestions

We appreciate comments from the people who use our system. In order to make it easy for you to communicate with us, we provide the User Change Request (UCR) system for software-related comments, and the Reader's Response form for documentation comments. By using these formal channels you make it easy for us to respond to your comments.

You can get more information about how to submit a UCR by consulting the *DOMAIN System Command Reference*. Refer to the CRUCR (Create User Change Request) Shell command description. You can view the same description on-line by typing:

```
$ HELP CRUCR <RETURN>
```

For your documentation comments, we've included a Reader's Response form at the back of each manual.



CONTENTS

Chapter 1 Introduction

Your Keyboard	1-3
Your Display	1-5
Moving the Cursor	1-6
Using the Arrow Keys and the <TAB> Key	1-7
Using the Touchpad	1-8
Using the Mouse	1-9
Logging In	1-9
The Operating System	1-11
Your Display After Logging In	1-11
Entering Commands	1-14
Entering Shell Commands	1-15
Entering Display Manager (DM) Commands	1-15
Using DM Function Keys	1-16
Using Control Key Sequences	1-16
Correcting Typing Errors	1-18
Ending the Session — Logging Off	1-18
Summary	1-19

Chapter 2 Managing Windows and Pads

Looking Inside a Window	2-1
Moving a Pad Under a Window	2-5
Moving to the Top and Bottom of a Pad	2-5
Moving (Scrolling) a Pad Vertically	2-6
Moving (Scrolling) a Pad Horizontally	2-6
Creating a Process	2-6
Managing Windows	2-7
Pushing or Popping a Window	2-8
Responding to Alarms	2-9
Copying Text to the Process Input Window	2-9
Changing Window Size	2-10
Moving a Window	2-12
Stopping a Process	2-12
Summary	2-13

Chapter 3 How the System Organizes Information

Using Pathnames	3-3
Where Am I?	3-4
The Network Root Directory (<i>//</i>)	3-5
Your Node Entry Directory (<i>/</i>)	3-5
Your Home Directory	3-6
Your Working Directory	3-7
Parent Directories	3-8
Your Naming Directory	3-9
Pathname Symbols — A Review	3-9
Using Links	3-11
Summary	3-12

Chapter 4 Using Files

Opening a Window to a File	4-1
Entering Filenames	4-2
Editing a File	4-2
Correcting Errors	4-3
Defining a Range of Text	4-4
Cutting and Pasting (Moving) Text	4-5
Copying and Pasting Text	4-6
Searching for Text	4-6
Cancelling a Copy, Cut, or Search	4-7
Substituting Text	4-7
Undoing Previous Commands	4-8
Closing the EDIT Window	4-8
Reading a File	4-9
Closing the READ Window	4-10
Copying a File	4-10
Printing a File	4-10
Printing a File Using the Print Command	4-11
Printing a File Using the Print Menu	4-11
Deleting a File	4-13
Summary	4-13

Chapter 5 Using the Shell

Command Format	5-2
Using Command Arguments	5-2
Using Command Options	5-2
Entering Multiple Commands on a Line	5-3
Entering Long Command Lines	5-3
Command Line Processing	5-4
Command Search Rules	5-4

Using Wildcards 5-5
Redirecting Input and Output 5-6
 Writing Output to a File 5-6
 Reading Input from a File 5-7
Creating Shell Scripts 5-8
 Substituting Arguments 5-9
Summary 5-9

Glossary Glossary-1

Index Index-1

Getting Started Command Summary

Illustrations

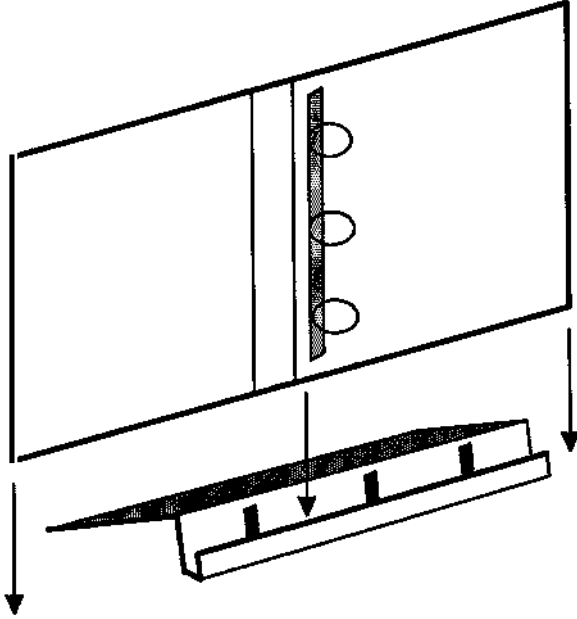
1-1	DOMAIN System Nodes	1-2
1-2	The 880 Keyboard	1-4
1-3	The Low-Profile Keyboard	1-4
1-4	The Landscape Display	1-5
1-5	The Portrait Display	1-6
1-6	Keys that Move the Cursor	1-7
1-7	A Touchpad	1-8
1-8	A Mouse	1-9
1-9	Window Position on a Landscape Display . . .	1-12
1-10	Window Position on a Portrait Display	1-13
1-11	Function Keys that Access Two Commands . .	1-17
2-1	A Window Over a Pad	2-2
2-2	Input and Transcript Pads	2-3
2-3	A Read-Only Window and Pad	2-4
2-4	Pushing and Popping Windows	2-8
2-5	Changing a Window's Size	2-11
3-1	A Sample Naming Tree	3-2
3-2	Sample Pathname	3-4
3-3	A New Directory in the Naming Tree	3-8
3-4	Pathnames Starting with //, /, and \	3-11
4-1	A Sample Editing Session	4-3
4-2	The Print Menu	4-11

Tables

2-1	Keys that Move a Pad	2-5
2-2	Window Control Keys	2-7
3-1	Pathname Starting Point Symbols	3-10

Using the Stand-Up Binder

The plastic page lifter is designed to function as an easel for propping up the binder on your desktop. The following illustration shows how to use it.



CHAPTER

1

Introduction

The **DOMAIN** system consists of two or more of our computers, called **nodes**, linked together with a local area network. Figure 1-1 shows the different **DOMAIN** system nodes.

Each node can use the data, programs, and devices of other network nodes. Each node contains main memory, and may have its own disk, or share one with another node.

The node you're using includes a keyboard and a color or black-and-white display screen. Display management software lets you create several different views, or **windows**, on the screen. Each window is a separate computing environment in which you can execute programs, edit text, or read text. The system can manage many different windows at one time — with each window running its own program. You can move the windows anywhere on your screen, change their size and shape, and overlap or shuffle them as you might papers on your desk.

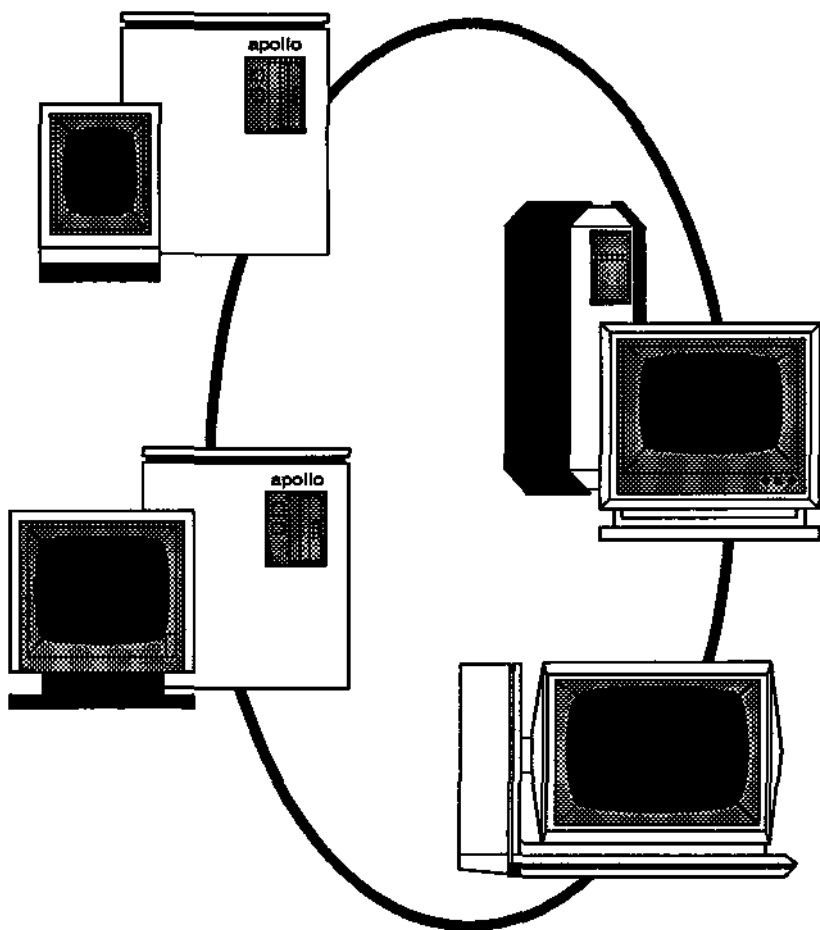


Figure 1-1. DOMAIN System Nodes

This chapter introduces the DOMAIN system and explains the display, how to use the keyboard, and how to issue instructions to the system. The best way to master this material is to try each example at your node while you read.

In the back of this book, you'll find a glossary that defines terms specific to the DOMAIN system. The glossary also lists terms used throughout our documentation.

Your Keyboard

The keyboard you're using resembles either the 880 keyboard in Figure 1-2 or the low-profile keyboard in Figure 1-3.

We provide files of standard key definitions with your system. If the keys on your keyboard don't work as we describe, your system is probably reading the wrong set of key definitions. If this happens, ask your system administrator for help, or see the "Defining Keys" section in Chapter 4 in the *DOMAIN System User's Guide*.

After you become an experienced user, you may want to change the definitions of certain keys. There are two ways to do this. You can either edit a key definitions file or use the Display Manager command **KD (KEY_DEFINITION)**. The *DOMAIN System User's Guide* explains the details of changing key definitions.

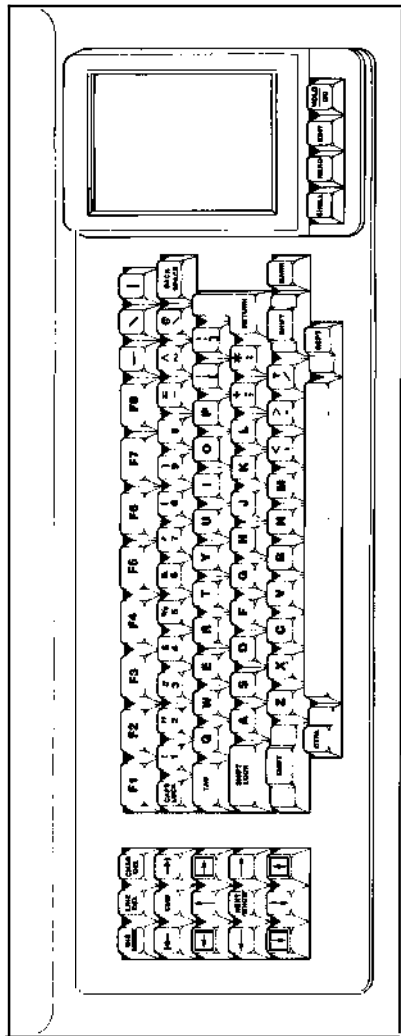


Figure 1-2. The 880 Keyboard

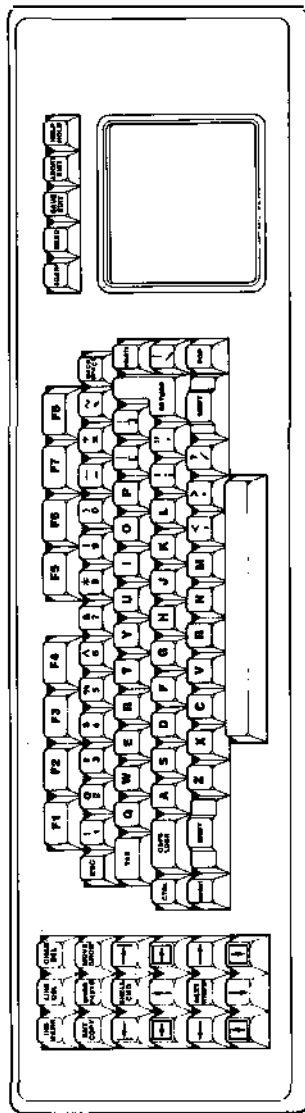


Figure 1-3. The Low-Profile Keyboard

Your Display

The display you're using resembles either the **landscape** (horizontal) display in Figure 1-4 or the **portrait** (vertical) display in Figure 1-5. If your screen looks blank, press any key to turn on the video display. The system automatically shuts off the video display if it is idle for more than 15 minutes.

If your screen still looks blank after pressing a key, your node is not running. For help with starting your node, see your system administrator (the person responsible for system maintenance and security at your installation), or refer to the *Owner's Guide* for your particular node.

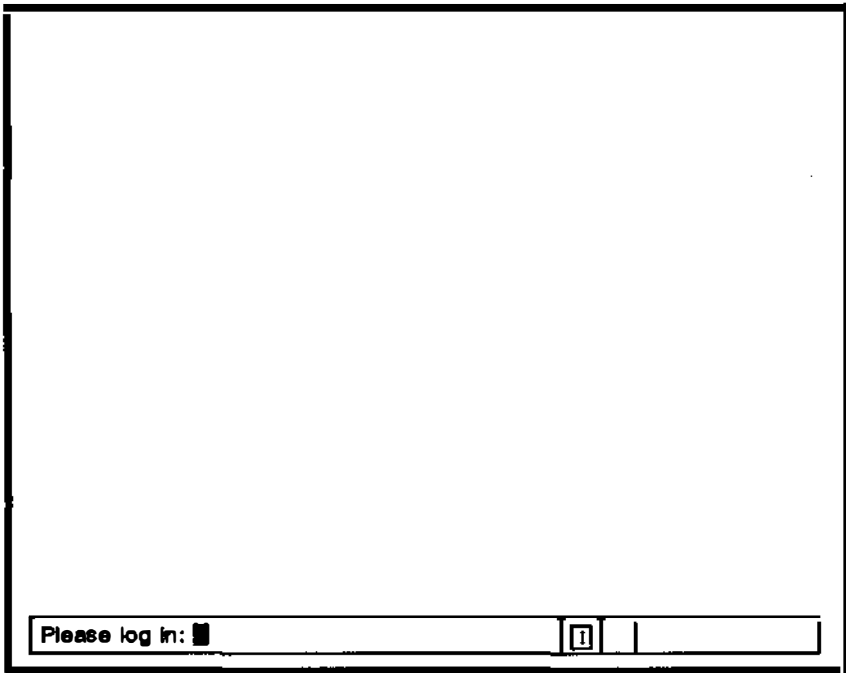


Figure 1-4. *The Landscape Display*

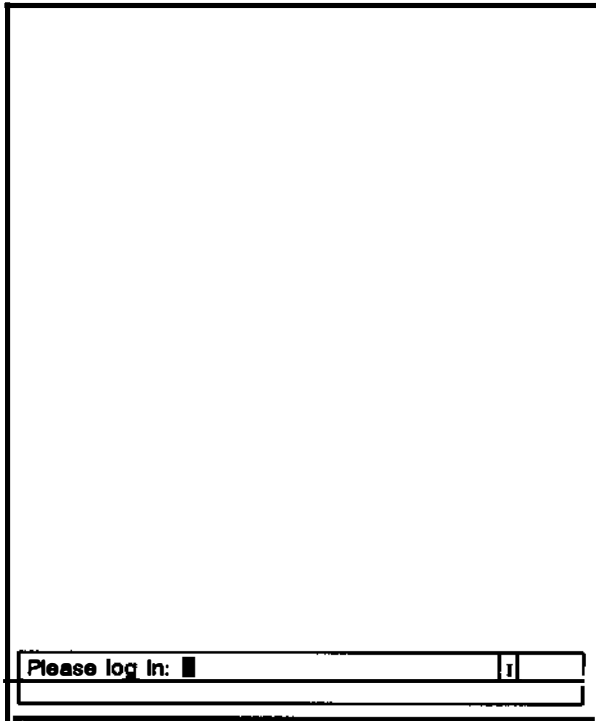


Figure 1-5. The Portrait Display

Notice the small, blinking box in the lower, left corner of your display. This is the **cursor**. The box cursor indicates where the system will display the information you type at your keyboard.

As you use the system, you may see two other cursor shapes: an arrow cursor (↑) and a cross in a circle cursor (⊕). The arrow cursor appears when you use the touchpad or mouse to move the cursor (see the “Using the Touchpad” and “Using the Mouse” sections). The circle cursor appears when the system is so busy that it cannot immediately respond to your commands. As a new user, we suggest that you simply wait until the circle cursor changes back to a blinking box or an arrow before you enter additional commands.

Moving the Cursor

Moving the cursor is the first step in learning to use the system. As you work through the examples in this book, you’ll see that you

must position the cursor at a specific location on your display before giving an instruction to the system.

To move the cursor, you use the arrow keys, the <TAB> key, a touchpad, or a mouse. The touchpad and the mouse are optional pieces of equipment. The arrow keys and <TAB> are available on all keyboards.

As you read about the ways to move the cursor in the next three sections, try using the keys, and the touchpad or mouse if you have one.

Using the Arrow Keys and the <TAB> Key

You can use any of the keys highlighted in Figure 1-6 to move the cursor. To move the cursor using the arrow keys (← → ↑ ↓), press the arrow key that points in the direction you wish to move, and hold it down until the cursor reaches the desired destination.

To move to the beginning or end of a line displayed beneath the cursor, use the keys labeled |← and →|.

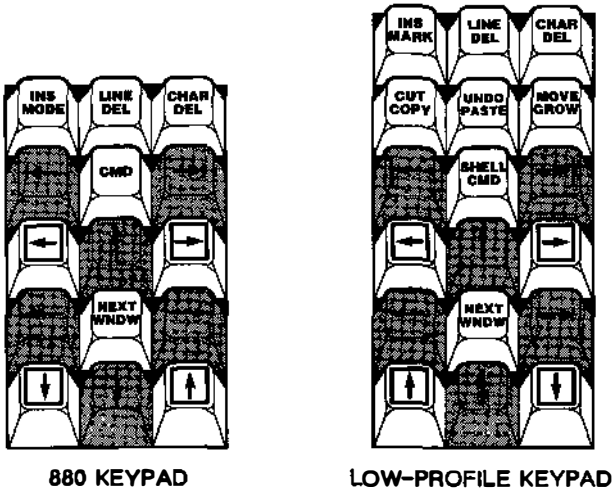


Figure 1-6. Keys that Move the Cursor

In addition to the arrow keys, you can use the <TAB> key to move the cursor. Pressing <TAB> moves the cursor to the right until it hits the next tab stop. Holding down <CTRL> while you press <TAB> causes the cursor to tab left. Please note that this key only positions the cursor; it does not insert a tab character.

Using the Touchpad

If you have a touchpad, you'll find it on the right side of your keyboard. (See Figure 1-7.) The touchpad is made of a pressure-sensitive material. When you press the material with your finger, the touchpad transmits data to the system, and the system moves the cursor.

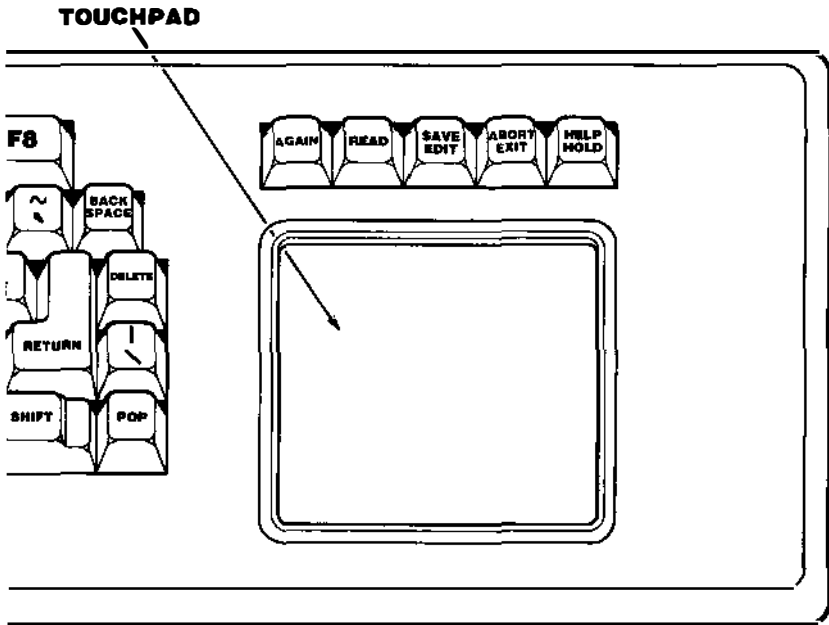


Figure 1-7. A Touchpad

While you're touching the material, the cursor appears as an arrow; when you lift your finger, the cursor reverts to a blinking box.

Take care not to puncture or scratch the touchpad's conductive material. Never use a sharp object on the touchpad. Scratches or punctures can move the cursor into undesired screen positions.

Using the Mouse

The mouse is a small, handheld device that you move across a flat surface, such as your desktop or a pad of paper. (See Figure 1-8.) A small ball bearing in the base of the mouse detects motion, the device transmits data to the system, and the system moves the cursor.

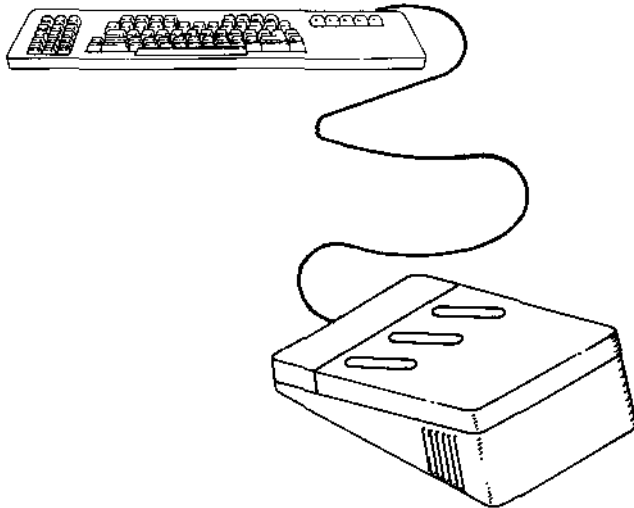


Figure 1-8. A Mouse

When you're moving the mouse, the cursor appears as an arrow; when you stop moving the mouse, the cursor reverts to a blinking box.

The mouse has three keys attached to it. You can use these keys to manipulate windows and read files. In Chapter 2, you'll learn how to expand, shrink, and shuffle windows with the mouse. Chapter 4 explains how to use the mouse to read files.

You can redefine the mouse keys to execute other commands. Refer to Chapter 3 in the *DOMAIN System User's Guide* for the details.

Logging In

When your node is running and the video display is turned on, the message "Please log in: " appears at the bottom of the screen. This message is called a prompt. Programs use prompts to indicate that

they are ready for you to type a command (instruction) at the keyboard.

The "Please log in:" prompt indicates that the system is waiting for you to enter your **username** and **password**. If you don't know what names to enter, ask your system administrator. This administrator defines a user account for every person authorized to use the system.

Your user account contains the name the computer uses to identify you (username), and your password. If security is important at your installation, user accounts might also contain project and organization names. The system uses this information to determine who can use the system and what resources they can use.

As we said earlier, the cursor indicates where the system will display the commands you type at your keyboard. Before logging in, move the cursor to the right of the "Please log in:" prompt.

To log in, follow these two steps:

1. Enter the log-in **L (LOGIN)** command, a blank space, and your username. For example:

Please log in: L USERNAME

If your system administrator says that project and organization names are required, also type these names. For example:

Please log in: L USERNAME.PROJECT.ORGANIZATION

If you make a mistake, press the **<BACKSPACE>** key. **<BACKSPACE>** works like the same key on a typewriter, except that it deletes characters as it moves the cursor back toward the beginning of the line.

When you have typed the required names correctly, press **<RETURN>** (shown in examples as **<RET>**) to submit this information to the system. (Press **<RETURN>** whenever you wish to submit the line you've typed to the system.) The system accepts the line you submitted and requests your password.

2. Type your password. The system does not display the password you enter, but displays a dot for each character in the password. For example:

Password: ... <RET>

If the system finds a user account that matches the names you supply, it displays a message at the bottom of your screen in the following format:

Logged in as user.project.org date time

If the system cannot find a user account that matches the names you supply, it repeats the log-in prompt and displays this message:

! name [project [org]] [-p] [-h]

The message shows you the format of the log-in command. If you receive this message, you are not using a valid username and password. Ask your system administrator for help.

The Operating System

The operating system is a program that supervises the execution of other programs on your node. It has two main parts: the Shell and the Display Manager (DM).

The Shell program provides access to the traditional computing operations, such as printing documents, compiling and running programs, and monitoring system activities. The Shell “listens” for commands that you type at your keyboard. Shell commands invoke utilities, which are programs that perform the tasks you request. For example, when you type DATE, the Shell invokes the utility program that displays the date.

The Display Manager, on the other hand, handles operations that modify the display on your screen, and create and edit files. The Display Manager also supervises the creation of computing environments in which you can execute programs. These computing environments are called processes. After you log in, the Display Manager creates a process in which the Shell program is running. (Chapter 4 in the *DOMAIN System User's Guide* provides detailed information about processes.)

Your Display After Logging In

After you log in, the system creates a process running the Shell program and presents some windows on your screen. The position of these windows depends upon the type of display you're using. Figure 1-9 shows a window position on a landscape display. Figure 1-10 illustrates a window position on a portrait display.

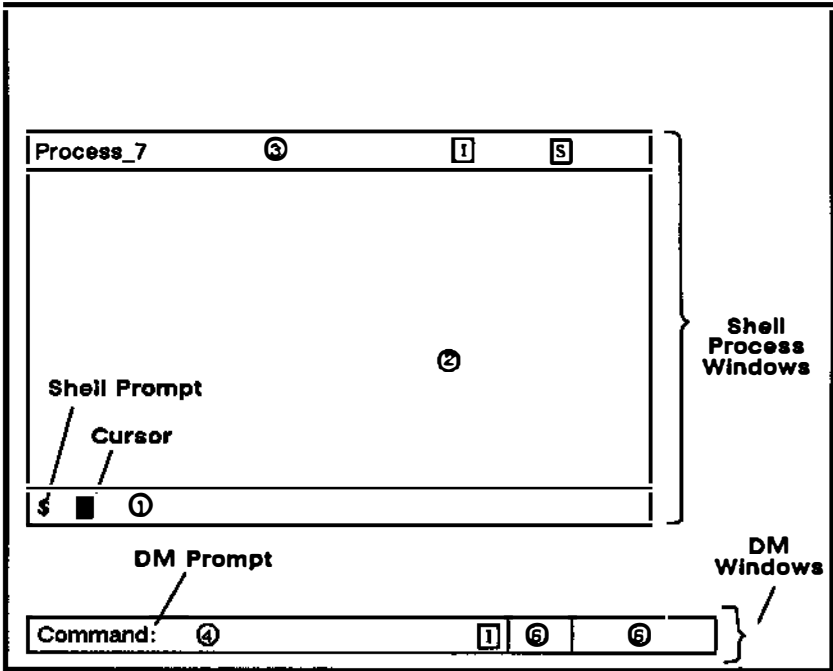


Figure 1-9. Window Position on a Landscape Display

Notice that the cursor is in the lower, left corner of your screen. When you logged in, you saw that the cursor position indicates where the system displays the commands you type. The cursor position also indicates which program (Shell or DM) receives your commands. Because the Shell doesn't recognize DM commands and the Display Manager doesn't recognize Shell commands, you must move the cursor into the appropriate window before entering a command.

You enter Shell commands (commands that invoke other programs) in the window that contains the dollar sign (\$) prompt. To move the cursor next to the \$ prompt, press <NEXT WNDW>. You may have to press <NEXT WNDW> more than once to move the cursor next to the \$ prompt.

You enter DM commands (commands that control windows and processes, and create and edit files) in the window that contains the "Command:" prompt. To move the cursor next to the "Command:" prompt, press <CMD>.

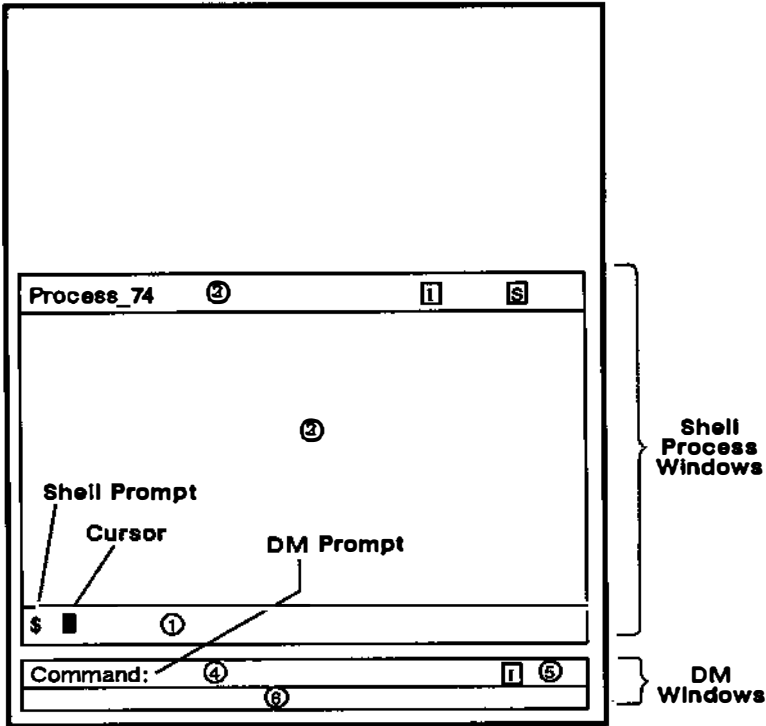


Figure 1-10. Window Position on a Portrait Display

Now let's look more closely at the parts of the initial display. Refer to Figures 1-9 and 1-10 as you read about the Shell process windows and the DM windows in the next few paragraphs. The numbers in the following list correspond to the numbers in these figures.

We'll begin with the Shell process windows.

1. The process input window contains the dollar sign (\$) prompt. You type Shell commands in the process input window. The Shell reads the commands from the process input window and invokes the appropriate utility.

When you use the Shell to execute another program, that program may display its own prompt in the process input window. The Shell prompt (\$) appears when the Shell is "listening" for your next instruction.

2. **The process output window** is the large window above the process input window. It displays your commands (after you press <RETURN>) and the Shell's response to your commands. The response can be information you requested, a process status report, or an error message. The process output window is a transcript, or record, of your interaction with the system.
3. **The window legend** is the top border of the process output window. It contains the process identification name (Process_xx) and the letters I and S. The letters are mode indicators that allow you to change the process environment. You'll learn about the mode indicator I at the end of this chapter; refer to Chapter 4 in the *DOMAIN System User's Guide* for information about the mode indicator S.

The DM windows are at the bottom of your screen.

4. **The DM input window** contains the "Command:" prompt. In this window, you type DM commands and the DM reads the commands you type. (Later in this chapter, you'll learn other ways to enter DM commands.)
5. **The DM alarm window** displays a small pair of bells when a process displays a message in an output window that is hidden by an overlapping window. The "Responding to Alarms" section in Chapter 2 explains more about this window.
6. **The DM output window**, like the process output window, displays messages. However, the DM does not display the commands you enter.

Entering Commands

Now that you're familiar with the parts of the display, try entering some Shell and DM commands. The examples in this book include prompts (\$ and Command:). We include the prompt to help you determine which input window to enter the command in. Don't type the prompt, just enter the command line shown in color.

We show commands in uppercase letters for emphasis. You can enter commands in uppercase or lowercase. Remember to press <RETURN> at the end of each command line.

Don't worry if you make a typing mistake when you enter a command. You won't damage the system; you'll simply cause it to display an error message on the screen. If you notice your mistake before you press <RETURN>, you can correct it using

<BACKSPACE>. **<BACKSPACE>** deletes characters as it moves the cursor back toward the beginning of the line. (The "Correcting Typing Errors" section later in this chapter explains other ways to correct mistakes.)

Entering Shell Commands

Let's try entering a Shell command. Move the cursor next to the Shell prompt (\$) and enter the **DATE** command. For example:

```
$ DATE <RET>
```

Press **<RETURN>** to submit the command. In the process output window, the Shell displays the command you typed and then invokes the utility program that displays the date and time.

The **DOMAIN** system provides a **HELP** facility that displays information about Shell and DM commands. To display introductory information about the **HELP** facility, type

```
$ HELP <RET>
```

In a separate window, you'll see an overview of the **HELP** facility. When you have finished reading the information, move the cursor into the **HELP** window. To remove the window from your screen, hold down **<CTRL>** while you press **<N>**.

Entering Display Manager (DM) Commands

There are three ways to enter DM commands:

- Type the command in the **DM** input window (next to the "Command:" prompt).
- Press a specially-defined (and labeled) key, called a function key.
- Press a control key sequence (**<CTRL>** combined with another key).

Try typing a command in the **DM** input window. First, press **<CMD>** to move the cursor next to the "Command:" prompt. Now type

```
Command: RS <RET>
```

After you press <RETURN>, the **RS (REFRESH_SCREEN)** command refreshes the entire screen. You'll see the display blink as the DM clears the screen and redraws windows.

Using DM Function Keys

You usually won't type DM command names at your keyboard. Instead, you will use certain single keys, called **DM function keys**, that invoke DM commands.

If you're using a low-profile keyboard, note that each function key highlighted in Figure 1-11 provides access to two commands. For example, the function key labeled **CUT** and **COPY** allows you to delete (cut) text and to copy text. To enter the command in the uppermost position (**CUT**, for example), hold down <SHIFT> while you press the function key. To enter the command in the lowermost position (**COPY**, for example), simply press the function key.

Let's try using a function key. If you're using an 880 keyboard, press <SHELL>. If you're using a low-profile keyboard, hold down <SHIFT> while you press <SHELL>. This key issues the DM command that creates a new process running the Shell program, and displays the windows associated with the new Shell process. You can create a new Shell process at any time. You may use it to execute programs while your initial Shell process is busy performing other tasks.

Using Control Key Sequences

Often you'll use control key sequences to invoke DM commands. To enter a control key sequence, hold down <CTRL> while you press another key. (In a previous exercise, you used the control key sequence **CTRL/N** to remove the **HELP** window from your screen.)

We'll now use another control key sequence (**CTRL/Z**) to stop the Shell process you just created. First, move the cursor into the Shell input window. Hold down <CTRL> while you press <Z>. The system displays

```
$ ***EOF***  
***Pad Closed***
```

The process input window disappears from your screen, but the output window remains. To remove this window from your screen, keep the cursor within the window and press **CTRL/N**.

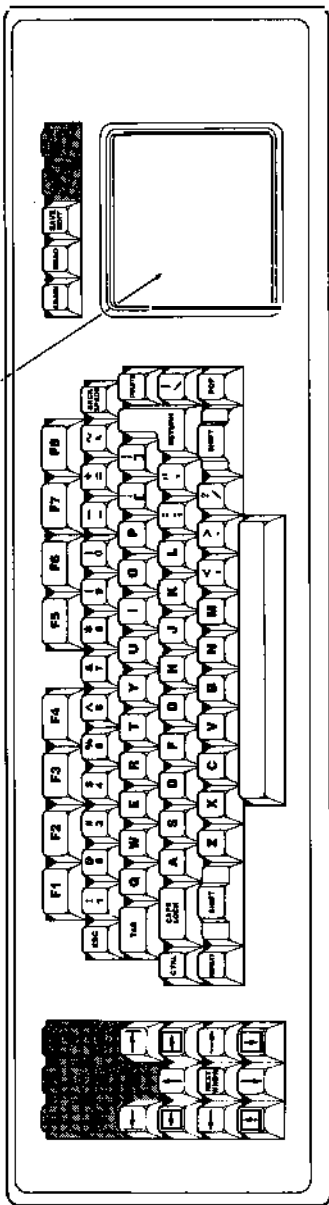


Figure 1-11. Function Keys that Access Two Commands

Correcting Typing Errors

The keys listed here help you correct any errors you might make typing commands:

- BACKSPACE
- CHAR DEL
- LINE DEL
- INS (or INS MODE)

As you learned earlier in this chapter, **<BACKSPACE>** deletes characters as it moves the cursor back toward the beginning of the line. For example, if you type DATE and then press **<BACKSPACE>**, the letter E disappears.

<CHAR DEL> deletes the character at the current cursor position. For example, if you position the cursor on the D in DATE and press **<CHAR DEL>**, the D disappears.

<LINE DEL> deletes the entire line, no matter where the cursor is positioned on the line.

Notice the letter I in the Shell's window legend. The letter also appears to the right of the DM input window. It indicates that the DM and Shell input windows are operating in insert mode. Insert mode enables you to change command lines in the input window by repositioning the cursor and inserting characters. The rest of the line moves right as you insert additional characters. The system inserts the text you enter rather than overstriking the existing command line.

To overstrike the command line, turn off insert mode by pressing **<INS MODE>** on 880 keyboards or **<INS>** on low-profile keyboards. If the cursor is in the process input window, the letter I in the window legend disappears. If the cursor is in the DM input window, the letter I following the DM input window disappears. The system then replaces existing characters with the new characters you type. You can turn insert mode back on by pressing **<INS MODE>** or **<INS>** again.

Ending the Session — Logging Off

When you're ready to end the session, log off the system. Logging off prevents others from using your user account. Log off if your node is in a public place.

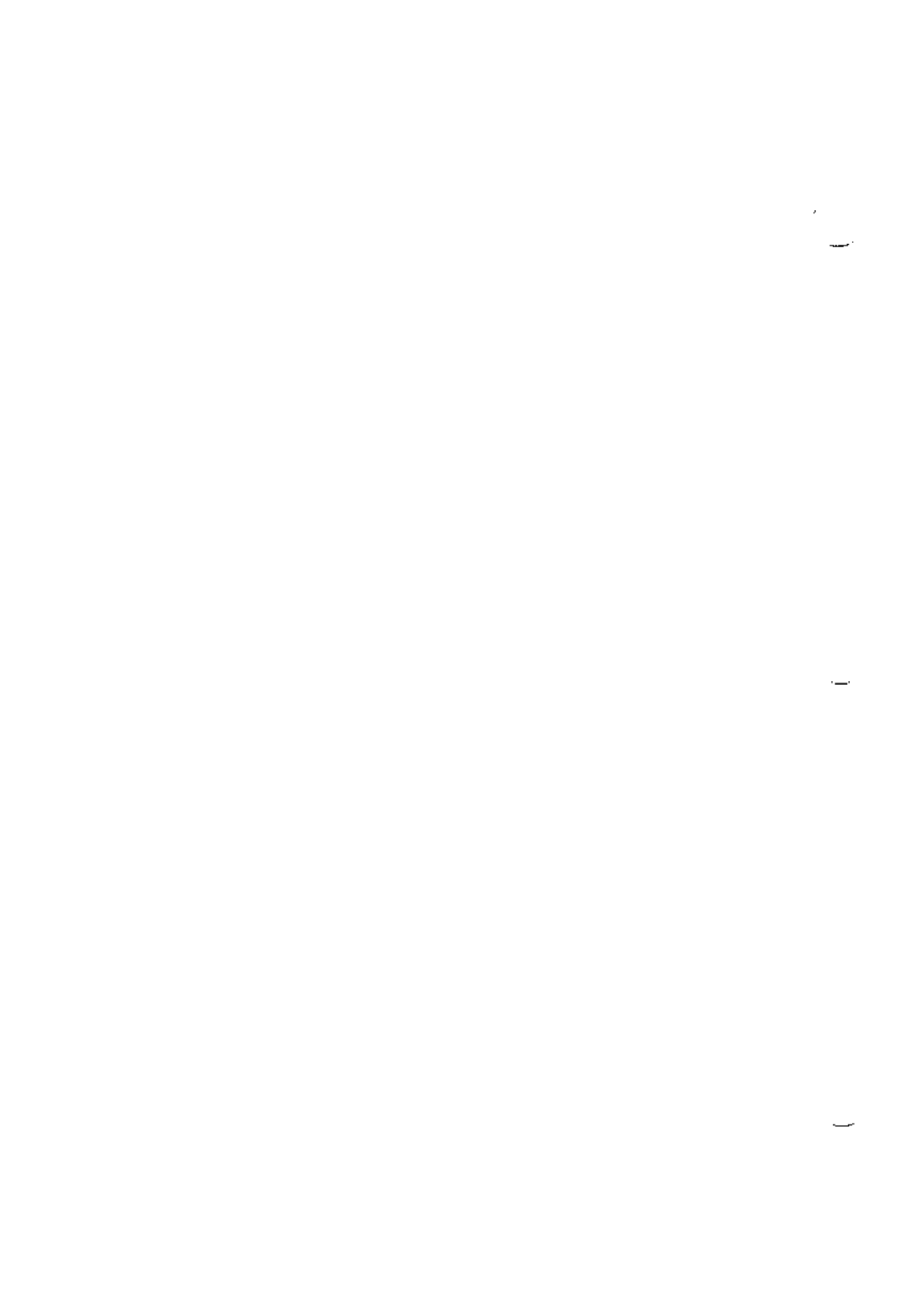
To log off, type the following command in the DM input window:

Command: LO <RET>

If you've already stopped each process, the DM displays its original "Please log in: " prompt. If you do not stop processes before logging off, the DM stops them for you then displays the "Please log in:" prompt.

Summary

Nice work. You have mastered the system's basic tools. You can log in and off, execute DM and Shell commands, and create a new Shell process and its windows. In the next chapter, you'll learn how to use other DM commands to manipulate windows and processes. Continue reading when you are ready to learn more about DOMAIN processing.



Managing Windows and Pads

Chapter 1 introduced you to the concepts of windows and processes. This chapter describes how to use the Display Manager (DM) to control the windows and processes on your display. Using DM function keys and control key sequences, you will learn how to:

- View hidden information in a window
- Shuffle windows
- Change the size of windows
- Create and stop Shell processes

Looking Inside a Window

Windows are pretty much what you think they are: openings on the screen that let you view information stored in the system. **Pads** are containers that hold the information you display. The information

can be commands that you type, responses from the system, text files, or pictures.

In other words, a window provides a view of a pad. The window can present the entire pad, or only show part of the pad. Figure 2-1 illustrates a window over a pad, where only part of the pad is visible through the window.

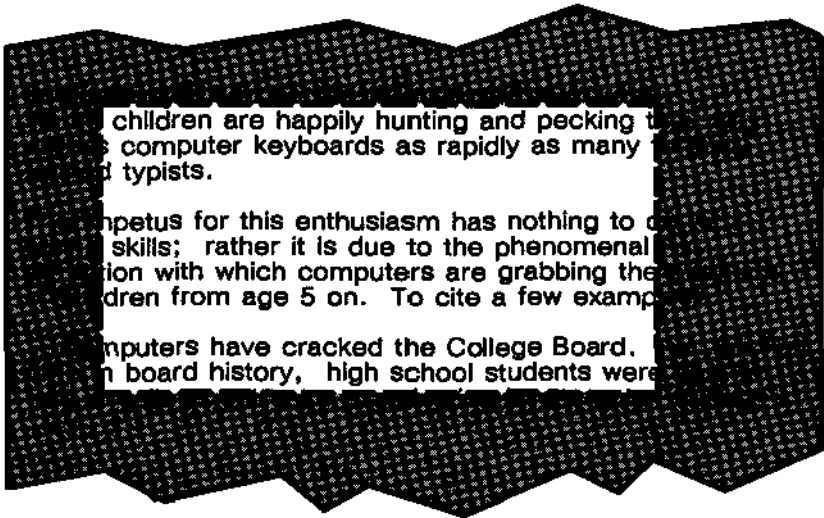


Figure 2-1. A Window Over a Pad

The dotted line in Figure 2-1 represents the window's edges. On the display, you can see only the characters inside the edges of the window. The rest of the pad is hidden. There are two ways to view the hidden parts of a pad: you can either move the pad under the window or enlarge the window. Later in this chapter, you'll learn both ways to view hidden information.

Windows have such attributes as size and position on the screen, position over a pad, and possibly color (if your node is a color model). You can imagine them as pieces of paper that you can stack and shuffle around on the screen just as you would if they were on your desk. Later in this chapter, you'll learn how to manipulate windows, but first we'll describe the kinds of pads the DM uses and the keys that control them.

The DM provides three kinds of pads: input, transcript, and edit pads. Input pads accept the commands you type at your keyboard. In Chapter 1, you typed commands through windows onto input pads.

Transcript pads (process output pads) keep a running record, or "transcript," of your interaction with programs by displaying your input and the program's output. You view transcript pads through process output windows. In the examples in Chapter 1, the Shell displayed the commands you entered and its response to your commands on the transcript pad.

Figure 2-2 shows a Shell process input pad, a DM input pad, and a transcript pad.

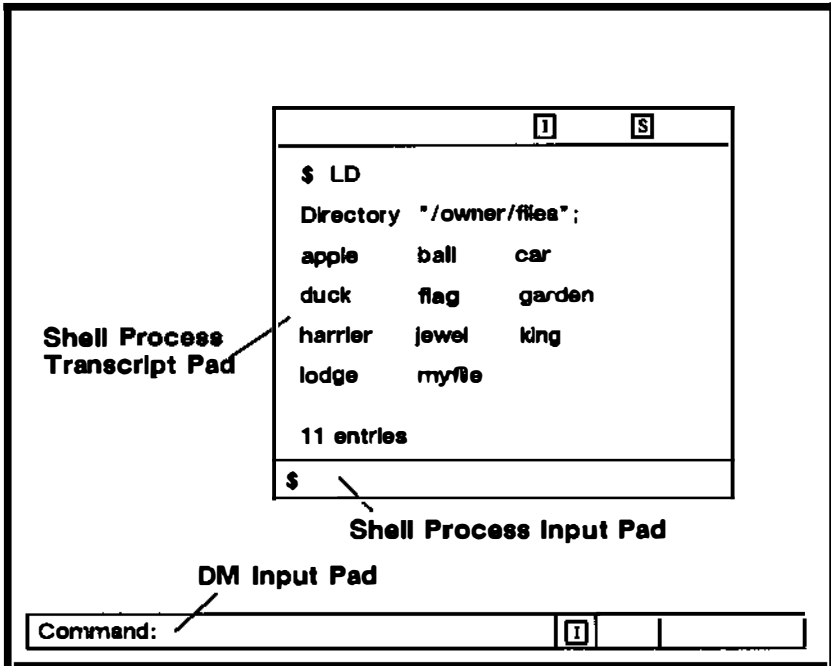


Figure 2-2. Input and Transcript Pads

Edit pads contain copies of files that the DM displays when you press <READ> or <EDIT>. You can change text displayed on pads created with <EDIT>. When you close the pad, the DM writes your changes to the permanent file. Pads created with <READ> are read-only; you can copy text from them, but you cannot add or delete text. Chapter 4 explains reading and editing files in detail.

To see what an edit pad and window look like on your display, you can read a text file supplied with your system. Press <READ>. The cursor moves into the DM input window, and the DM displays the prompt "Read file:".

To read a file called */Doc/SR9.Release_Notes* that describes the current software version, type the filename and press <RETURN> as follows:

Read file: /Doc/SR9.Release_Notes <RET>

The DM creates a read-only window and pad to display the file */Doc/SR9.Release_Notes*. See Figure 2-3.

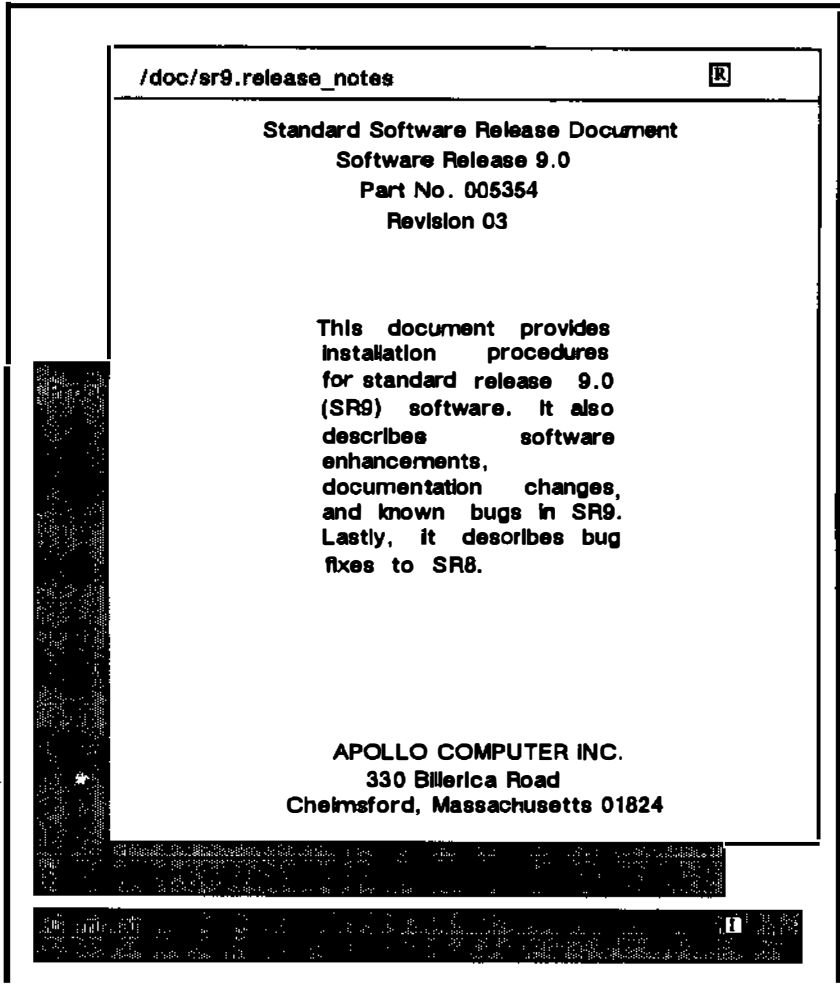








Figure 2-3. A Read-Only Window and Pad

For now, leave this read-only window and pad on your display. You will use it to learn about the keys discussed in the next several sections.

Moving a Pad Under a Window

The keys listed in Table 2-1 move a pad around under a window so that hidden text is in view.

Table 2-1. Keys that Move a Pad

Task	Predefined Key	
	Low-Profile Keyboard	880 Keyboard
Move to first character in a pad	CTRL/T	CTRL/T
Move to last character in a pad	CTRL/B	CTRL/B
Move pad by pages		
		
Move pad by lines	<SHIFT> / ↑	<F2>
	<SHIFT> / ↓	<F3>
Move pad by characters	<SHIFT> / ←	
	<SHIFT> / →	

As you read about moving a pad with the keys discussed in the next three sections, try using them to move the pad containing */Doc/SR9.Release_Notes*.

Moving to the Top and Bottom of a Pad



The CTRL/T sequence causes the DM to position the cursor at the top of the current pad. CTRL/B performs the complementary function of positioning the cursor at the bottom of the pad.

Since the cursor is at the top of */Doc/SR9.Release_Notes*, let's move it to the bottom. Press CTRL/B. (Note that CTRL/B may take a while because the file is long.) The DM positions the cursor on the last character in */Doc/SR9.Release_Notes*. Now press CTRL/T to move the cursor back to the first character in the file.

Moving (Scrolling) a Pad Vertically

The vertical scrolling keys  and  scroll a pad vertically under a window in units of half the height of the window.

The DM command PP (PAD_PAGE) allows you to modify the amount of text that the boxed arrow keys scroll. See the description of the PP command in Chapter 4 in the *DOMAIN System User's Guide*.

You may also scroll a pad vertically by lines rather than by pages. The <SHIFT>/ and <SHIFT>/ sequences on low-profile keyboards, and <F2> and <F3> on 880 keyboards, move the pad up and down one line, respectively. The cursor position does not change relative to the pad; the pad simply slides by under the window.

Moving (Scrolling) a Pad Horizontally

The  and  keys scroll a pad left and right under a window in 10-character increments.

The left and right scrolling keys <SHIFT>/← and <SHIFT>/→ on low-profile keyboards scroll a window over a pad in single character increments.

When you have finished experimenting with the keys that move a pad under a window, close the window and pad containing */Doc/SR9.Release_Notes*. To do so, keep the cursor in the window and press <ABORT> (on low-profile keyboards) or CTRL/N (on 880 keyboards).

Creating a Process

The DM controls not only the windows and pads associated with your screen, but also the creation and destruction of Shell processes in which you do all those other things commonly associated with using a computer (compiling and running your own programs, for instance).

You can create a new Shell process any time you need another computing environment. For example, if your initial Shell process is busy compiling a program, you can create a new Shell process to perform another task, such as formatting a text file.

In Chapter 1, you used the <SHELL> key to create a new process running the Shell program. Try it again. Press <SHELL>. The DM creates a new Shell process, along with all the necessary pads and windows. On your screen, you'll see input and transcript (output) pads as well as windows into those pads. The DM places the cursor next to the Shell prompt (\$) in the new process input window, and the process is ready to receive Shell commands. (Chapter 5 provides additional information about the Shell and its commands.)

In the next few examples, you'll use the new Shell process windows to learn how to shuffle and stack windows, and to change a window's size and position on the screen.

Managing Windows

The window control keys listed in Table 2-2 change the size, position, and characteristics of windows on the screen. In addition, all window control keys cause the DM to completely display the selected window if any part of it is hidden.

Table 2-2. Window Control Keys

Task	Predefined Key	
	Low-Profile Keyboard	880 Keyboard
Shuffle windows	<POP>	CTRL/P
Copy text to process input window	<AGAIN>	<F8>
Enlarge or reduce a window	<GROW>	CTRL/G
Move a window	<MOVE>	CTRL/W

Pushing or Popping a Window

Pushing or popping windows allows you to display windows that are partially or completely hidden by other windows on your screen. Pressing <POP> (on low-profile keyboards) or the CTRL/P sequence (on 880 keyboards) pops a window to the top of the pile or pushes a window to the bottom of the pile of windows on the screen. These keys allow you to shuffle windows the way you would papers on your desk. Figure 2-4 illustrates how <POP> and CTRL/P work.

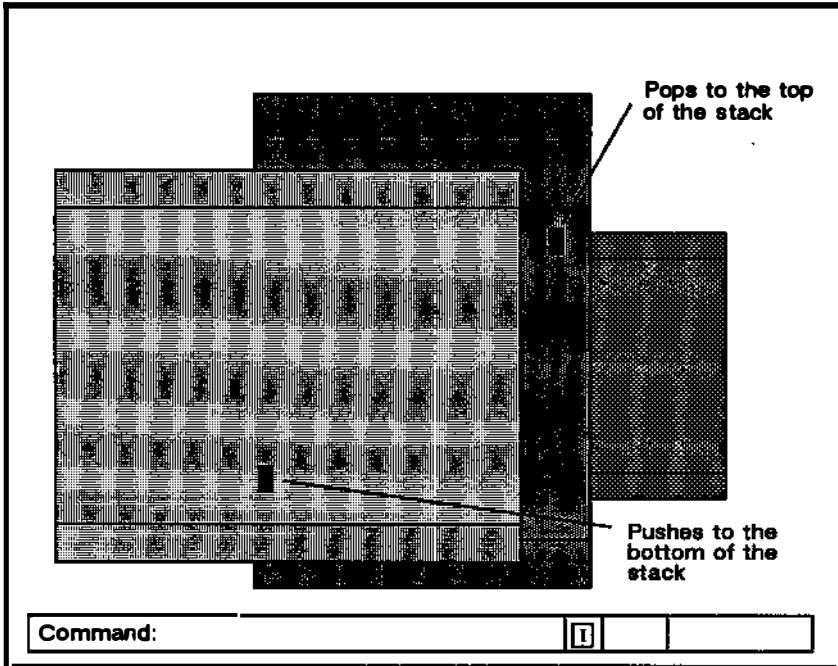


Figure 2-4. Pushing and Popping Windows

Try using <POP> or CTRL/P to shuffle the windows on your screen. Move the cursor into a window that is partially hidden by another window. Press <POP> or CTRL/P. The window pops to the top of the pile.

Now move the cursor into a window that is completely visible. Press <POP> or CTRL/P again. This time the DM pushes the window to the bottom of the pile.

If you have a mouse, you can use its center key to push and pop windows.

Responding to Alarms

Experiment with `<SHELL>` and `<POP>` or `CTRL/P` until you're comfortable manipulating windows. Now, position the windows so that one window overlaps the other's input window. Use `<POP>` or `CTRL/P` to bring the hidden window to the top, and type

`$ HELP date <RET>`

Quickly press `<POP>` or `CTRL/P` again and send the top window to the bottom of the stack.

If this exercise worked, the DM displays two small bells in its alarm window. If your system has a speaker, the DM also emits an alarm tone. The alarm informs you that the system is responding to a window you can't see. Enter the DM command `AP` (`ALARM_POP`) to look at the window requesting your attention. For example, press `<CMD>` and type

Command: `AP <RET>`

After you press `<RETURN>`, the DM presents the hidden window. Refer to Chapter 4 in the *DOMAIN System User's Guide* for more information on alarms.

For now, leave the Help window on your screen; we'll copy one of the sample commands from it in the next section.

Copying Text to the Process Input Window

Pressing `<AGAIN>` (on low-profile keyboards) or `<F8>` (on 880 keyboards) is an easy way to repeat a one-line Shell command that you typed earlier. These keys copy a line from a transcript pad into the process input window.

To see how this works, copy the command `HELP date` that you typed earlier. Follow these steps:

1. Move the cursor into the transcript pad, and position it on the `H` in `HELP`.

2. Press **<AGAIN>** or **<F8>**. The DM copies all the text from the current cursor position to the end of the line, and displays the copied text in the process input window.

If there is anything in the process input window waiting for processing, the DM appends the copied line to that text. Once you have copied the command line to the process input window, you can edit the line or enter it just as it is by pressing **<RETURN>**.

Since the DM appends new information to the transcript pad, the command you want to copy may be hidden. Use the scrolling keys to move the transcript pad until the command is in view.

You can also copy text from other pads (pads displaying Help information or pads created with **<READ>** or **<EDIT>**, for example), and display the text in the process input window. You first mark the text you want to copy and then use **<COPY>** and **<PASTE>** (on low-profile keyboards) or **CTRL/C** and **CTRL/O** (on 880 keyboards).

Let's copy one of the sample commands from the pad displaying Help information about the **DATE** command. If the Help window is hidden, use **<POP>** or **CTRL/P** to bring it to the top. Then follow these steps:

1. Position the cursor on the *d* shown in the second example (`$ date -t`) at the bottom of the Help window.
2. Press **<MARK>** then **→**. The DM highlights the sample command.
3. Now press **<COPY>** or **CTRL/C**.
4. Move the cursor into the process input window.
5. Press **<PASTE>** or **CTRL/O**. The DM displays the copied command in the process input window.
6. Press **→** to move the cursor to the end of the pasted line.

After you press **<RETURN>** to enter the command, you'll see the current system time displayed on the transcript pad.

Changing Window Size

If you want to make a window larger or smaller, use **<GROW>** (on the low-profile keyboards) or the **CTRL/G** sequence (on 880 keyboards). The DM lets you select a window edge or corner,

which you can move across the screen to change the window's size. To change the size of one of the windows on your screen, follow these steps:

1. Decide which edge or corner you want to change, and move the cursor to it.
2. Press **<GROW>** or **CTRL/G**. A flexible "rubberband" border appears. Figure 2-5 illustrates this flexible border.
3. Move the cursor to stretch or shrink the flexible border until it indicates the new window size you want.
4. Press **<MARK>**. The old window will shrink or expand according to the position of the flexible border.

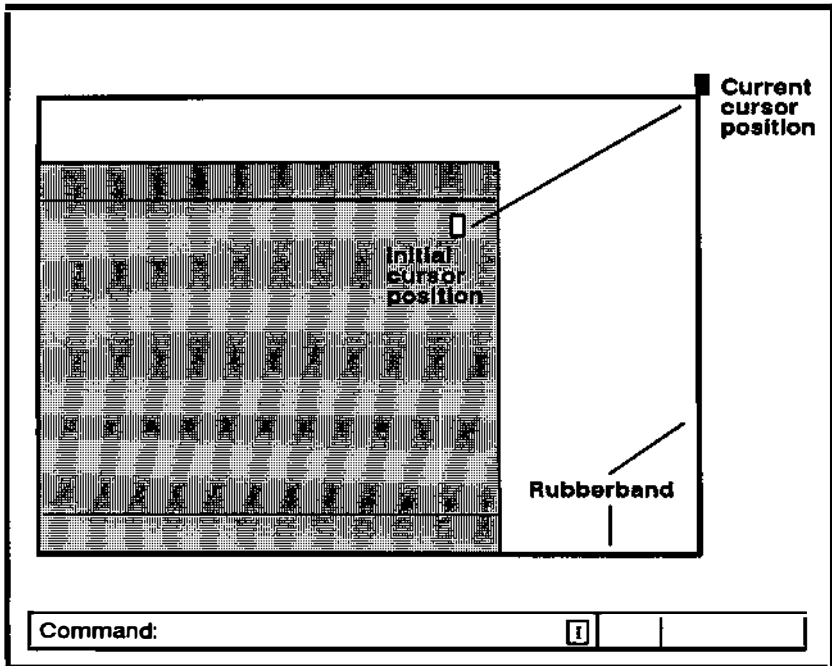


Figure 2-5. Changing a Window's Size

If you want to move just one edge, move the cursor only in the direction perpendicular to that edge. Moving the cursor vertically and horizontally causes a corner to move.

If, after the flexible border appears, you decide not to change the window size, type **CTRL/X**.

You can also change window size using a mouse. To do so, follow these steps:

1. Place the cursor near the edge or corner you wish to move.
2. Press the left-most mouse key and hold it down. The DM executes the **GROW** command, and the flexible border appears.
3. While you hold the left key down, move the cursor to indicate the new window size.
4. Now release the key. The DM executes the **MARK** command, and the window changes size.

Moving a Window

If you want to change the position of a window without changing its size, use **<MOVE>** (on low-profile keyboards) or the **CTRL/W** sequence (on 880 keyboards). Use these steps to move one of the windows on your screen:

1. Place the cursor on an edge or corner of the window.
2. Press **<MOVE>** or **CTRL/W**. A movable copy of the window border appears.
3. Move the cursor to position this border where you want the window to appear.
4. Press **<MARK>**. The window then moves to its new location.

If, after the movable border appears, you decide not to move the window, type **CTRL/X**.

Stopping a Process

When you've finished experimenting, you can stop the Shell processes you created and close the associated pads and windows. To stop a Shell process and close its windows and pads, follow these steps:

1. Move the cursor into the process input window and press CTRL/Z. CTRL/Z stops the process, deletes the input window, and closes all pads associated with the process. The system displays:

```
$ ***EOF***  
***Pad Closed***
```

The process input window disappears from your screen, but the window to the transcript pad remains.

2. To remove the window to the transcript pad from your screen, keep the cursor within the window and press CTRL/N.

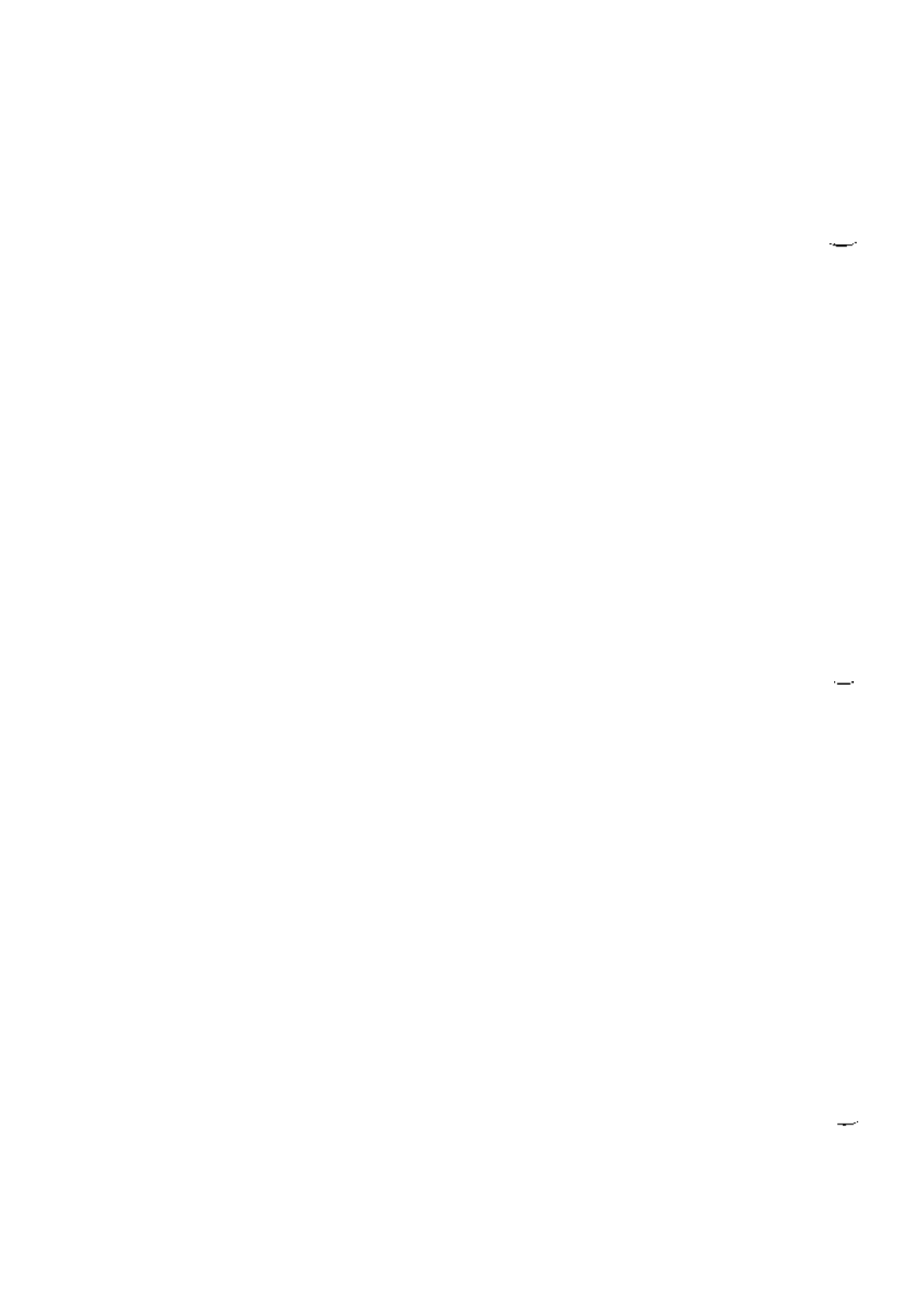
The DM deletes the transcript pad when you close the window or log off. If you want to save the contents of the transcript pad, enter the DM command PN (PAD_NAME) before you press CTRL/Z. For example, move the cursor onto the transcript pad, then press <CMD> and type

Command: PN Save_File <RET>

The PN command makes the temporary transcript pad a permanent file named *Save_File*. The DM continues to add transcript pad output to *Save_File* until you stop the process and close the windows. Refer to Chapter 4 in the *DOMAIN System User's Guide* for more information about the PN command.

Summary

In this chapter, you learned about DM commands and keys that control windows, pads, and processes. You now know how to move a pad under a window, shuffle windows, and change their size and position on the screen. You can also create and stop Shell processes. In Chapter 3, you'll learn how the system organizes information.



How the System Organizes Information

The system organizes related information in files. We supply certain files with system software. You can create other files, and determine their contents. A file might contain a memo, a program, or a picture — anything you like.

The system organizes related files in directories. You can also create directories, and decide which files to store in them. We will explain how to create files and directories later in this chapter. The system keeps track of its files and directories by arranging them in a hierarchical structure called the naming tree.

This chapter introduces the basic concepts you'll need to understand and use the naming tree. Figure 3-1 shows a sample naming tree. (Chapter 1 in the *DOMAIN System User's Guide* provides additional information about the naming tree. Appendix A in the *DOMAIN*

System User's Guide shows how the system organizes the system software that we supply with your node.)

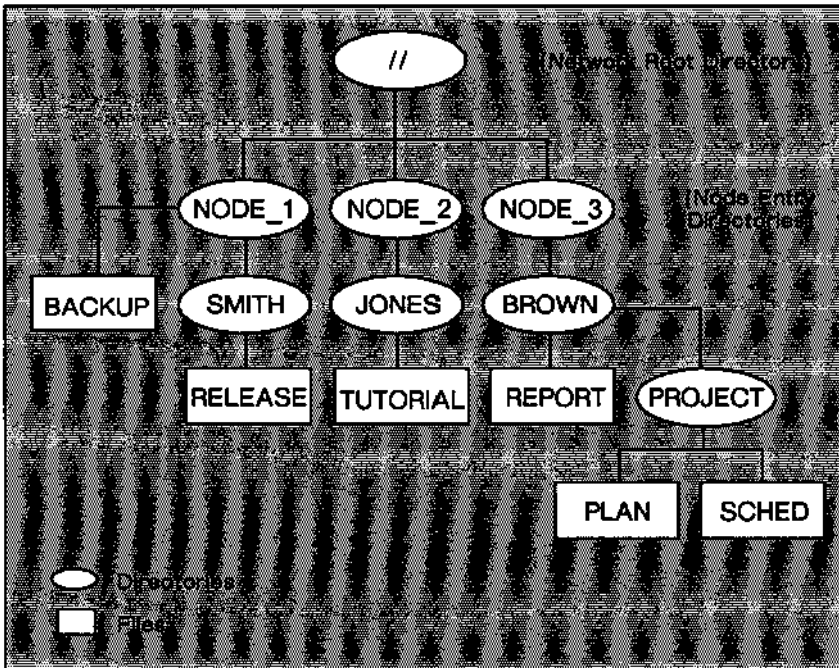


Figure 3-1. A Sample Naming Tree

The naming tree includes every file and directory in the network (not just your node). Each directory in the naming tree appears above the files and subdirectories that it contains. Naming tree components are called **objects**.

In addition to files and directories, the naming tree includes a third type of object — **links**. A link contains the name of another network object. When the system uses a link, it replaces the link name with the object name the link contains. We will explain how to create and use links later in this chapter. For now, just think of links as a special object type that enables you to take a detour from one part of the naming tree to another.

The object names that we use in the examples in this chapter are based on the sample naming tree in Figure 3-1. The object names in your naming tree will be different.

Using Pathnames

In order to use an object in the naming tree, we must be able to locate it. A **pathname** describes the path the operating system must take to get from some starting point in the naming tree to a destination object. For example, consider the file *TUTORIAL* in Figure 3-1. One pathname we could use to locate *TUTORIAL* looks like this:

//NODE_2/JONES/TUTORIAL

A pathname begins with the starting point's name, and includes every directory name between the starting point and the destination object. A pathname ends with the destination object's name. Slashes separate names within a pathname. An individual name (between slashes) may not exceed 32 characters, and the entire pathname may not exceed 256 characters, including the slashes.

In the pathname **//NODE_2/JONES/TUTORIAL**, the starting point is the directory at the top of the naming tree, called the **network root directory**. (Double slashes (**//**) refer to the network root directory.) The destination object is the file *TUTORIAL*. The directories between the starting point (**//**) and the destination object (*TUTORIAL*) are *NODE_2* and *JONES*. Figure 3-2 highlights the path the system follows in its search for *TUTORIAL*.

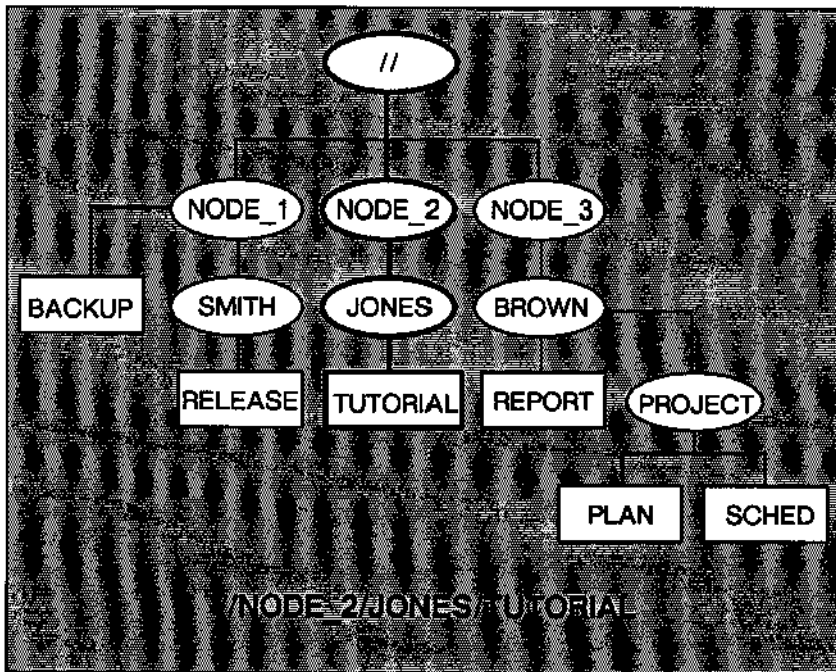


Figure 3-2. Sample Pathname

As you'll see in the following sections, not all pathnames begin with the network root directory. The system provides shorter ways to specify a pathname, depending on your current location in the naming tree.

Let's examine the parts of the naming tree more closely.

Where Am I?

Perhaps Figure 3-1 would be more meaningful if it included a "You Are Here" sign. Let's work our way from the top of the naming tree, the network root directory, to the directory you're currently using — your working directory.

The Network Root Directory (//)

The network's top-level directory, the root directory (//), is a list of directory names. It contains the name of each network node's top directory. To display the contents of your root directory, type the Shell command **LD (LIST_DIRECTORY)** and the pathname // as follows:

```
$ LD // <RET>
```

Use the Shell command **LD** to list the contents of any directory in your network. You can refer to any object in your network by specifying a pathname that begins with the root directory (//). (You must have the appropriate access rights to the objects you specify. Chapter 8 in the *DOMAIN System User's Guide* explains access rights.)

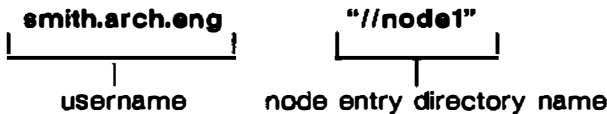
If you list the contents of the root directory in our sample naming tree (Figure 3-1), you will see *NODE_1*, *NODE_2*, and *NODE_3* listed on your screen. Thus, the sample naming tree represents a network of three nodes, called *NODE_1*, *NODE_2*, and *NODE_3*.

Your Node Entry Directory (/)

The node entry directory is the top directory on each node and is a subdirectory of the network root directory. In our sample naming tree, the node entry directories are *//NODE_1*, *//NODE_2*, and *//NODE_3*. To display your username and your node's entry directory name, type the **LUSR (LIST_USER)** command as follows:

```
$ LUSR -me <RET>
```

Here's an example of the information that the *LUSR -me* command displays:



The **LUSR** command you entered included the *-me* command option. Most commands allow you to modify command execution by specifying one or more options. An option follows a command name, and consists of a hyphen and one or more letters. One or more spaces separate a command from an option. No space appears between the hyphen and the option name. (For more information about Shell command options, see Chapter 5.)

The Shell command `LUSR` with the `-me` option displays your username and the node's entry directory name. (If you're using a diskless node, your node uses the entry directory of a disked partner node. Refer to Chapter 1 in the *DOMAIN System User's Guide* for more information about diskless nodes.) If we omit the command option `-me`, `LUSR` lists all network users and their respective node entry directories.

You can use a single slash (`/`) as a shorthand way of referring to your node's entry directory. For example, the following command lists the contents of your node entry directory:

```
$ LD / <RET>
```

When you begin a pathname with a single slash (`/`), the system begins its search in the node entry directory of the node that is physically connected to your display unit. (If your node is diskless, the system begins its search in the entry directory of your node's disked partner.)

You could also list the contents of your node entry directory by typing the `LD` command and a pathname that includes the root directory (`//`) and your node entry directory name. For example, if you were working on `NODE_1` in the sample naming tree shown in Figure 3-1, the command

```
$ LD //NODE_1
```

will list `BACKUP` and `SMITH`.

To refer to the entry directory on *another* node, use a pathname that starts with the root directory (`//`) and includes the name of the other node's entry directory.

Your Home Directory

The directory where the system locates you when you log in is called your **home directory**. Your user account contains your home directory name. When you log in, the system places you in the home directory specified in your user account.

You can change your home directory anytime you log in. This makes it possible for you to choose the directory where you want to begin working. To change your home directory, enter your username (and project and organization names, if required), enter a space, type `-h`, and press `<RETURN>`. For example:

```
Please log In: L USERNAME -h <RET>
```

After you supply your current password, the system prompts you to enter your new home directory's name. It changes the home

directory specification in your user account, and places you in the new home directory.

Your Working Directory

Your **working directory** is the directory in which you are currently located. When you log in, the system sets the working directory to the home directory specified in your user account.

To display the name of the directory you're using (your working directory), enter the **WD (WORKING_DIRECTORY)** Shell command:

```
$ WD <RET>
```

The working directory influences where a process creates or searches for object names (the names of files, directories, or links). For example, let's say your working directory is *//NODE_1/SMITH* in our sample naming tree. If you instruct the process to create a directory named *NEWDIR* using the Shell command **CRD (CREATE_DIRECTORY)**, the Shell process creates the new directory in *//NODE_1/SMITH* (the current working directory). For example:

```
$ CRD newdir <RET>
```

The new directory, *//NODE_1/SMITH/NEWDIR*, becomes a subdirectory of the current working directory, *//NODE_1/SMITH*. The new directory appears beneath the current working directory in the naming tree. (If you enter the **LD** command, *NEWDIR* appears in the list of the working directory contents.) Figure 3-3 illustrates the location of *NEWDIR* in the naming tree.

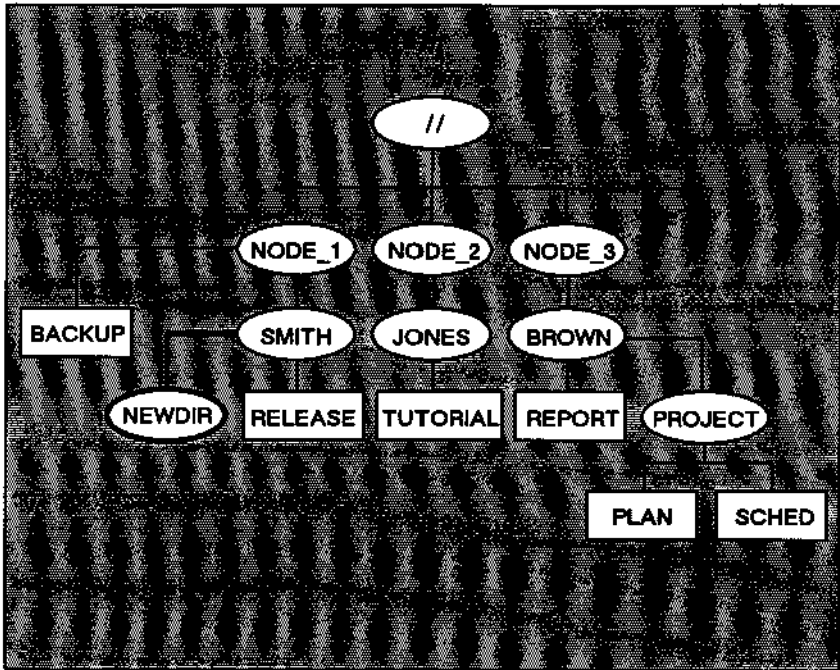


Figure 3-3. A New Directory In the Naming Tree

You can change the working directory to another directory using the WD command. Type

```
$ WD new_directory <RET>
```

where *new_directory* is the name of the new working directory.

Parent Directories

A **parent directory** is the directory above the current working directory. For example, if your current working directory is *//NODE_1/SMITH/NEWDIR*, the parent directory is *//NODE_1/SMITH*. To display the parent directory's name and contents, type

```
$ LD \ <RET>
```

The backslash character (\) refers to the directory one level above the current working directory. You can use adjacent backslashes to cause the system to back up more than one level above the current working directory. For example, let's say your current working directory is *//NODE_1/SMITH/NEWDIR*, and you want to change the working directory to *//NODE_1*. You would type

```
$ WD \ \ <RET>
```

Your Naming Directory

A naming directory is a shorthand method to refer to a particular directory in the naming tree. After you specify a directory as the naming directory, you can use the shorthand symbol "~" (tilde) in place of the pathname of the naming directory.

To display your current naming directory, enter the Shell command **ND (NAMING_DIRECTORY)**.

```
$ ND <RET>
```

The system uses your home directory as the initial naming directory for each process. To change the naming directory, type the **ND** command followed by the pathname of the new naming directory. For example:

```
$ ND //NODE_3/BROWN/PROJECT <RET>
```

specifies *//NODE_3/BROWN/PROJECT* as the new naming directory. You could then use the shorthand symbol "~" in place of *//NODE_3/BROWN/PROJECT*. For example, the pathname *~SCHEDULE* refers to the file

```
//NODE_3/BROWN/PROJECT/SCHEDULE
```

Pathname Symbols — A Review

As you have seen in the previous sections, you can refer to any object in your network by using a pathname that begins with the root directory (*//*). The system also provides shorthand symbols that you can use in place of longer pathnames. Table 3-1 summarizes pathname starting point symbols.

Table 3-1. Pathname Starting Point Symbols

If your pathname starts with this symbol:	The system begins the name search in this directory:
//	Network root directory
/	Node entry directory
No symbol or .	Working directory
\	Parent directory
~	Naming directory

Figure 3-4 illustrates how you can use different symbols to refer to the same directory in the naming tree.

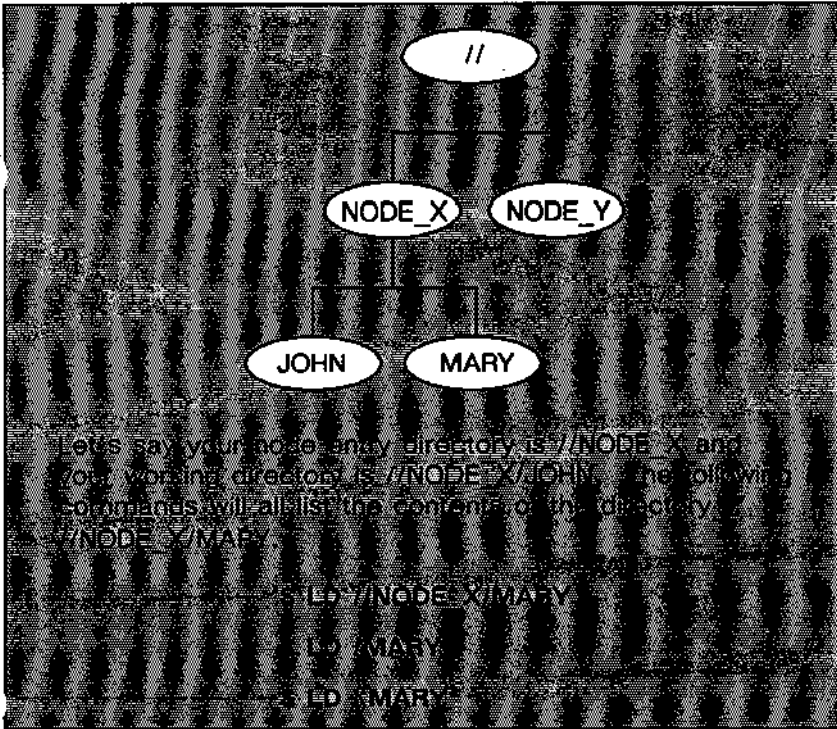


Figure 3-4. Pathnames Starting with //, /, and \

Remember that the examples in Figure 3-4 assume that the node entry directory is `//NODE_X` and that the working directory is `//NODE_X/JOHN`. Each of the pathnames in Figure 3-4 searches for a destination object named `MARY`. The object's full pathname is `//NODE_X/MARY`. The pathname `/MARY` instructs the system to look for `MARY` in the node's entry directory (`/`). The pathname `\MARY` instructs the system to move up to `//NODE_X` and then look for an object named `MARY`.

Using Links

As you use the system, you may find that you frequently access objects with very long pathnames. Instead of typing the long pathname each time you want to use an object, you can create a special object type called a link.

A link is a shorthand name for a pathname. It contains the pathname of another object. When you use a link name as a

pathname or as part of a pathname, the system substitutes the name inside the link (the resolution name) for the link name.

For example, let's say you work on *NODE_1* in our sample naming tree (Figure 3-1), and you often use the file *SCHED*. To access *SCHED*, you could type the following pathname each time you want to use the file:

```
//NODE_3/BROWN/PROJECT/SCHED
```

However, it would be much easier to create a link containing this pathname. Then you could use the shorter link name whenever you want to use *SCHED*. To create a link, use the Shell command **CRL (CREATE LINK)**. To create a link called *MYLINK* in the current working directory, type

```
$ CRL MYLINK //NODE_3/BROWN/PROJECT/SCHED <RET>
```

Now you can refer to *//NODE_3/BROWN/PROJECT/SCHED* with the name *MYLINK*. The Shell substitutes the resolution name (*//NODE_3/BROWN/PROJECT/SCHED*) for the link name (*MYLINK*).

If you keep links in the naming directory, you can use them regardless of your current working directory. To use a link in your naming directory, simply precede the link name with the naming directory symbol (-).

Refer to Chapter 7 in the *DOMAIN System User's Guide* for a more detailed explanation of links.

Summary

This chapter described how the system organizes files, directories, and links in the naming tree. You learned how to use pathnames to get from one part of the naming tree to another, and how to create new objects in the naming tree. Chapter 4 explains how to use the DM to create, edit, and read text files. It also describes how to use Shell commands to print, copy, and delete files.

Using Files

This chapter describes how to create, edit, and read files using DM function keys, control key sequences, and commands. In a sample editing session, you'll create a new file, enter text, and modify the text. This chapter also explains the Shell commands that print, copy, and delete files.

Opening a Window to a File

The function keys <EDIT> and <READ> instruct the DM to open a window to a file. Use <EDIT> to create a new file or to change the contents of an existing one. Use <READ> to view an existing file. You cannot modify text displayed in a read window, but you can change a read window into an edit window by pressing CTRL/M. (You must have the appropriate access rights to the file you specify. Chapter 8 in the *DOMAIN System User's Guide* explains access rights.)

Entering Filenames

After you press <EDIT> or <READ>, the DM prompts you to specify the name of the file you wish to use. When you press <EDIT>, you can specify an existing filename, or create a new file by responding with a filename that does not already exist. When you press <READ>, you must specify a filename that already exists.

A new name can consist of 1 to 32 alphanumeric characters, periods (.), underscores (_), or dollar signs (\$). All names must begin with a letter or a dollar sign. No name can contain spaces.

NOTE: Don't create text files with the same names as commands, or the Shell will attempt to execute the text file as if it were a command.

Editing a File

After you press <EDIT>, the cursor moves into the DM input window, and the DM displays the prompt

Edit file:

You can enter any new or existing filename. If you enter the name of an existing file, the DM opens the file for editing, and it also copies the original version of the file for backup purposes. This "backup" copy of the file appears in your directory with the original filename plus the .BAK suffix.

We've supplied a text file, called *Sample_Edit*, that you can use to learn about the DM's editing functions. To open *Sample_Edit*, type the following pathname next to the "Edit file:" prompt:

Edit file: /Domain_Examples/Getting_Started/Sample_Edit <RET>

When you press <RETURN>, the DM opens an edit window to the file *Sample_Edit* as shown in Figure 4-1. Notice that the pathname appears in the window legend.

We've included a few errors (highlighted in Figure 4-1) that we'll use to demonstrate some editing functions.

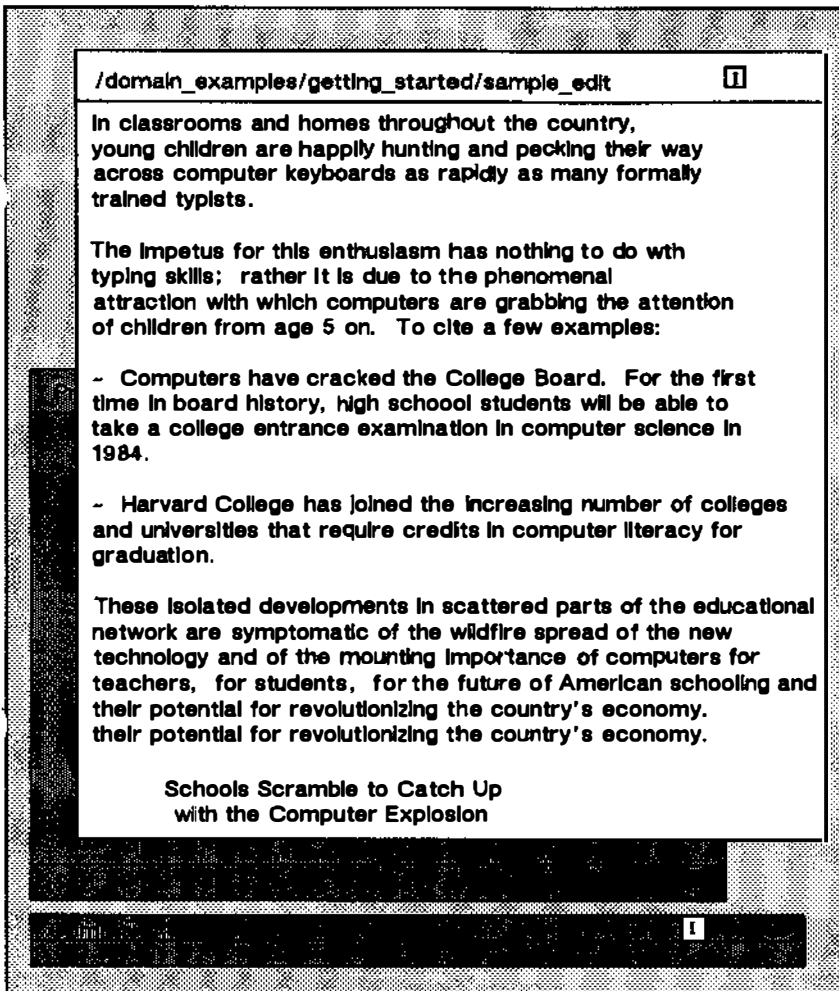


Figure 4-1. A Sample Editing Session

Correcting Errors

You can move the cursor to any text position using the arrow keys, the touchpad, or the mouse. As you know from Chapter 1, once you've positioned the cursor, you can use the following keys to correct errors:

- BACKSPACE
- INS (or INS MODE)
- CHAR DEL
- LINE DEL

Let's start by correcting the errors highlighted in Figure 4-1. We omitted the letter *i* from the word *with* in the first line of the second paragraph. Position the cursor on the letter *t* and type *i*. Because the edit window is in insert mode, the letters to the right of the cursor move right when you insert a new character. By default, insert mode is on when you begin an editing session, and the letter I appears in the edit pad's window legend.

Press <INS> (on low-profile keyboards) or <INS MODE> (on 880 keyboards) to turn insert mode off. Notice that the letter I disappears from the window legend. Now, any new characters that you type will overstrike (replace) existing characters. Let's try it. Move the cursor to the *s* in *schooling* in the last paragraph. Type *education*. To turn insert mode back on, press <INS> or <INS MODE> again.

Move the cursor to an *o* in *school* in the second line of the third paragraph. To delete an extra *o*, press <CHAR DEL>.

To delete the extra line in the last paragraph, move the cursor onto the line and press <LINE DEL>.

Defining a Range of Text

Before you can move, copy, or substitute text, you must define the range (or block) of text you want the DM to operate on.

To define a range of text, place the cursor at the beginning of the range and press <MARK>. Then move the cursor to the end of the range. As you move the cursor, the DM highlights the text, showing you exactly what the range is. (Please note that the character under the cursor at the end of the range is NOT included within the range.) After you position the cursor at the end of the range, you either press a DM function key or type a command in the DM input window.

If you do not define a range, the default range for cut, copy, and substitute commands begins at the current cursor position and ends at the end of the line.

Cutting and Pasting (Moving) Text

The DM allows you to delete portions of text from a file, and to paste them in a different location in the same file, or in another file. You must define a range of text before you execute the DM cut command.

To accomplish a cut or copy (see the next section) operation, the DM uses a paste buffer. A paste buffer is a temporary file that holds text that you have cut or copied so that it can be pasted in elsewhere.

Let's try a cut and paste operation. We'll move *Schools Scramble To Catch Up with the Computer Explosion* in Figure 4-1 from the bottom to the top of the file. Follow these steps:

1. To define the range for the operation, move the cursor to the first character position in the line before *Schools Scramble To Catch Up*, and press <MARK>. Now move the cursor to the bottom of the file. As you move the cursor, notice the highlighting. This lets you see exactly what text falls within the range.
2. Press <CUT> (on low-profile keyboards) or CTRL/E (on 880 keyboards). The DM deletes the text in the range you specified and writes it into the paste buffer.

NOTE: The DM writes all deleted text into this buffer; however, it saves only the text deleted during the last DM operation. Therefore, don't delete anything else until you reinsert the paste buffer contents. Otherwise, you will lose the text you are attempting to move.

3. To place the contents of the paste buffer at the top of the file, press CTRL/T to move the cursor to the top of the file, and then press <PASTE> (on low-profile keyboards) or CTRL/O (on 880 keyboards).

In this exercise, you used the DM's default paste buffer to hold the text you cut. You can create your own paste buffers (up to 100), each containing different blocks of text. To learn how to create and use paste buffers, see the cut, copy, and paste commands in Chapter 5 in the *DOMAIN System User's Guide*.

To delete a range of text without pasting it in elsewhere, define the range and then press <CUT> or CTRL/E.

Copying and Pasting Text

The DM allows you to copy text into the paste buffer without deleting it from your file. The steps for copying text are similar to the steps for moving text. The difference is that you use <COPY> (on low-profile keyboards) or CTRL/C (on 880 keyboards) instead of <CUT> or CTRL/E. You can copy text to a different location within a file or into another file.

In the next exercise, we'll copy a range of text into a new file. To do so, follow these steps:

1. Define any range of text in the file *Sample_Edit* in Figure 4-1. Move the cursor to the beginning of the range you want to copy, and press <MARK>. Then move the cursor to the end of the range. The DM highlights the text as you move the cursor.
2. Press <COPY> or CTRL/C. The DM copies the range of text into the paste buffer. The original text remains undisturbed.
3. To create the new file, press <EDIT>. Next to the "Edit file:" prompt in the DM input window, type

Edit file: *Copied_Text* <RET>

The DM opens an edit window and pad to the new file *Copied_Text*. The cursor is in the upper-left corner of the new file.

4. Press <PASTE> or CTRL/O. The text you copied to the paste buffer appears in the file *Copied_Text*.

To close *Copied_Text* and save its contents, keep the cursor in the window and press CTRL/Y.

Searching for Text

The DM search function makes it easy for you to locate a particular text string in a file. In the DM input window, you type the string between slashes (/string/) or backslashes (\string\). A string enclosed in slashes instructs the DM to search forward from the current cursor position, and a string enclosed in backslashes instructs the DM to search backward from the current cursor position. For example, to search for the string *computer* in *Sample_Edit*, follow these steps:

1. Press CTRL/T to move the cursor to the top of the file.
2. Press <CMD> to move the cursor to the DM input window.
3. Enter the search command as follows:

Command: /computer/ <RET>

The DM then moves the cursor to the first occurrence of *computer* (in the first paragraph).

The CTRL/R and CTRL/U sequences repeat the last search forward or backward, respectively. For example, if you press CTRL/R now, the DM will search forward for the next occurrence of *computer* (in the second paragraph). If you then press CTRL/U, the DM will move the cursor back to the first occurrence of *computer* in the first paragraph.

Because the DM saves your most recent search command, you can repeat a search even if you've entered other (non-search) commands since you entered the search command.

Searches are case-insensitive by default. This means that /mary/ will locate *mary*, *MARY*, *Mary*, and even *mARY*. You can use the DM command SC to perform case-sensitive searches. Chapter 5 in the *DOMAIN System User's Guide* explains the SC command.

Canceling a Copy, Cut, or Search

To cancel a copy, cut, or search operation, use the CTRL/X sequence. For copy and cut operations, the DM removes the highlighting that indicates the range of text you are defining. For a search, the DM displays the message "Search aborted".

Substituting Text

The DM also provides an easy-to-use search and substitute command, S (SUBSTITUTE). You can use it to search a file or part of a file for a text string, and to replace the string with a new string. First, you must define the range of text you wish to search and substitute. Then, you can use the DM search and substitute command to perform the replacement.

For example, to replace every tilde (~) in *Sample_Edit* with the letter *o*, you must:

1. Define the range of text you wish to search and substitute. The range for this search/substitute operation is from the top to the bottom of the file. To communicate this information to the DM:

- A. Move the cursor to the top of the file (CTRL/T does this) and press <MARK>.

- B. Next move the cursor to the bottom of the file (CTRL/B does this) and press <CMD>.

Steps A and B tell the DM that the range for the next DM command begins at the top of the file and ends at the bottom.

2. Next, enter the following search and substitute command:

Command: *S/-/o/* <RET>

The DM replaces every tilde (-) in the file with the letter *o*. If you had not marked a range, the DM would have searched only the line that was to the right of the cursor when you pressed <CMD> in Step 1B.

Undoing Previous Commands

The UNDO function reverses the effect of the most recent DM command you entered. UNDO works by keeping a history of activities in the DM input pad and edit pads in reverse chronological order. To undo the previous DM command, press <UNDO> (on low-profile keyboards) or press <CMD> and type UNDO (on 880 keyboards). Successive use of the UNDO function key or command will undo DM commands further back in history.

The history of DM activities consists of circular lists (one for input pad activities and one for each edit pad). When the lists are full, they eliminate the oldest entries to make room for new ones.

Please note that UNDO works for DM operations only; it will not reverse the effect of Shell commands once those commands have been executed.

Closing the EDIT Window

When you've finished editing the file *Sample_Edit*, leave the cursor in the window and press <EXIT> (on low-profile keyboards) or

CTRL/Y (on 880 keyboards). This closes the edit window and saves *Sample_Edit* with all of the changes you made.

To discard changes you've made during an editing session, use **<ABORT>** (on low-profile keyboards) or **CTRL/N** (on 880 keyboards). The DM closes the edit pad and window without saving the changes you made during the session.

If you've created a new file during the editing session, **<ABORT>** or **CTRL/N** ends the session without creating the file. If you've edited an existing file during the session, **<ABORT>** or **CTRL/N** does not make any of the changes you've entered, and saves the old version of the file. When you press **<ABORT>** or **CTRL/N**, the DM displays the message

File modified. OK to quit?

Enter **Y** or **YES** to discard the changes you've made and close the edit pad and window. Enter **N** or **NO** to resume editing.

Reading a File

As you saw in Chapter 2, when you press **<READ>** the cursor moves into the DM input window and the DM displays the prompt

Read file:

You must specify a file that already exists. If you specify a filename that does not exist, the DM issues the error message

(CV) filename – File not found

CV (CREATE_VIEW) is the DM command that **<READ>** invokes.

You can supply the name of a file in the current working directory, or use a pathname to refer to a file anywhere in the network.

Let's read the file you used in the sample editing session. Press **<READ>** and type the file's pathname as follows:

Read file: /Domain_Examples/Getting_Started/Sample_Edit <RET>

The DM displays the file *Sample_Edit*. Notice the letter **R** (for Read) displayed in the window legend. Because *Sample_Edit* is now displayed in a read-only window, you cannot modify its contents. If you attempt to do so, the DM displays this message:

Text is read-only

To change the read-only window to an edit window, place the cursor in the window and press **CTRL/M**.

The right-most mouse key also opens files for reading. To use this key, you must first display the name of the file you wish to read on your screen (the Shell command **LD** does this), and then point to it with the cursor. When you press the right-most mouse key, the DM displays the contents of the filename nearest the cursor.

Closing the READ Window

When you finish using *Sample_Edit*, execute the following steps to close the file's window and pad:

1. Move the cursor into the file's window.
2. Press **<ABORT>** (on low-profile keyboards) or **CTRL/N** (on 880 keyboards).

Copying a File

The Shell command **CPF (COPY_FILE)** copies the contents of a file and stores it in another file. You specify the file to be copied and the new file. For example, to make a copy of *Sample_Edit* and store it in a new file called *Copied_File* in your current working directory, enter the following command:

```
$ CPF /Domain_Examples/Getting_Started/Sample_Edit Copied_File <RET>
```

The DM copies *Sample_Edit* to *Copied_File* in the current working directory. If you enter the Shell command **LD**, you'll see *Copied_File* listed in the contents of your working directory.

Printing a File

In order to print a file, your network must support a printing device, and it must execute a process called the Print Server. When the Print Server is running, you can print any file using the print command or the print menu.

The next two sections introduce the print command and the print menu. For more information about printing files, see Chapter 7 in the *DOMAIN System User's Guide* and the *DOMAIN System Command Reference*.

Printing a File Using the Print Command

To print a file using the print command, enter the Shell command **PRF (PRINT_FILE)** and the pathname of the file. For example, the following command prints the file *Copied_File* on the line printer:

```
$ PRF Copied_File <RET>
```

To print the file on another printer, use the **-pr** option to specify the printer name. For example:

```
$ PRF Copied_File -pr spin <RET>
```

prints the same file on a printer called *spin*. The system displays the following message:

"//NODE/OWNER/COPIED_FILE" queued for printing

The PRF command copies the file *Copied_File* to a temporary file. This file resides on the node that controls your printer. Because PRF has submitted a copy of your file to the printer, you can edit the original version without changing the copy that is printing.

Ask your system administrator for the names of the printing devices at your installation.

Printing a File Using the Print Menu

In addition to the print command, you can print a file using the print menu shown in Figure 4-2. This menu allows you to specify the same arguments and options as the print command by typing information and making selections from the menu.

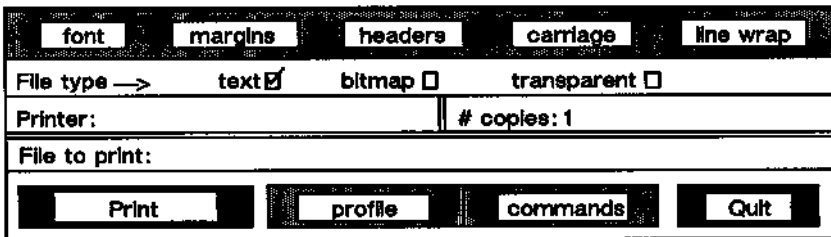


Figure 4-2. The Print Menu

To display the print menu on your screen, enter the Shell command **PRFD (PRINT_FILE_DIALOG)** as follows:

```
$ PRFD <RET>
```

The **PRFD** command displays the print menu in a special window at the top of the Shell process transcript pad. An arrow cursor appears in the upper left corner of the menu.

To use the print menu to print the file *Copied_File*, move the arrow cursor to the space following **File to print:** A small typing cursor appears on the line. Type *Copied_File* as follows:

```
File to print: Copied_File ▲
```

To print *Copied_File* on a printer other than the line printer, move the arrow cursor to the space following **Printer:** The typing cursor appears on the line. Type the name of the printer you want to use. For example:

```
Printer: spin ▲ # copies: 1
```

When you're satisfied with the information you've supplied on the print menu, move the arrow cursor to the box labeled **Print**. Press either the space bar or function key **<F1>**. If you're using a mouse, press the left mouse key. The following messages appear on the transcript pad:

```
Queuing data file "COPIED_FILE".  
//NODE/OWNER/COPIED_FILE queued for printing.  
File "/SYS/PRINT/QUEUE/SPIN.OWNER.COPIED_FILE"  
queued for printing.
```

This message means that the **PRFD** command has copied *Copied_File* to a temporary file that is called *SPIN.OWNER.COPIED_FILE*. This file resides on the node that controls your printer in the directory */SYS/PRINT/QUEUE*.

The print menu remains on your screen, allowing you to print additional files. When you're ready to exit from the print menu, move the arrow cursor to the box labeled **Quit**. Press the space bar, **<F1>**, or if you're using a mouse, the left mouse key. The menu disappears from your screen, and the cursor reappears next to the Shell prompt.

Deleting a File

The Shell command **DLF (DELETE FILE)** deletes files from directories. To delete a file, enter the **DLF** command and a filename. For example:

```
$ DLF Copied_File <RET>
```

deletes the file *Copied_File* from your current working directory. To delete a file anywhere else in the network, supply a pathname.

Summary

In this chapter, you learned how to use DM function keys and commands to create, edit, and read files. You also learned how to print, copy, and delete files using Shell commands. Chapter 5 contains more information about the Shell and Shell commands.

—

—

—

Using the Shell

As you saw in Chapter 1, the Shell allows you to perform traditional computing activities, such as printing files and compiling and running programs. When you enter a Shell command, the Shell invokes the utility program to perform the task you request. For example, when you entered the `DLF` command at the end of Chapter 4, the Shell invoked the utility program that deletes files.

This chapter explains additional information about the Shell and its commands. You'll learn:

- The parts of a Shell command line
- How the Shell locates and interprets the commands you enter
- To use symbols that have special meaning to the Shell
- To create your own Shell commands, called Shell scripts

Command Format

The simplest Shell command consists of a command name. For example:

```
$ LD <RET>
```

This **LD (LIST_DIRECTORY)** command lists the contents of your current working directory.

Using Command Arguments

Most Shell commands accept one or more **command arguments**. A command argument specifies the object (file, directory, or link) upon which the command will act. For example:

```
$ LD Your_Directory <RET>
```

In this example, *Your_Directory* is the argument. The LD command lists the contents of *Your_Directory*. You must use one or more spaces to separate a command from its arguments, and to separate arguments from each other.

Using Command Options

In addition to command arguments, most Shell commands accept **command options**. Command options modify the action of commands. They can generally appear either before or after command arguments. Precede each option with a hyphen. Do not include any space between the hyphen and the option. For example:

```
$ LD -c <RET>
```

The LD command displays the contents of the current working directory; the **-c** option instructs LD to list the contents in a single column.

You can use the following options with any Shell command:

- HELP** Displays detailed information about the command and its format.
- USAGE** Displays a brief description of the command and its format.

-VERSION Displays the software version number.

Enter the LD command with one of the options in the preceding list. You'll see that the Shell displays the information on its transcript pad.

For information about options that you can use with a specific command, check the system Help files (type HELP and the command name), or refer to the *DOMAIN System Command Reference*.

Entering Multiple Commands on a Line

To enter more than one command on a single line, separate the commands with a semicolon (;). For example:

```
$ LD;DATE <RET>
```

This command lists the contents of the current working directory, then displays the date. Here's another example that you'll probably use often:

```
$ WD //Your_Node/Your_Dir;LD <RET>
```

The WD (**WORKING DIRECTORY**) command changes the working directory to *//Your_Node/Your_Dir*, and the LD command lists the contents of this new working directory.

Entering Long Command Lines

A <RETURN> character usually delimits a command line. To enter a command line that is longer than the current input window, type @ before pressing <RETURN>. This causes the Shell to ignore the <RETURN> character, and continue the command on the next line of the input pad. For example:

```
$ WD //Node/Owner@ <RET>  
$_Directory <RET>
```

This command sets the current working directory to *//Node/Owner/Directory*. The @ symbol makes the Shell ignore the <RETURN> character it precedes and places an underscore () character in the first position on line two. The underscore indicates that the command begins on the previous line; the Shell does not interpret this character when it reads your command.

The @ symbol suppresses the meaning of the special character it precedes. In the previous example, the @ symbol preceded a <RETURN> character and canceled the <RETURN> character's usual effect. The <RETURN> character usually delimits a command line. In this case, the @ symbol makes the Shell ignore the <RETURN> character.

Command Line Processing

Most commands are files. When you press <RETURN> to enter one of these commands, the Shell searches for a file with the same name as the command you specified. If the Shell finds such a file, it loads and executes it as a program. Next, the Shell passes any arguments and options on your command line to this program. After it executes the program, the Shell returns its "\$" prompt. This prompt means the Shell is ready for your next command. For example, when you press <RETURN> to enter the following command:

```
$ LD My_Directory -c <RET>
```

the Shell searches for a file named *LD*. The Shell then loads and executes the program it finds in the file *LD*. Next, the Shell passes the argument *My_Directory* and the option *-c* to the program. After it executes the program and displays the contents of *My_Directory* in a single column, the Shell redisplay the "\$" prompt on its input pad. You can then enter another Shell command.

Command Search Rules

When you enter a command that is the name of a file, the Shell follows command search rules to determine which directories it searches to locate the command file. You can display the current search rules by using the Shell command *CSR* (*COMMAND_SEARCH_RULES*). For example:

```
$ CSR <RET>
```

A set of command search rules comes with your system software. If you're using these default search rules, the Shell displays these three directory names when you enter the *CSR* command:

```
. -COM /COM
```

This list of directory names means the Shell looks for command files in this order:

1. Your working directory (.)

2. Your personal command directory (-COM)
3. The system command directory (/COM)

You can also use the CSR command to change the current search rules (see Chapter 6 in the *DOMAIN System User's Guide*).

As soon as the Shell finds a filename that matches the command you specify, it attempts to execute the program. As we noted in Chapter 4, don't create text files with the same name as commands because the Shell attempts to execute the text file as if it were a command file.

The -COM directory is a subdirectory that you can create in your naming directory (-). When the Shell does not find the command file you specify in the working directory, it checks the -COM directory. Therefore, -COM is a good place to store Shell scripts (see the "Creating Shell Scripts" section later in this chapter) and other frequently-used programs. You are not required to create a -COM directory — no error occurs if it does not exist.

The /COM directory contains the command files supplied with your system. The Shell checks /COM if it does not find the command you specify in your working directory or in your personal command directory (-COM). For example, when you type

```
$ PRF Myfile <RET>
```

the Shell looks for the PRF command in your working directory first, then in your -COM directory. Finally, it finds the PRF command file in /COM and prints *Myfile* on the line printer. (This example assumes that you have not created an object named PRF in your working directory or in -COM.)

Using Wildcards

Wildcards are special characters that you can use to represent one or more pathnames. For example, the following command line contains the *?** wildcard:

```
$ LD ?*.BAK <RET>
```

The *?** wildcard matches every file that ends in *.BAK* (backup versions of text files). Therefore, this command line lists all the files that end in *.BAK* in your current working directory.

The next example lists all the objects (files, directories, and links) on *Node7*.

```
$ LD //Node7/... <RET>
```

The ... wildcard matches zero or more characters below the starting point in the naming tree. In this example the starting point is *//Node7*.

Most Shell commands that accept pathnames as arguments also accept wildcards. Wildcards are easy to use, but there are a lot of them. This section has shown two examples to introduce you to the concept. See Chapter 6 in the *DOMAIN System User's Guide* for complete information about wildcards.

Redirecting Input and Output

In the previous examples in this book, the Shell process read the commands you entered (input) from its input pad, and it wrote the responses to your commands (output) on its transcript pad. The next two sections explain a few examples of special characters that you can use to redirect input and output (I/O). For complete information about the special characters that redirect I/O, see Chapter 6 in the *DOMAIN System User's Guide*.

Writing Output to a File

Many Shell commands write output to the transcript pad. For example, the following **FMT** (**FORMAT TEXT**) command line, with only a filename as an argument, directs its output to the transcript pad:

```
$ FMT Myfile <RET>
```

Here, **FMT** formats the text in *Myfile* and displays the output (the formatted text) on the transcript pad.

To direct the output to a file instead of the transcript pad, use the *greater-than* (>) symbol and another filename. For example:

```
$ FMT Myfile >Myfile.fmt <RET>
```

FMT formats *Myfile* and writes the formatted text to *Myfile.fmt*. The Shell creates the output file (*Myfile.fmt*) if it does not already exist. Now that the formatted text is stored in a file, you can read it or prepare it for printing.

Reading Input from a File

To instruct a Shell command to read input from a file instead of the input pad, use the *less-than* (<) symbol and a filename. One example might be the TLC (TRANSLITERATE_CHARACTERS) command, which replaces single characters or a range of characters with other characters that you specify. TLC normally reads input from the transcript pad. In this section, you'll learn how to instruct TLC to read input from a file.

Before entering the TLC command, create a text file to use as input to the command. To do so, follow these steps:

1. Press <EDIT> to create the file.
2. In the DM input window, enter the filename as follows:

Edit file: Replace_Chars <RET>

3. When the DM creates the edit window and pad on your screen, type the following lines of text:

TLC will replace the number two with
the number three in the following lines:
2 years ago
2 days from now
2 hours earlier

4. Press CTRL/Y to close the edit window and pad.

Now instruct TLC to get its input from the file *Replace_Char*. Type the following command line next to the \$ prompt:

```
$ TLC 2 3 <Replace_Chars <RET>
```

The numbers 2 and 3 are arguments to the command. The number 2 is the character to be replaced, and the number 3 is the replacement character. <Replace_Chars instructs TLC to perform this replacement task on the text in the file *Replace_Chars*. TLC writes its output to the transcript pad, like this:

TLC will replace the number two with
the number three in the following lines:
3 years ago
3 days from now
3 hours earlier

If you omit the *less-than* (<) symbol and the input filename, TLC waits for you to type the text for the replacement task in the process input window.

To instruct TLC to write the output to a file instead of the transcript pad, use the *greater-than* (>) symbol and an output filename. For example:

```
$ TLC 2 3 <Replace_Chars >Replace_Chars.New <RET>
```

TLC replaces every occurrence of the number 2 in the file *Replace_Chars* with the number 3 and writes the output to the file *Replace_Char.New*.

Creating Shell Scripts

A **Shell script** is a file that contains a list of Shell commands. To create a script, use <EDIT> to create the file, type the Shell commands, one per line, then press CTRL/Y to close the file. To execute the script, you type the script's filename in the process input window next to the \$ prompt.

If you find yourself repeating a sequence of commands over and over again, create a Shell script containing the commands. You can then execute the entire command sequence with a single command. For example, to change your working directory, then list the contents of the new working directory with special options, enter these Shell commands:

```
$ WD My_Dir <RET>  
$ LD -st -dtm <RET>
```

The **WD** command sets the working directory to *My_Dir*. The **LD** command displays the name of each object in *My_Dir*. The *-st* option displays each object's system type (file, directory, or link), and the *-dtm* option displays the date and time you last modified each object.

If you create a Shell script containing the **WD** and **LD** command lines, you can execute both commands by simply entering the filename of the script. To create a script called *DIR*, follow these steps:

1. Press <EDIT> to create the file.
2. In the DM input window, enter the filename of the script as follows:

```
Edit file: DIR <RET>
```

3. When the DM creates the edit window and pad, type the Shell commands, one per line, like this:

```
WD My_Dir <RET>
LD -st -dtm
```

4. Press CTRL/Y to close the edit window and pad.

You can now execute both the WD and LD command lines by typing

```
$ DIR <RET>
```

Substituting Arguments

The Shell script *DIR* isn't very useful because it operates on a single directory (*My_Dir*) only. To create a more useful script, we need a way to substitute any directory name for *My_Dir*. The Shell provides a way to substitute arguments. To do so, edit the *DIR* script and change *My_Dir* to *^1*. Your results should look like this:

```
WD ^1
LD -st -dtm
```

After you press CTRL/Y to close the edit window and pad, instruct *DIR* to operate on any directory by supplying the directory name as an argument. For example:

```
$ DIR Any_Directory <RET>
```

The Shell substitutes the first command argument you specify (in this case *Any_Directory*) for *^1*, and then executes the script. *DIR* then sets the working directory to *Any_Directory* and lists the name, type, and date and time of the last modification for each object in *Any_Directory*.

The previous two sections provided basic information about Shell scripts. When you're ready to create more sophisticated scripts, read Chapter 9 in the *DOMAIN System User's Guide*.

Summary

In this chapter, you learned more about the Shell and its commands. You read about how the Shell processes your commands and how it locates the command files you request. We also introduced you to some of the more powerful features of the

Shell, namely wildcards, symbols that redirect input and output, and Shell scripts.

Now that you're familiar with the Shell and the DM, read the *DOMAIN System User's Guide* to learn how to perform specific tasks. See the *DOMAIN System Command Reference* for detailed command descriptions.

GLOSSARY

- Access Rights** An *access control list (ACL)* contains a set of access rights. These rights list the people who can use each object in the network, and specify how each person can use the object.
- Alarm Window** The Display Manager alarm window appears near the bottom of your screen. It displays a small pair of bells when a process displays a message in an output window that is hidden by an overlapping window.
- Argument** See *Command Argument*.
- Command** An instruction you give a program.
- Command Argument** An object name you include in a command line to specify the object upon which the command will act. Use one or more spaces to separate commands from arguments, and to separate arguments from each other. To include blank spaces within an argument, enclose the argument in single or double quotation marks.
- Command Option** Information you provide on a command line to indicate the type of action you want the command to take. Precede each option with a hyphen. Do not include any space between the hyphen and the option. (Also see *Default*.)
- Control Key Sequence** A keystroke combination consisting of <CTRL> followed by another key. To enter a control key sequence, hold <CTRL> down while you press another key. Control key sequences invoke certain DM commands. For example, the control key sequence CTRL/C invokes the DM copy command XC. To enter the CTRL/C sequence, press <CTRL> while you type the letter C.

Cursor	The small, blinking box initially displayed in the screen's lower left corner. The cursor marks your current typing position on the screen and indicates which program (Shell or DM) receives your commands.
Data	Information processed by a computer.
Default	Most programs give you a choice of one or more options. If you do not specify an option, the program automatically assigns one. This automatic option is called the default.
Disk	A thin, record-shaped plate that stores data on its magnetic surfaces. The system uses <i>heads</i> (similar to heads in tape recorders) to read and write data on concentric disk tracks. The disk spins rapidly, and the heads can read or write data on any disk track during one disk revolution.
Diskless Node	A node that has no disk for storage, and therefore uses the disk of another node.
Directory	A special type of object that contains information about the objects beneath it in the naming tree.
Display Manager (DM)	The program that executes commands that start and stop processes, and commands that open, close, move, or modify windows and pads.
DM Function Keys	Single keys that invoke DM commands.
DOMAIN	Distributed Operating Multi-Access Interactive Network.
DOMAIN System	A high-speed communications network connecting two or more nodes. Each node can use the data, programs, and devices of other network nodes. Each node contains main memory, and may have its own disk, or share one with another node.

File	The basic named unit of data stored on disk. A file can contain a memo, manual, program, or picture.
Function Keys	See <i>DM Function Keys</i> .
Home Directory	Your initial working directory. Your user account specifies the name of your home directory.
Initial Working Directory	The working directory of the first user process created after you log in.
Input Window	The window that displays a program's prompt and any commands you type.
Insert Mode	Insert mode enables you to change text displayed in windows. You can modify text by repositioning the cursor and inserting characters. The rest of the line moves right as you insert additional characters. To turn insert mode on and off, use <INS> on low-profile keyboards or <INS MODE> on 880 keyboards.
Link	A special type of object that points from one place in the naming tree to another. A link contains the pathname of another object.
Logging In	To begin using the system, log in by typing the L command followed by your username and password. This creates your initial user process.
Main Memory	The node's primary storage area. It stores the instruction that the node is executing, as well as the data it is manipulating.
Memory	Any device that can store information.
Name	A 1- to 32-character string associated with a file, directory, or link. A name can include alphanumeric characters, periods (.), underscores (_), or dollar signs (\$). All names must begin with a letter or dollar sign.

Naming Directory	In addition to its working directory, each process uses a naming directory. Like the working directory, the naming directory points to a certain destination directory. You can display or set the naming directory at any time using the Shell command ND (NAMING_DIRECTORY). The system uses your home directory as the initial naming directory. You can use the shorthand symbol “.” in place of the naming directory name.
Naming Tree	A hierarchical tree structure that organizes network objects.
Network	Two or more nodes sharing information.
Network Root Directory	The top directory in the network, represented by a pair of slashes (//). Each node contains a copy of the network root directory.
Node	A network computer. Each node in the DOMAIN system can use the data, programs, and devices of other network nodes. Each node contains main memory, and may have its own disk, or share one with another node.
Node Entry Directory	A subdirectory of the network root directory. The node entry directory, represented by a single slash (/), is the top directory on each node. Diskless nodes share the node entry directory of their disked partner node.
Object	Any file, directory, or link in the network.
Operating System	The program that supervises the execution of other programs on your node.
Option	See <i>Command Option</i> .
Output Window	The window that displays a process' response to your command.

Pad	A temporary, unnamed file that holds the information you display in a window. A window can display an entire pad, or show only part of the pad.
Parent Directory	The directory one level above the current working directory. A backslash (\) represents the parent directory.
Partner Node	A node that shares its disk with a diskless node.
Password	The word you enter next to the "Password:" prompt when you log in. As you type your password, the system displays periods (.) instead of the letters in your password.
Pathname	A series of names separated by slashes that describe the path the operating system must take to get from some starting point in the network to a destination object. It begins with the starting point's name, and includes every directory name between the starting point and the destination object. A pathname ends with the destination object's name. A pathname may not exceed 256 characters, including the slashes.
Process	A computing environment in which you can execute programs.
Prompt	A message or symbol displayed by a program to indicate that it is ready for your input. The Shell prompt is a dollar sign (\$) and the DM prompt is "Command:".
Root Directory	See <i>Network Root Directory</i> .
Script	A file that you create that contains one or more Shell commands. A script allows you to execute a sequence of commands by entering a single command (the script name).
Shell	A command-line interpreter program used to invoke operating system utility programs.

Shell Command	An instruction you give the system to execute a utility program. For example, when you type DATE, the Shell invokes the utility program that displays the date.
Software	Programs, such as the Shell and the DM, that allow you to perform various tasks.
System Administrator	The person responsible for system maintenance and security at your site.
Transcript Pad	A transcript pad contains a record of your interaction with a process. The process output window provides a view of its transcript pad.
User Account	The system administrator defines a user account for every person authorized to use the system. Each user account contains the name the computer uses to identify the person (<i>username</i>), and the person's <i>password</i> . User accounts also contain project and organization names. The system uses this information to determine who can use the system, and what resources they can use.
Username	The name the computer uses to identify you. Your system administrator assigns you your username. You enter your username during the log-in procedure when the system displays the "Please log in:" prompt.
Utilities	Programs provided with the operating system to perform frequently-required tasks, such as printing a file or displaying the contents of a directory.
Wildcards	Special characters that you can use to represent one or more pathnames.

Window

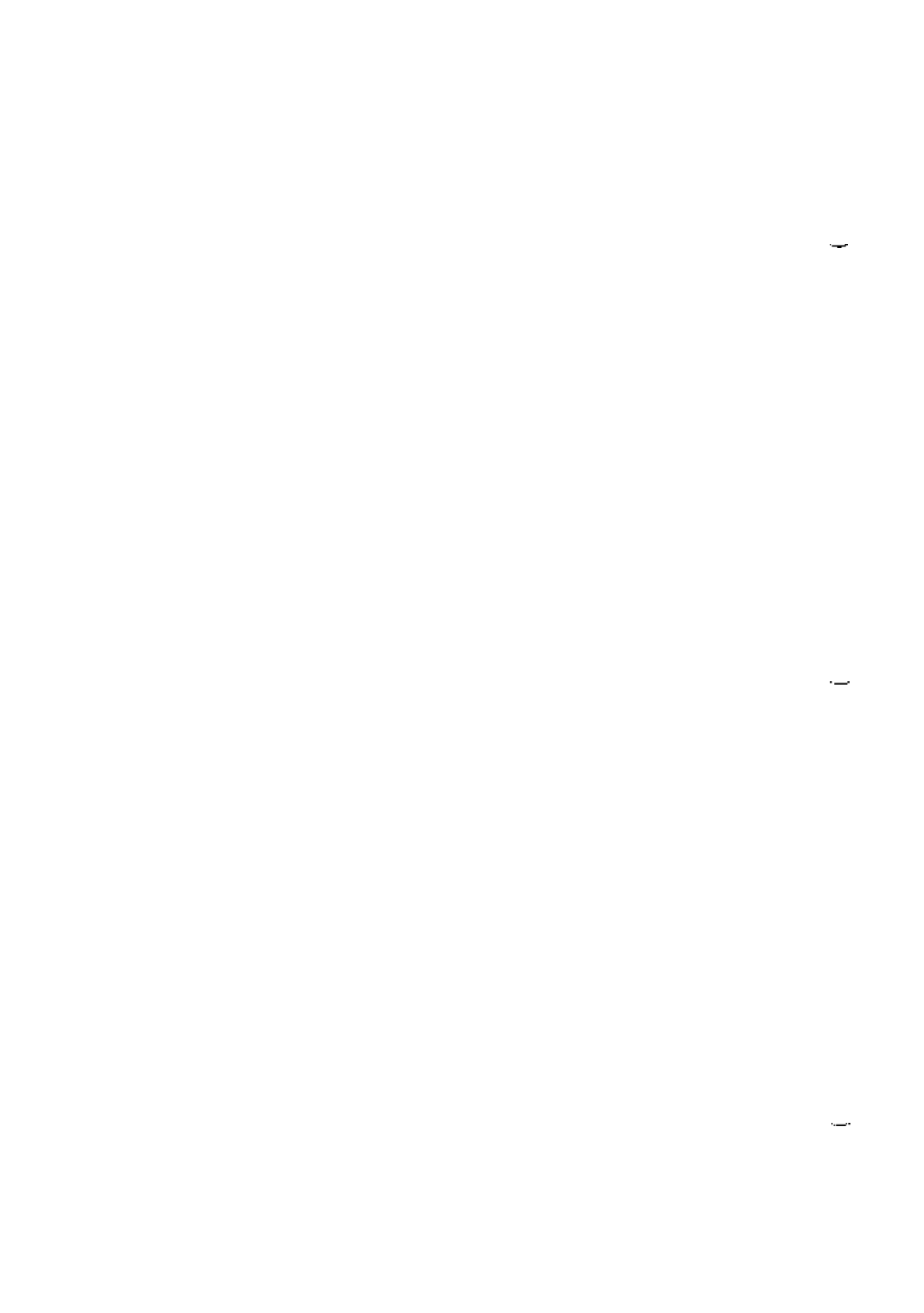
Openings on the screen through which you view information stored in the system. Display management software lets you create several different windows on the screen. Each window is a separate computing environment in which you can execute programs, edit text, or read text. You can move the windows anywhere on your screen, change their size and shape, and overlap or shuffle them as you might papers on your desk.

Window Legend

The area of a window that displays window status information. For example, the window legend of an edit window contains such information as the pathname of the file you're editing, the letter I if the window is in insert mode, and the number of the line at the top of the window.

Working Directory

The default directory in which a process creates or searches for objects.



INDEX

Primary page references are listed first. The letter *f* means “and the following page”; the letters *ff* mean “and the following pages”. Task-oriented entries appear in color. Symbols are listed at the beginning of the index.

Symbols

- < > (angle brackets) iv
- * (asterisk) 5-5
- @ (at) 5-3f
- \ (backslash)
 - in pathnames 3-9
 - in search strings 4-6
- // (double slashes) 3-3, 3-5
- > (greater than symbol) 5-6, 5-8
- < (less than symbol) 5-7f
- ? (question mark) 5-5
- / (slash)
 - in pathnames 3-3, 3-6
 - in search strings 4-6
- ~ (tilde) 3-9, 3-12

A

- access rights 3-5, 4-1
- ABORT key 2-6, 4-9, 4-10
- AGAIN key 2-9f
- alarm window 1-14, 2-9
- AP (ALARM_POP) command 2-9
- argument, command 5-2
- arguments in Shell scripts 5-9
- arrow keys 1-7

B

BACKSPACE key 1-10, 1-15, 1-18, 4-4
BAK filename suffix 4-2

C

cancelling a cut, copy, or search 4-7
case

in searches 4-7
of commands 1-14

changing

window size 2-10ff
your home directory 3-6f
your naming directory 3-9
your working directory 3-8

CHAR DEL key 1-18, 4-4

closing

a Help window 1-15
a read-only pad and window 4-10
an edit window 4-8f

CMD key 1-12

COM directory 5-4f

command

arguments 5-2
format 5-2
line processing 5-4
options 3-5f, 5-2f
search rules 5-4f

commands, DM

CV (CREATE_VIEW) 4-9
L (LOGIN) 1-10
LO 1-18
PN (PAD_NAME) 2-13
PP (PAD_PAGE) 2-6
S (SUBSTITUTE) 4-7f
SC 4-7
UNDO 4-8

commands, Shell

CPF (COPY_FILE) 4-10
CRD (CREATE_DIRECTORY) 3-7
CRL (CREATE_LINK) 3-12
CSR (COMMAND_SEARCH_RULES) 5-4
DATE 1-15
DLF (DELETE_FILE) 4-13

FMT (FORMAT_TEXT) 5-6
 HELP 1-15, 2-9
 LD (LIST_DIRECTORY) 3-5, 5-2f
 LUSR (LIST_USER) 3-5f
 PRF (PRINT_FILE) 4-11, 5-5
 PRFD (PRINT_FILE_DIALOG) 4-12
 RS (REFRESH_SCREEN) 1-15f
 TLC (TRANSLITERATE_CHARACTERS) 5-7
 WD (WORKING_DIRECTORY) 3-7, 5-3
 Command: prompt 1-12, 1-14
 control key sequences 1-16, 1-15
 control keys, window 2-7ff
 COPY key 2-10, 4-6
 copying
 a command 2-9f
 a file 4-10
 text
 in a file 4-6
 to the process input window 2-9f
 correcting errors 1-18, 4-3f
 CPF (COPY_FILE) command 4-10
 CRD (CREATE_DIRECTORY) command 3-7
 creating
 a Shell process 2-6f, 1-16
 a text file 4-1f
 Shell scripts 5-8f
 CRL (CREATE_LINK) command 3-12
 CRUCR (CREATE_USER_CHANGE_REQUEST) v
 CSR (COMMAND_SEARCH_RULES) command 5-4
 CTRL key 1-16, 1-15
 CTRL/B 2-5f
 CTRL/C 4-6, 2-10
 CTRL/E 4-5
 CTRL/G 2-10f
 CTRL/M 4-1, 4-10
 CTRL/N 1-15, 1-16, 2-6, 2-13, 4-9, 4-10
 CTRL/O 2-10, 4-6
 CTRL/P 2-8
 CTRL/R 4-7
 CTRL/T 2-5f
 CTRL/<TAB> 1-8
 CTRL/U 4-7
 CTRL/W 2-12
 CTRL/X 2-12, 4-7
 CTRL/Y 4-9, 4-6
 CTRL/Z 1-16, 2-13

- current directory *see* working directory
- cursor 1-6, 1-12
 - arrow 1-6, 1-8, 1-9
 - box 1-6
 - circle 1-6
 - in print menu 4-12
 - moving the 1-6ff
- CUT key 4-5
- cutting text 4-5

D

- DATE command 1-15
- defining a range of text 4-4
- deleting
 - a file 4-13
 - a range of text 4-5
 - characters 1-18, 4-4
 - lines 1-18, 4-4
- destination object 3-3
- diskless node 3-6
- displaying
 - a file 4-2, 4-9, 2-3f
 - hidden text 2-5f, 2-2
 - hidden windows 2-8
- displays
 - after logging in 1-11ff
 - landscape 1-5
 - portrait 1-5f
- Display Manager 1-11
 - alarm window 1-14, 2-9
 - commands 1-15
 - function keys 1-16
 - input window 1-14
 - output window 1-14
 - prompt 1-12, 1-14
 - windows 1-14
- directories 3-1
 - /COM 5-4f
 - COM 5-4f
 - home 3-6f
 - naming 3-9, 3-12
 - network root 3-3, 3-5
 - node entry 3-5f

- parent 3-8f
- working 3-7f
- discarding changes in an edit file 4-9
- DLF (DELETE_FILE) command 4-13
- DM *see* Display Manager
- dollar sign (\$) prompt 1-12, 1-13, 2-7
- DOMAIN system 1-1

E

- EDIT key 2-3, 4-1, 4-2
- edit window and pad 2-3f, 4-2f
- editing a file 4-2ff
- end-of-file mark (EOF) 1-16, 2-13
- ending the session 1-18f
- enlarging a window 2-10ff
- entering
 - commands 1-14ff
 - DM commands 1-15f
 - filenames 4-2
 - long command lines 5-3f
 - multiple commands on a line 5-3
 - Shell commands 1-15
- errors, correcting 1-18, 4-3f
- EXIT key 4-8

F

- F2 key 2-6
- F3 key 2-6
- F8 key 2-9f
- filenames, entering 4-2
- files 3-1
 - closing 4-8f, 4-10
 - creating 4-1f
 - editing 4-2ff
 - locating *see* pathname
 - naming 4-2
 - opening 4-1f
 - reading 4-9f
- FMT (FORMAT_TEXT) command 5-6
- format *see* command format
- function keys 1-16, 1-15

G

glossary Glossary-1ff
GROW key 2-10f

H

HELP command 1-15, 2-9
home directory 3-6f

I

I (insert mode indicator) 1-14, 1-18
input
 pads 2-2
 windows 1-13, 1-14
input, redirecting 5-6f
INS key 1-18, 4-4
INS MODE key 1-18, 4-4
insert mode 1-18, 4-4
inserting characters 1-18, 4-4
inserting a range of text *see* pasting text
KD (KEY_DEFINITION) command 1-3

K

keyboard
 low-profile 1-3f, 1-16
 880 1-3f
key definitions 1-3
keys
 ABORT 2-6, 4-9, 4-10
 AGAIN 2-9f
 arrow 1-7
 BACKSPACE 1-10, 1-15, 1-18, 4-4
 CHAR DEL 1-18, 4-4
 CMD 1-12
 COPY 2-10, 4-6
 CTRL 1-16, 1-15
 CUT 4-5
 defining 1-3
 EDIT 2-3, 4-1, 4-2
 EXIT 4-8
 F2 2-6

- F3 2-6
- F8 2-9f
- function 1-16, 1-15
- GROW 2-10f
- INS 1-18, 4-4
- INS MODE 1-18, 4-4
- LINE DEL 1-18, 4-4
- MARK 2-10f, 4-4, 4-5, 4-6, 4-8
- MOVE 2-12
- NEXT WNDW 1-12
- PASTE 2-10, 4-5, 4-6
- POP 2-8
- READ 2-3f, 4-9, 4-1
- RETURN 1-10, 1-14
- scrolling 2-6
- SHELL 1-16, 2-7
- SHIFT 1-16
- TAB 1-7f
- UNDO 4-8
- window control 2-7ff

L

- (LOGIN) command 1-10
- landscape display 1-5
- LD (LIST_DIRECTORY) command 3-5
- LINE DEL key 1-18, 4-4
- links 3-11f, 3-2
- LO command 1-18
- log in prompt 1-9f
- logging in 1-9f
- logging off 1-18f
- looking inside a window 2-1f
- LUSR (LIST_USER) command 3-5f

M

- managing windows 2-7
- MARK key 2-10f, 4-4, 4-5, 4-6, 4-8
- marking a range of text 4-4
- mode indicators 1-14, 1-18
- mouse 1-9
 - pushing and popping windows 2-8
 - changing window size 2-10f
 - reading a file 4-10

- mouse keys 1-9
- MOVE key 2-12
- moving
 - a pad under a window 2-5f
 - horizontally 2-6
 - vertically 2-6
 - a window 2-12
 - the cursor 1-6ff
 - to the top and bottom of a pad 2-5f

N

- naming directory 3-9, 3-12
- naming tree 3-1f
- ND (NAMING_DIRECTORY) command 3-9
- network root directory 3-3, 3-5
- NEXT WNDW key 1-12
- node entry directory 3-5f
- nodes 1-1
 - diskless 3-6

O

- objects 3-2
- opening a window to a file 4-1
- operating system 1-11
 - Display Manager 1-11
 - Shell 1-11
- options *see* command options
- organization name 1-10
- organization of the system 3-1ff
- output
 - pads 2-3
 - windows 1-14
- output, redirecting 5-6f
- overstriking characters 1-18, 4-4

P

- pads 2-1ff
 - edit 2-3
 - input 2-2f
 - moving to the top and bottom 2-5f
 - moving under a window 2-5

- output 2-3
- read-only 2-3f, 4-9
- scrolling 2-6
- transcript 2-3
- parent directories 3-8f
- password 1-10
- paste buffer 4-5, 4-6
- PASTE key 2-10, 4-5, 4-6
- pasting text 4-5, 4-6
- pathname 3-3ff
 - symbols 3-9f
- PN (PAD_NAME) command 2-13
- POP key 2-8
- portrait display 1-5f
- PP (PAD_PAGE) command 2-6
- PRF (PRINT_FILE) command 4-11, 5-5
- PRFD (PRINT_FILE_DIALOG) command 4-12
- printing a file 4-10
 - using the print command 4-11
 - using the print menu 4-11f
- process 1-11
 - creating 2-6f, 1-16
 - identification name 1-14
 - input window 1-13
 - output pad 2-3
 - output window 1-14
 - stopping 2-12f, 1-16
 - transcript pad 2-3
- project name 1-10
- prompt 1-9
 - \$ 1-12, 1-13, 2-7
 - Command: 1-12, 1-14
 - DM 1-12, 1-14
 - Edit file: 4-2
 - Please log in: 1-9f
 - Read file: 2-3f, 4-9
 - Shell 1-12, 1-13, 2-7
- pushing or popping windows 2-8f

Q

queuing a file for printing *see* printing a file

R

- R in window legend 4-9
- range of text, defining 4-4
- READ key 2-3, 4-1, 4-9
- read-only window and pad 2-3f, 4-9
- reading
 - a file 4-9f
 - input from a file 5-7
- redirecting input and output 5-6f
- reducing window size 2-10ff
- refreshing the screen 1-16f
- related manuals v
- replacing text *see* substituting text
- resolution name 3-12
- responding to alarms 2-9
- RETURN key 1-10, 1-14
- root directory 3-3, 3-5

S

- S (mode indicator) 1-14
- S (SUBSTITUTE) command 4-7f
- saving
 - editing changes 4-8f
 - a transcript pad 2-13
- scripts, Shell 5-8f
- scrolling
 - a pad 2-6
 - keys 2-6
- search rules, command 5-4f
- searching for text 4-6f
- semicolon (;) 5-3
- Shell 1-11
 - command format 5-2
 - commands 1-12, 1-15, 5-1ff
 - pads 2-1ff
 - process, creating 2-6f, 1-16
 - process windows 1-13f
 - prompt 1-12, 1-13, 2-7
 - scripts 5-8f
 - windows 1-13f
- SHELL key 1-16, 2-7
- SHIFT key 1-16
- SHIFT/↑ 2-6

- SHIFT/↓ 2-6
- SHIFT/← 2-6
- SHIFT/→ 2-6
- shuffling windows 2-8f
- stopping a process 2-12f, 1-18f
- substituting
 - arguments in Shell scripts 5-9
 - text 4-7f
- system administrator 1-5
- system organization 3-1ff

T

- TAB key 1-7f
- TLC (TRANSLITERATE_CHARACTERS) command 5-7f
- touchpad 1-8
- transcript pad 2-3

U

- underscore character (`_`) 4-2, 5-3
- UNDO
 - command 4-8
 - key 4-8
- undoing previous commands 4-8
- user account 1-10
- User Change Request (UCR) v
- username 1-10
- utilities (programs) 1-11, 5-1

V

- video display 1-5
- viewing
 - a file 4-9f
 - hidden text 2-5f, 2-2
 - hidden windows 2-8f

W

- WD (WORKING_DIRECTORY) command 3-7f, 5-3
- wildcards 5-5f
- window legend 1-14

windows 1-1, 2-1f
alarm 1-14, 2-9
changing the size of 2-10ff
control keys 2-7ff
edit 4-2f
input 1-13, 1-14
legend 1-14
moving 2-12
output 1-14
popping (pushing) 2-8f
position 1-11ff
read-only 2-3f, 4-9
working directory 3-7f
writing output to a file 5-6

NOTES

NOTES

GETTING STARTED COMMAND SUMMARY

The following tables summarize the commands and keys described in *Getting Started With Your DOMAIN System*.

DISPLAY MANAGER (DM) COMMANDS

MOVING THE CURSOR

Task	Low-Profile Keyboard	880 Keyboard
Move left one character	←	←
Move right one character	→	→
Move up one line	↑	↑
Move down one line	↓	↓
Move to start of next line	CTRL/K	CTRL/K
Move to beginning of line	⌊←	⌊←
Move to end of line	→⌋	→⌋
Tab right	<TAB>	<TAB>
Tab left	CTRL/<TAB>	CTRL/<TAB>
Move to DM input window	<CMD>	<CMD>
Move to next window on screen	<NEXT WNDW>	<NEXT WNDW>

CREATING AND CONTROLLING A PROCESS

Task	Low-Profile Keyboard	880 Keyboard
Create a new process, transcript pad, and window	<SHELL>	<SHELL>
Stop a process, delete the input window, and close pads	CTRL/Z	CTRL/Z
Save the transcript pad in a file	<CMD> PN pathname	<CMD> PN pathname

MOVING A PAD UNDER A WINDOW

Task	Low-Profile Keyboard	880 Keyboard
Move cursor to first character in pad	CTRL/T	CTRL/T
Move cursor to last character in pad	CTRL/B	CTRL/B
Move pad by pages	<div style="display: flex; justify-content: space-around;"> ↑ ↓ </div>	<div style="display: flex; justify-content: space-around;"> ↑ ↓ </div>
Move pad by lines	<SHIFT>/↑ <SHIFT>/↓	<F2>, <F3> <F2>, <F3>
Move pad by characters	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center; margin-bottom: 10px;"> ← , <SHIFT>/← </div> <div style="display: flex; align-items: center;"> → , <SHIFT>/→ </div> </div>	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="display: flex; align-items: center; margin-bottom: 10px;"> ← </div> <div style="display: flex; align-items: center;"> → </div> </div>

MANAGING WINDOWS

Task	Low-Profile Keyboard	880 Keyboard
Change a window's size	<GROW>	CTRL/G
Move a window	<MOVE>	CTRL/W
Cancel changing window size	CTRL/X	CTRL/X
Cancel moving a window	CTRL/X	CTRL/X
Close pad and window; update file	<EXIT>	CTRL/Y
Close pad and window; no update of file	<ABORT>	CTRL/N
Copy text to process input window	<AGAIN>	<F8>

MANAGING THE DISPLAY

Task	Low-Profile Keyboard	880 Keyboard
Request help	<HELP>	See Shell command HELP
Login	<CMD> L username	<CMD> L username
Log off	<CMD> LO	<CMD> LO
Place a mark	<MARK>	<MARK>
Respond to DM alarm and pop window	<CMD> AP	<CMD> AP

CREATING A PAD AND WINDOW TO READ AND EDIT FILES

Task	Low-Profile Keyboard	880 Keyboard
Create edit pad and window through which to view it	<EDIT>	<EDIT>
Create read-only pad and window	<READ>	<READ>

EDITING A PAD

Task	Low-Profile Keyboard	880 Keyboard
Set read/write mode	CTRL/M	CTRL/M
Set Insert/overstrike mode	<INS>	<INS MOD▶
Insert NEWLINE character	<RETURN>	<RETURN>
Insert new line after current line	<F1>	<F1>
Insert end-of-file mark	CTRL/Z	CTRL/Z
Delete character at cursor	<CHAR DEL>	<CHAR DEL>
Delete character before cursor	<BACKSPACE>	<BACKSPACE>
Delete word of text	<F6>	<F6>
Delete from cursor to end of line	<F7>	<F7>
Delete entire line	<LINE DEL>	<LINE DEL>
Copy defined range of text to paste buffer	<COPY>	CTRL/C

EDITING A PAD (continued)

Task	Low-Profile Keyboard	880 Keyboard
Cut (delete) defined range of text and write it to paste buffer	<CUT>	CTRL/E
Paste (write) text in paste buffer into pad	<PASTE>	CTRL/O
Search forward for string	<CMD> /string/	<CMD> /string/
Repeat last forward search	CTRL/R	CTRL/R
Search backward for string	<CMD> \string\	<CMD> \string\
Repeat last backward search	CTRL/U	CTRL/U
Cancel copy, out, or search	CTRL/X	CTRL/X
Undo previous command	<UNDO>	<CMD> UNDO
Update edit file without closing edit pad	<SAVE>	<CMD> PW
Substitute <i>string2</i> for all occurrences of <i>string1</i> in defined range	<CMD> S/string1/string2/	
Substitute <i>string2</i> for first occurrence of <i>string1</i> in each line of defined range	<CMD> SO/string1/string2/	

SHELL COMMANDS

Task	Shell Command
Copy a file	CPF source [target]
Create a directory	CRD pathname
Create a link	CRL pathname
Display command search rules	CRS
Display the current date and time	DATE
Delete a file	DLF pathname
Display information about commands	HELP [command]
List the contents of a directory	LD [pathname]
List network users and their node entry directories	LUSR [--me]
Print a file (command)	PRF pathname [--pr name]
Print a file (menu)	PRFD
Set or show the naming directory	ND [pathname]
Set or show the working directory	WD [pathname]

Reader's Response

Apollo Computer uses readers' comments to revise and improve our manuals.

Document Title: *Getting Started With Your DOMAIN System*

Order No.: 002348

Revision: 03

What is the best feature of this manual? _____

Please list any errors, omissions, or problem areas in the manual. (Identify errors by page, section, figure, or table number wherever possible.)

What type of user are you?

- System programmer; language _____
- Applications programmer; language _____
- Manager/Professional
- Technical Professional
- Administrative/Support Personnel
- Other

How often do you use the DOMAIN system? _____

Nature of your work on the DOMAIN system: _____

Your Name

Date

Organization

Street Address

City

State

Zip

No postage necessary if mailed in the U.S.

cut along dotted line



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS

PERMIT NO. 78

CHELMSFORD, MA 01824

POSTAGE WILL BE PAID BY ADDRESSEE

**APOLLO COMPUTER INC.
Technical Publications
P.O. Box 451
Chelmsford, MA 01824**



Reader's Response

Apollo Computer uses readers' comments to revise and improve our manuals.

Document Title: *Getting Started With Your DOMAIN System*

Order No.: 002348

Revision: 03

What is the best feature of this manual? _____

Please list any errors, omissions, or problem areas in the manual. (Identify errors by page, section, figure, or table number wherever possible.)

What type of user are you?

- System programmer; language _____
- Applications programmer; language _____
- Manager/Professional
- Technical Professional
- Administrative/Support Personnel
- Other

How often do you use the DOMAIN system? _____

Nature of your work on the DOMAIN system: _____

Your Name

Date

Organization

Street Address

City

State

Zip

No postage necessary if mailed in the U.S.

cut along dotted line



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 78 CHELMSFORD, MA 01824

POSTAGE WILL BE PAID BY ADDRESSEE

**APOLLO COMPUTER INC.
Technical Publications
P.O. Box 451
Chelmsford, MA 01824**

