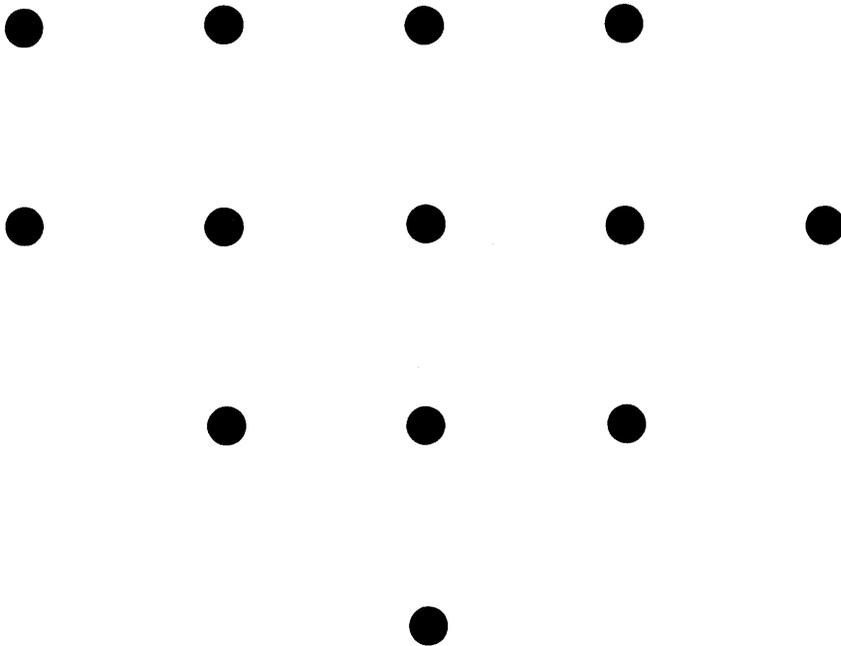


SunView User's Guide



The Sun logo, Sun Microsystems, and Sun Workstation are registered trademarks of Sun Microsystems, Inc.

Sun, Sun-3, Sun-4, SunOS, and SunLink, are trademarks of Sun Microsystems, Inc.

All other products or services mentioned in this document are identified by the trademarks or service marks of their respective companies or organizations.

Copyright © 1988,1989, Sun Microsystems, Inc. All Rights Reserved. No part of this work covered by copyright hereon may be reproduced or used in any form or by any means — graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems — without permission of the copyright owner.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. We acknowledge the following individuals and institutions for their role in its development: The Regents of the University of California and the Electrical Engineering and Computer Sciences Department at the Berkeley Campus of the University of California and Other Contributors.

Contents

PART ONE: LESSONS	1
Chapter 1 Getting Started with SunView	1
1.1. Starting SunView	1
The Tools at Startup	2
The Console at Startup	3
1.2. The Mouse and the Pointer	3
Left-Handed Mousing	4
What to Do When the Pointer Won't Move	4
The Mouse Buttons	4
Bringing Up and Dismissing a Menu	5
1.3. Opening and Closing Frames	6
Toggles	8
Grayed Menu Items	8
1.4. Using the Text Editor	8
Choosing a Menu Item	8
The Text Editor Frame	9
Text Windows	9
The Text Menu	10
Previewing a Menu	10
The Keyboard Focus	11
The Delete Key	12
The File⇒ Pull-Right Menu	12
Alert Windows	13
1.5. Selecting Text	14

Selecting a Word	15
Extending the Selection	16
More About Selections	17
Inserting Text	17
Extending the Selection Backward	18
More About the Edit ⇒ Pull-Right Menu	18
The Clipboard	18
Copy, Paste, and Cut	19
Some Menu Items Require a Selection	20
1.6. Saving a New File	21
About the Directory	22
Undoing Edits	22
1.7. The SunView Menu	23
Printing from the SunView Menu	24
Locking the Screen	24
Chapter 2 SunView Accelerated	27
2.1. Introduction to Accelerators	27
2.2. Unlocking the Locked Screen	28
2.3. Working with Several Frames	29
“Back” and “Front”	30
Function Keys	31
Back and Front Accelerators	32
Moving and Resizing Frames	32
2.4. Rearranging the Screen	33
Moving Icons	37
Zoom and FullScreen	37
2.5. Loading a File	38
2.6. Scrolling	40
Using the Scroll Boxes	40
Scrolling by the Line	41
Scrolling by the Window	41
The Bar and the Bubble	42

Scrolling Directly to a Location	42
2.7. Displaying and Moving the Caret	43
2.8. Finding and Replacing Text	43
Pushing Panel Buttons	44
Replacing Text	44
Menus in Panels	46
2.9. Wrapping Long Lines of Text	47
Wrap at Character	48
Wrap at Word	49
Clip Lines	50
2.10. Exiting SunView	51
2.11. A Word to the Wise	51
PART TWO: REFERENCE	53
Chapter 3 Starting and Exiting SunView	55
3.1. Starting SunView	55
Customizing the Background Pattern	55
Background in Inverse Video	56
3.2. Saving the Layout	57
3.3. Locking the Screen	58
3.4. Exiting SunView	58
Emergency Exit	58
Chapter 4 All about Keyboards	59
4.1. Which Sun Keyboard Do You Have?	59
4.2. How to Label the Keyboard	62
4.3. Left-Handed Keyboard Use	62
4.4. Click-to-Type Keyboard Focus	66
Why Use Click-to-Type?	66
How to Get Click-to-Type	67
4.5. Type4 Keyboards	67
International Key Caps	67
Key Clicks	67

4.6. Mapping Functions to Keys	68
Tty Windows and the <code>.ttyswrc</code> File	68
Enabling Page Mode	68
Format of the <code>.ttyswrc</code> File	69
Sample <code>.ttyswrc</code> File	70
Text Windows and the <code>.textswrc</code> File	71
Function Keys That Cannot Be Mapped	71
Special Filters	71
Format of the <code>.textswrc</code> File	71
Chapter 5 All About Frames and Windows	75
5.1. Anatomy of a Frame	75
The Frame Header	76
Pop-Up Frames	76
5.2. Text Windows	76
Scratch Windows	77
Changing the Scratch Window Size	77
Changing Direction with the Shift Key	77
Mousing Ahead	78
Filtering Selections	78
5.3. Command Windows	78
The Command Line	78
Changing Command Modes	79
5.4. Panels	79
Editing Panel Text Items	80
Alert Windows	80
Accelerators for Alert Buttons	81
5.5. Tty Windows	81
The Block Cursor	81
Scrollbar Not Restored	81
5.6. Canvases	81
5.7. Managing Multiple Frames	81

Chapter 6 All about Selections	83
6.1. Mouse, Pointer, Caret, and Block Cursor	83
About the Mouse	83
About the Pointer	84
Scrollbar Pointer Shapes	84
About the Caret	85
About the Block Cursor	85
6.2. Primary Selections	85
Selecting a Character	86
Extending the Selection	86
The Insertion Point	87
6.3. Basic Editing	87
Inserting Text	87
Erasing a Character	88
Erasing Words	88
Erasing Lines	88
Deleting Text	88
Reinserting Deleted Text	88
Deleting Line Breaks	88
6.4. Copying, Pasting, and Cutting	89
Copying and Pasting a Selection	90
Cutting and Pasting a Selection	91
Cutting and Typing	92
Copying Text Between Windows	93
6.5. Undoing Editing	94
6.6. Selecting by Multi-Clicking	95
Extending a Multi-Click Selection	95
6.7. Secondary Selections	96
Finding a Secondary Selection	98
Stopping a Secondary Selection	98
6.8. Pending-Delete Selections	98
6.9. Quick Editing with Secondary Selections	100
Secondary Paste	102

Secondary Copy	103
Secondary Cut	104
Swapping Text	105
6.10. Text Handling Comparison	106
6.11. Using <code>get_selection</code>	107
Chapter 7 All about Scrolling	109
7.1. Anatomy of the Scrollbar	109
7.2. Basic Scrolling	109
Scrolling Up or Forward	109
Scrolling Down or Backward	110
Scrolling a Windowful at a Time	110
Continuous Scrolling	110
7.3. Direct Scrolling	110
Thumbing in the Bar	110
Fine Tuning	111
Chapter 8 All about Menus	113
8.1. Menus in General	113
How to Bring Up a Menu	113
Grayed Menu Items	114
Defaults in Pull-Right Menus	114
Stay-Up Menus	115
Menus in Panels	115
8.2. The SunView Menu	116
The “Shells⇒” Pull-Right Menu	116
The “Editors⇒” Pull-Right Menu	116
The “Tools⇒” Pull-Right Menu	116
The “Services⇒” Pull-Right Menu	116
“Redisplay All”	116
“Printing⇒”	116
“Remote Login⇒”	117
“Save Layout”	117

“Lock Screen”	117
“Exit SunView”	117
8.3. Customizing the SunView Menu	118
Making a <code>.rootmenu</code> File	118
Icons in the SunView Menu	120
8.4. The Frame Menu	121
“Close” and “Open”	121
The “Move⇒” Pull-Right Menu	121
The Bounding Box	121
“Move⇒Unconstrained”	122
“Move⇒Constrained”	122
The “Resize⇒” Pull-Right Menu	122
“Resize⇒Unconstrained”	122
“Resize⇒Constrained”	123
“Resize⇒Zoom” and “Resize⇒UnZoom”	124
“Resize⇒FullScreen”	124
“Front”	125
“Back”	125
“Props”	126
“Redisplay”	126
“Quit”	126
8.5. The Text Menu	127
8.6. The “File⇒” Pull-Right Menu	127
“Save Current File”	127
Backup Files	127
Remember to Save Frequently	127
“Store as New File”	127
“Load File”	128
“Include File”	128
“Set Directory”	128
“Empty Document”	129
“Finishing Up⇒”	129
8.7. The “Edit⇒” Pull-Right Menu	130

“Again”	130
Again Ignores Undo	130
“Again” and “Find”	131
“Undo⇒”	133
“Undo Last Edit”	133
“Undo All Edits”	133
“Copy”, “Paste”, and “Cut”	133
“Show Clipboard⇒”	134
“Copy then Paste”	134
8.8. The “Display⇒” Pull-Right Menu	134
“Select Line at Number”	134
“What Line Number?”	134
“Split View”	135
Many Views, One Caret	135
Why Split a View?	135
“Destroy View”	137
“Show Caret at Top”	137
“Move Caret to Start”	137
“Move Caret to End”	138
“Change Line Wrap⇒”	138
“Wrap at Character”	138
“Wrap at Word”	138
“Clip Lines”	138
8.9. The “Find⇒” Pull-Right Menu	138
The Find and Replace Frame	138
All Text Cycle Item	139
“Find Selection⇒”	140
“Find Clipboard⇒”	140
Accelerator for Find⇒	140
“Show Clipboard⇒”	140
The Find Marked Text Frame	140
“Match Delimiter”	141
“Replace >field< ⇒”	141

8.10. The “Extras⇒” Pull-Right Menu	141
“Format”	141
“Pretty-print C”	142
“Capitalize⇒”	142
“abcd->ABCD”	142
“ABCD->abcd”	142
“abcd->Abcd”	142
“Shift Lines⇒”	142
“Insert Brackets⇒”	142
“Remove Brackets⇒”	142
8.11. Customizing the “Extras⇒” Menu	142
8.12. The “Cmd Modes⇒” Pull-Right Menu	144
“Enable Editing”	144
“Disable Scrolling”	144
8.13. The Tty Menu	145
“Stuff”	145
“Enable Page Mode”	145
“Copy then Paste”	145
Chapter 9 All About Accelerators	147
9.1. Function Key Accelerators	147
The Control Key	148
The Shift Key	148
9.2. Mouse Button Accelerators	149
9.3. Keyboard Accelerators	150
The Meta (◆) Keys	150
9.4. Erase Accelerators	150
9.5. Caret Accelerators	150
The Home and End Keys	152
Arrow Keys	152
Caret Action Mnemonics	152
9.6. Menu Accelerators	153
9.7. Other Keyboard Accelerators	154

9.8. SunOS 3.x Keyboard Accelerators	154
9.9. Alert Button Accelerators	155
Chapter 10 The Defaults Editor	157
10.1. Starting the Defaults Editor	157
10.2. Quitting the Defaults Editor	157
10.3. The Defaults Editor Windows	158
The Control Panel	158
The Category Cycle Item	158
The Save Button	159
The Quit Button	159
The Reset Button	159
The Edit Item Button	159
The Message Panel	159
The Defaults Panel	159
The Scratch Window	159
10.4. Cycling Through Categories	160
Compatibility Category	160
Defaults Category	160
Help Category	160
Indent Category	160
Input Category	160
Mail Category	160
Menu Category	160
Scrollbar Category	161
SunView Category	161
Text Category	161
Tty Category	161
10.5. Changing Defaults	161
Saving Changes	161
When Changes Take Effect	161
10.6. Some Interesting Defaults	162
10.7. Editing Default Lines	164

Chapter 11 All about Tools	167
11.1. Starting a Tool	167
11.2. Tools Cross-Reference	168
11.3. Shells	168
Command Tool	169
Shell Tool	170
Graphics Tool	171
Console	172
11.4. Editors	173
Text Editor	173
Defaults Editor	174
Icon Editor	175
Font Editor	177
11.5. Other Tools	179
Mail Tool	179
Dbx (debug) Tool	180
Performance Meter	181
Clock	182
Chapter 12 SunView Terminology	183
12.1. General SunView Terms	183
12.2. Mouse and Pointer Terms	183
12.3. Text Terms	184
12.4. Scrolling Terms	185
12.5. Menu Terms	185
12.6. Frame and Window Terms	186
Appendix A Command Line Arguments	189
Index	191

Tables

Table 2-1 The Left Function Keys	32
Table 4-1 Keys Not Available for Remapping in <code>.ttyswrc</code>	68
Table 4-2 Keys Not Available for Remapping in <code>.textswrc</code>	71
Table 6-1 Copy, Paste, and Cut — Alternative Methods	89
Table 6-2 Quick Editing with Secondary Selections	101
Table 6-3 Text Handling in Windows and Panels	106
Table 9-1 Function Key Accelerators	148
Table 9-2 Mouse Button Accelerators	149
Table 9-3 Erase Accelerators	150
Table 9-4 Caret Accelerators	151
Table 9-5 Menu Accelerators	153
Table 9-6 Other Keyboard Accelerators	154
Table 9-7 SunOS 3.x Keyboard Accelerators	154
Table 10-1 When Changed Defaults Take Effect	162
Table 11-1 Tool Names Cross-Reference	168

Figures

Figure 1-1 Starting SunView from a SunOS Command Line	1
Figure 1-2 Top of SunView Startup Screen	2
Figure 1-3 The Pointer	3
Figure 1-4 Moving the Mouse on Its Pad	4
Figure 1-5 Bringing Up a Menu	5
Figure 1-6 The Frame Menu	7
Figure 1-7 The Console Frame Closed to an Icon	7
Figure 1-8 An Open Text Editor Frame	9
Figure 1-9 Previewing the Text Menu	11
Figure 1-10 The File⇒ Pull-Right Menu	13
Figure 1-11 An Alert Window	14
Figure 1-12 The Electronic Mail Limerick	15
Figure 1-13 Selecting and Cutting a Word	16
Figure 1-14 Result of Cutting and Typing	16
Figure 1-15 Showing the Clipboard	19
Figure 1-16 Copy and Cut Require a Selection	20
Figure 1-17 Saving a New File	21
Figure 1-18 Filename in the Frame Header	22
Figure 1-19 The File Has Been Edited	23
Figure 1-20 The SunView Menu	23
Figure 1-21 Printing the Selection	24
Figure 1-22 The Locked Screen	25
Figure 2-1 Unlocking the Screen	28

Figure 2-2 Three Open Frames	29
Figure 2-3 Exposing a Partly Hidden Frame	31
Figure 2-4 Resize⇒ Alert Window	34
Figure 2-5 Rearranging the Screen—Step 1	34
Figure 2-6 Rearranging the Screen—Step 2	35
Figure 2-7 Rearranging the Screen—Step 3	36
Figure 2-8 Rearranging the Screen—Final Step	37
Figure 2-9 Loading the Limerick File	39
Figure 2-10 Scrollbar in the Scratch Window	40
Figure 2-11 The Find and Replace Frame	43
Figure 2-12 Find and Replace Frame Menu	44
Figure 2-13 Finding and Replacing Text	45
Figure 2-14 A Cycle Item’s Menu	46
Figure 2-15 A Document with Character Wrap	48
Figure 2-16 A Document with Word Wrap	49
Figure 2-17 A Document with Clipped Lines	50
Figure 3-1 A Cheering Background	56
Figure 3-2 The Customized Screen in Inverse Video	57
Figure 4-1 The Type4 Keyboard	60
Figure 4-2 The Type3 Keyboard	60
Figure 4-3 The Type2 Keyboard	61
Figure 4-4 How to Label the Type4 Keyboard	63
Figure 4-5 How to Label the Type3 Keyboard	64
Figure 4-6 How to Label the Type2 Keyboard	65
Figure 4-7 Tty Escape Sequences	69
Figure 4-8 Example of a <code>.ttyswrc</code> File	70
Figure 5-1 The Parts of a Frame	75
Figure 5-2 Choice Buttons in the Find Marked Text Frame	80
Figure 5-3 Alert Buttons	80
Figure 6-1 Extending a Multi-Click Selection	96

Figure 8-1 SunView Menu File in /usr/lib/rootmenu	119
Figure 8-2 The Scratch Window Resized	124
Figure 8-3 Overlapping Frames	126
Figure 8-4 Split View Alert Window	135
Figure 8-5 Split View in Command Tool	136
Figure 8-6 Split View in Text Editor	137
Figure 8-7 The Find and Replace Frame	139
Figure 8-8 The Find Marked Text Frame	140
Figure 8-9 Extras⇒ Menu File in /usr/lib/.text_extras_menu	143
 Figure 9-1 Alert Buttons	 155
 Figure 10-1 The Defaults Editor Icon	 157
Figure 10-2 The Defaults Editor Frame	158
 Figure 11-1 Starting a Tool	 167
Figure 11-2 Command Tool Frame	169
Figure 11-3 Shell Tool Frame	170
Figure 11-4 Graphics Tool Frame	171
Figure 11-5 Console Frame	172
Figure 11-6 Text Editor Frame	173
Figure 11-7 Defaults Editor Frame	174
Figure 11-8 Icon Editor Frame	175
Figure 11-9 Font Editor Frame	177
Figure 11-10 Mail Tool Frame	179
Figure 11-11 Dbx Tool Frame	180
Figure 11-12 Clock Frame	182

Preface

This manual helps you get started using SunView windows whether you're a brand-new user or an expert. This manual is organized in two sections:

- The *Lessons* section takes you step by step through some of the most important window operations. You'll acquire a working knowledge of SunView and be able to perform useful work when you're done.
- The *Reference* section describes SunView fully, including many advanced features that are not covered in the lessons. After you're comfortable with the contents of the lessons, it's a good idea to read the *Reference* section carefully to add additional skills to your repertoire.

If you're new to SunOS™ and UNIX† operating system commands, make sure you read *SunOS User's Guide: Getting Started* before you begin this manual.

If you're new to window-based systems, the lessons will get you started editing text and using SunOS command interpreters in a window environment. Windows are fun to work with — don't be afraid to try things out! The lessons provide a painless way to get started.

Other manuals for beginners are:

SunOS User's Guide: Doing More

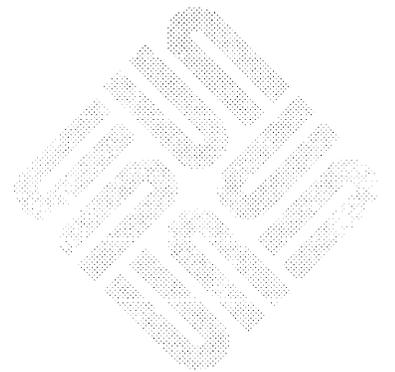
SunOS User's Guide: Customizing Your Environment

SunOS User's Guide: Basic Troubleshooting

If you're an experienced user, you'll probably want to jump right in...and you can. But do take time to skim the *Reference* section so you won't miss out on some of SunView's most attractive new features. You'll find additional information on SunView and its tools in the *SunOS Reference Manual*, also known as the "Man Pages." The *SunOS Reference Manual* is available online through the `man` command.

† UNIX is a registered trademark of AT&T.

PART ONE: LESSONS



Getting Started with SunView

SunView is a window-based environment that lets you run SunOS command shells, editors, debuggers, and other tools, each in its own window or “frame.” SunView lets you look at several files or directories at the same time and run several commands at the same time.

This lesson teaches you how to:

- Enter the SunView environment from a SunOS command line.
- Use the three-button mouse.
- Work with icons, frames, windows, and menus.
- Select and edit text.
- Create and save a file.
- End a work session.

Note that a new facility, Help, is available from most windows, frames, and panels. However, the Help text must be supplied by someone at your installation. Refer to *4.1 Release Notes* for more details.

1.1. Starting SunView

To bring up SunView:

- Reply to the command prompt by typing the commands `cd` (for *change directory*) and `sunview`, as shown in the following figure. Press the `Return` key after each command.

Figure 1-1 *Starting SunView from a SunOS Command Line*

```
venus% cd
venus% sunview
```

NOTE: The command prompt that appears throughout this manual is `venus%`.

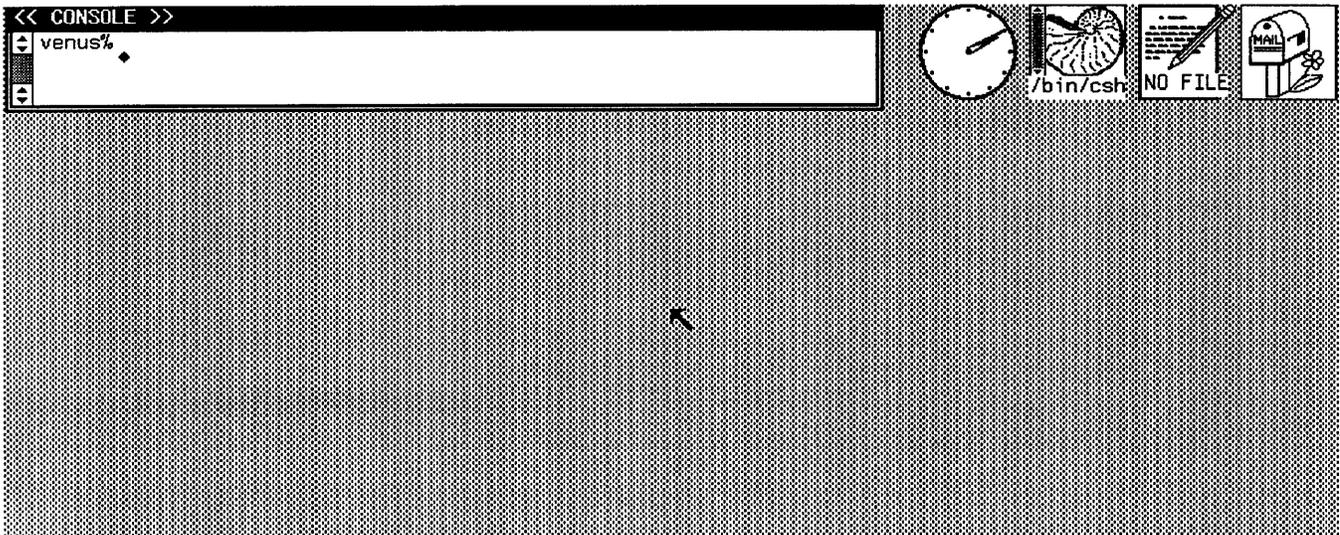
The `cd` command puts you in your home directory before starting SunView. The illustrations in this manual will be easier to follow if you start SunView from your home directory.

For more information on `cd` and other SunOS commands referred to in this manual, see *SunOS User's Guide: Getting Started* and the *SunOS Reference*

Manual, which is also available online through the `man` command.

After a few seconds the screen turns gray and images appear one by one. This is the SunView startup screen. The top part of the screen is shown in the following figure. The rest of the startup screen is gray. The gray part is known as the *background*.

Figure 1-2 Top of SunView Startup Screen



NOTE: If your system administrator has customized your system, your SunView startup screen may not look exactly like this.

The Tools at Startup

On the gray background you'll see five images: a wide rectangle and four small pictures. From left to right, these are:

- An open Console frame (where system messages are usually displayed).
- A Clock icon (a clock face — the hands move to display the time).
- A Command Tool icon (a nautilus shell with scrollbar).
- A Text Editor icon (a document with a pencil).
- A Mail Tool icon (a mailbox — the flag is up when there's mail for you).

Each of the images is a *frame* belonging to a *tool*, or window application program. The Console frame is open, and the other frames are closed. In other words, open frames display *windows* and closed frames are represented by *icons*. You'll hear more about closed frames later in this lesson. When a frame is open, its windows are ready to receive input from the keyboard and mouse. You can close a frame to an icon and the command it's processing continues to process.

Frames can contain several windows. The Text Editor frame, which you'll be learning about throughout this manual, contains two windows. The Console frame, described in the next section, contains only one.

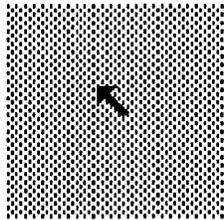
The Console at Startup

The *Console* is a Command Tool (a tool that contains a command interpreter) with a special function: this is where the SunOS operating system, SunView, and some applications display messages. For example, if you're having network problems, the error messages appear in the Console window. That's why it's a good idea to leave the Console just as it is at startup. If you need a Command Tool, you can always open another one. You'll find out how to do this in a few pages.

1.2. The Mouse and the Pointer

Look in the middle of the screen for a black arrow pointing “northwest.”

Figure 1-3 *The Pointer*



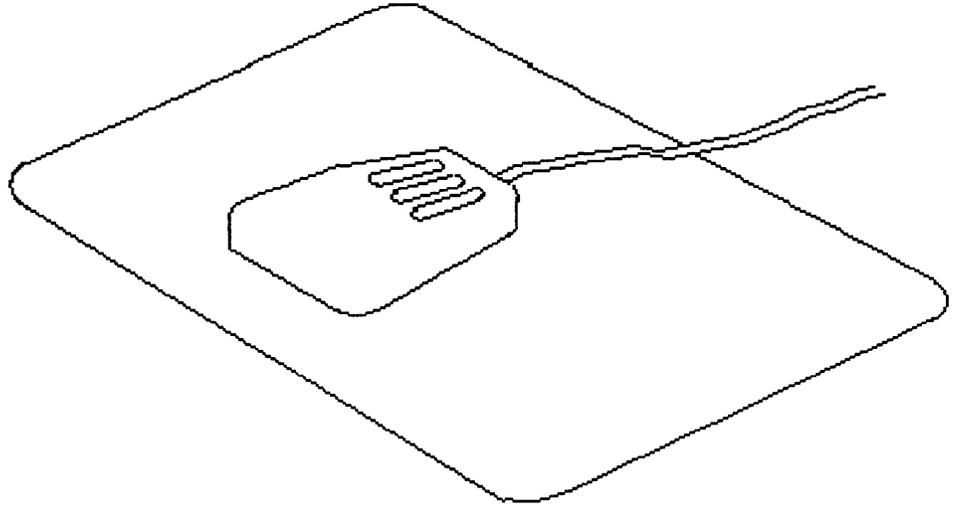
This arrow is known as the *pointer*. The mouse moves the pointer around the screen so that you can open icons, bring up menus, select text, and perform other SunView tasks.

First make sure the mouse pad (the silver plate with the blue grid) is on the right of the keyboard with its wide side facing you. Place the mouse on the mouse pad: buttons up, red light down, perpendicular to the wide side of the pad.¹

To move the pointer, move the mouse around on its pad and note the corresponding movement of the pointer on the screen. Try that now. Your fingers should be resting comfortably on the mouse buttons, with the “tail” facing away from you.

The pointer can take many shapes. No matter what shape the pointer is, it can be moved by the mouse.

¹ This description of the optical mouse doesn't apply to the mechanical mouse. If you have a mechanical mouse with a rolling ball on the bottom, you can disregard references to the pad.

Figure 1-4 *Moving the Mouse on Its Pad*

Left-Handed Mousing

If you're left-handed, you can use the mouse on the left side of the keyboard. You can also change the function keys so that the functions are accessible from the right side of the keyboard. With left-handed mousing, the LEFT mouse button is the one under your index finger, closest to the keyboard. See Chapter 4, "All about Keyboards", in the *Reference* section for more information about SunView options for left-handed users.

What to Do When the Pointer Won't Move

If the pointer doesn't respond to the mouse, try moving the mouse rapidly in large circles on the pad several times. This initializes the mouse. It should then start to control the pointer position.

If the pointer still won't move, check to make sure the mouse is plugged in to your workstation without a loose connection. If that fails to solve the problem, unplug the mouse from the workstation and plug it in again.

If the mouse still doesn't work, the mouse or the workstation may be defective — contact your system administrator.

The Mouse Buttons

Here's how the three-button mouse works:

- The LEFT mouse button — LEFT for short — is used mainly to select text, open icons, and push buttons in panels and alert windows.
- The MIDDLE mouse button — MIDDLE for short — is used mainly to extend or adjust a selection, move a frame or icon, and scroll in a window.
- The RIGHT mouse button — RIGHT for short — is used mainly to bring up and choose from menus.

There are two actions you can take with a mouse button: *click* or *hold down*. To try the buttons, first move the mouse so that the pointer moves into the Console window — the white rectangle on the top left of the screen.

You *click* a mouse button by pressing and releasing it, similar to the way you type by pressing and releasing a key on the keyboard.

- With the pointer over the gray background, click the MIDDLE mouse button, just to see how it feels. (You can do this as many times as you like. Nothing will happen.)
- With the pointer still over the gray background, click the MIDDLE mouse button. You'll hear a single click. Move the pointer around while continuing to hold down the MIDDLE mouse button. The pointer will change shape as it moves over different objects on the screen. Then release the mouse button.

Holding down the mouse button while moving the pointer is a combination you'll be using often in SunView.

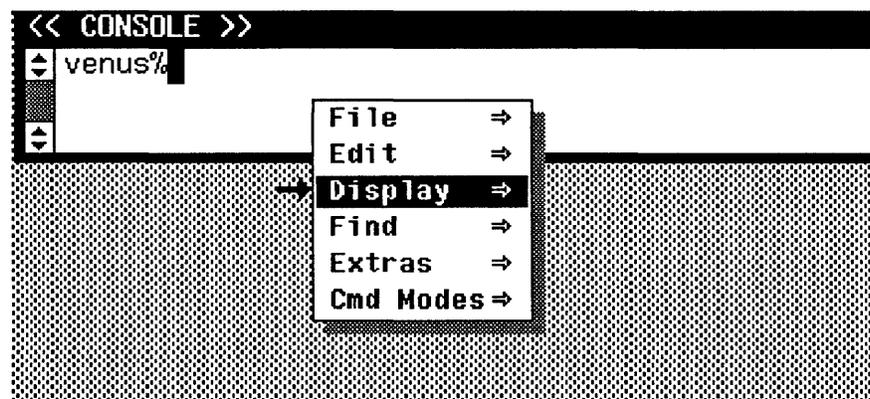
Bringing Up and Dismissing a Menu

Read this section *before* you try the instructions it contains. It's important to follow them exactly to avoid choosing a menu action you don't intend to choose.

SunView tools don't show you a menu until you request one. To bring up a menu, hold down the RIGHT mouse button. The location of the pointer determines which menu comes up.

- With the pointer in the white part of the Console window, hold down the RIGHT mouse button. Don't let go until you either choose a menu action or dismiss the menu without making a request — you'll learn how in this section.

Figure 1-5 *Bringing Up a Menu*



The figure above shows a *menu* — a list of actions you can request. The **Display⇒** menu item is highlighted.

When a menu item is *highlighted*, it's ready to be chosen. Highlighted text is displayed in *inverse video* (white characters on a black background).

When you want to choose an item in a menu, you move the pointer over the item so that it appears highlighted, then release the mouse button. But don't do that yet.

- Keep holding down the RIGHT mouse button as you move the pointer left, away from the menu, so that no menu item is highlighted.
- Release the mouse button. The menu disappears.

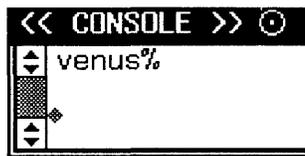
To *dismiss a menu* — close it without taking any action — move the pointer off the menu so that no menu item is highlighted, then release the mouse button. Now you're ready to choose a menu item to close the Console frame.

NOTE: If your system has stay-up menus, things are a bit different. You click RIGHT to bring up a menu, and it stays up until you dismiss it or choose from it by clicking RIGHT again. Chapter 10, "The Defaults Editor", in the *Reference* section, tells you how to get stay-up menus. Normal menus are assumed for this chapter and the following chapter.

1.3. Opening and Closing Frames

Most of the work you do in a tool is done with its frame open. You can't really work in a closed frame, although you can start work (say, a compilation) and then close the frame. The *frame menu*, one of the standard SunView menus, lets you open or close a frame, change its appearance, and quit the tool it belongs to.

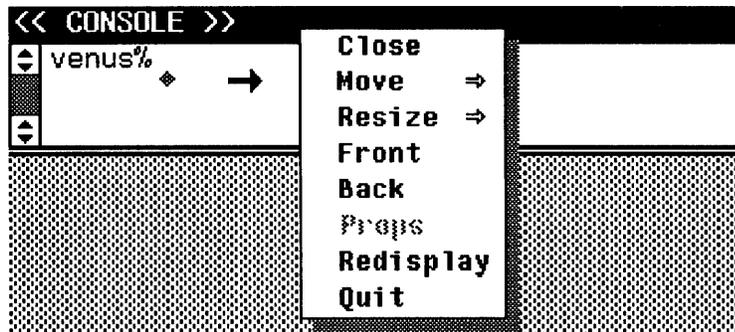
- Move the pointer into the *frame header* — the black stripe at the top of the Console frame where the word << CONSOLE >> appears.



Notice that the pointer changes shape. It's now a *target* — a circle with a dot inside. The target means that you can bring up the Frame menu. The pointer appears as a target whenever it's over a frame header, any part of a frame border, or an icon.

The Console frame completely surrounds the Console window, so you can point anywhere on the border of the frame — not just in the frame header — to bring up the Frame menu. You can prove this for yourself:

- Move the pointer around the edge of the Console frame and notice that it remains a target shape as long as the pointer touches the frame border. (When you want to bring up a Frame menu, just point to the nearest edge of the frame border.)
- Move the pointer slowly across the screen to the right so that it passes over the icons. Whenever the pointer is on an icon, it takes a target shape. This means you can bring up the Frame menu whether the frame is open or closed.
- Point again at the Console frame header, then hold down the RIGHT mouse button. The Frame menu appears, with **Close** at the top and **Quit** at the bottom.
- Keep holding down the RIGHT mouse button and moving the mouse until the item you want is highlighted. This is previewing.

Figure 1-6 *The Frame Menu*

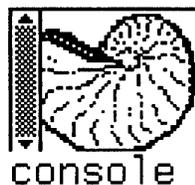
When you bring up a menu, the pointer changes shape again. It's now pointing directly "east."

- Still holding down the mouse button, move the pointer slightly to the right until one of the menu items is highlighted.
- Now move the pointer slowly up and down the list. Different items are highlighted.

To close the Console frame, choose **Close** from the Frame menu:

- Move the pointer over the **Close** menu item to highlight it.
- Release the mouse button. The frame closes to an icon like the one in the figure below.

Did you notice that lines flashed from the corners of the Console frame to the place where the icon is now? You'll see a similar effect when you open an icon — lines temporarily flash from the corners of the icon to the place where the open frame will be. These flash lines will help you locate the frame or icon on a crowded screen.

Figure 1-7 *The Console Frame Closed to an Icon*

The Console icon went to the top of the screen because SunView's default *icon gravity* is north. The icon gravity determines which edge of the screen an icon will "fall" toward when you close its frame. You can change the icon gravity by using the Defaults Editor, described in Chapter 10.

Now bring up the Text Editor's Frame menu. You've already learned how to do this:

- Move the pointer over the Text Editor icon so that the pointer becomes a target.
- *Hold down* the RIGHT mouse button. The Frame menu pops up.

This is the standard way to get a Frame menu. From now on we'll just say, "Bring up the Frame menu." You can do this whether the tool — the Console, the Text Editor, or any other SunView application — is open (in window form) or closed (in icon form).

Toggles

The Frame menu for the Text Editor icon has **Open** at the top, while the Frame menu for the Console frame has **Close** at the top. **Open** and **Close** *toggle* back and forth. You only see one aspect of a toggle at a given time. After you choose it, its opposite is listed in the menu. Toggles are common in SunView in menus and function keys, so you'll be hearing more about them.

Grayed Menu Items

You may have noticed that the **Props** menu item in the Frame menu is in gray, not black, letters. When a menu item is *gray*, it's not available. It won't highlight when you preview it, and you can't choose it.

Props is a feature that hasn't yet been implemented in many SunView tools. Other menu items may be gray because they require a selection and there's no current selection. Table 11-1 in Chapter 11, "All about Tools", contains a list of SunView tools and their corresponding command names. (More on selections later.)

1.4. Using the Text Editor

It's time to learn about text windows. Since all the windows are closed let's open some.

Choosing a Menu Item

It's a good idea to keep the Console window open in case any messages appear, so open that window now:

- Bring up the Console icon's Frame menu. Highlight the item and release the mouse button.
- Choose **Open**.

From now on we'll just say, "Choose **Open** from the Frame menu."

- Now open the Text Editor by choosing **Open** from the Text Editor's Frame menu.

Each frame has its own Frame menu, and each text window has its own Text menu.

With the Console and the Text Editor both open, the screen should look like Figure 1-8, "An Open Text Editor Frame".

The Text Editor Frame

Let's begin by taking a look at how the open Text Editor frame appears on the screen.

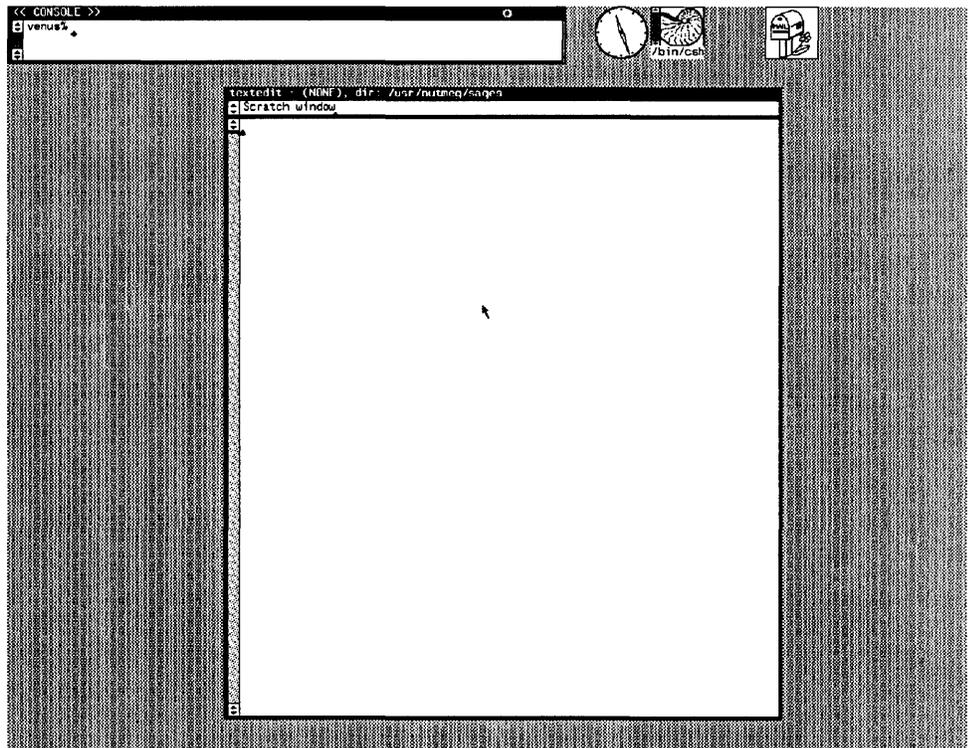
Like the Console, the Text Editor tool has a frame with a frame header at the top. The name of the tool appears on the left side of the frame header. The name shown is the command name by which the tool is known to the SunOS system.

Text Windows

Unlike the Console frame, which contains only one command window, the Text Editor frame contains two *text windows*. Look at the figure below.

- The small *scratch window* just under the frame header is used as a work area for filenames, directory names, and other short text items.
- The large *edit window* below the scratch window is used to display the file you're editing. The text in the edit window is known as the *document*.

Figure 1-8 *An Open Text Editor Frame*



The scratch window and the edit window are both text windows. Each has slightly different properties, but they have a lot in common.

The Text Menu

One thing all text windows have in common is the *Text menu*. Almost everything you learn about working with the Text Editor can be applied to text windows in other tools. The Mail Tool has two or more text windows, the Defaults Editor has one, and so on. Although the Console window is a command window, it also has a Text menu. The Text menu in command windows contains some additional menu items.

If the menu associated with a window looks like the one in Figure 1-9, "Previewing the Text Menu", it's a Text menu. (The resemblance doesn't have to be one hundred percent. The Console window's Text menu contains an extra line, but it's still a Text menu.) All Text menus contain **File⇒**, **Edit⇒**, **Display⇒**, **Find⇒**, and **Extras⇒** menu items.

Previewing a Menu

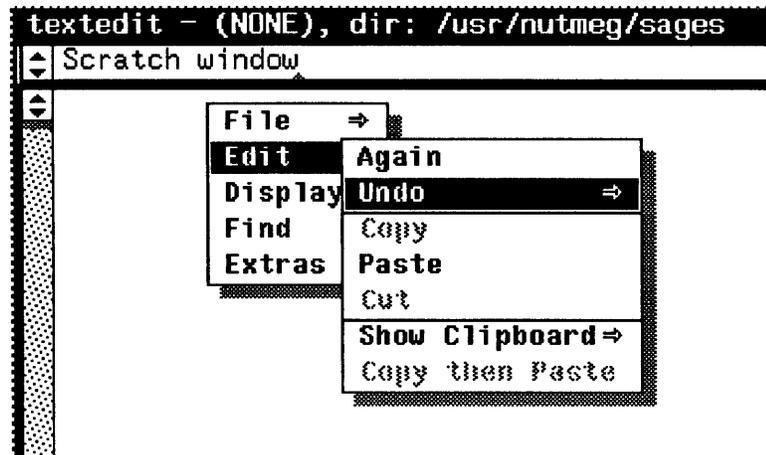
To *preview a menu*, hold down the RIGHT mouse button, then highlight the menu items. When you're finished previewing, either choose a menu item or dismiss the menu.

One reason for previewing is to examine the pull-right menus. A *pull-right menu* is a menu that belongs to a menu item with a pull-right symbol (⇒) at the end of its name, such as **File⇒**.

- To preview a pull-right menu, highlight the pull-right menu item, then move the pointer to the right and highlight its menu items.

Another reason to preview a menu is to find out which menu items require a selection. (More about selections later.)

- Move the pointer into the scratch window and preview its Text menu. Notice that the pull-right menus contain pull-right menus of their own. Preview them too. When you're done previewing, dismiss the Text menu without choosing an item (move the pointer back to the left so that no menu item is highlighted, then release the mouse button).
- Move the pointer into the edit window, preview its Text menu, then release the mouse button.
- Move the pointer into the Console window, preview its Text menu, then release the mouse button. Notice that the Text menu in the Console window has an extra pull-right menu at the bottom, **Comd Modes⇒**.

Figure 1-9 *Previewing the Text Menu*

The Keyboard Focus

It's time to begin typing. But wait a minute — there are three open windows. Where will the typing go?

The window that receives whatever you type is said to have the *keyboard focus*. There are three ways to tell which window has the keyboard focus:

- *The window border thickens.* When a window receives the keyboard focus, its border changes to a bold black line. When the keyboard focus is not in the window, its border is a thin line.
- *The pointer is in the window.* The way you change keyboard focus is to move the pointer. If the pointer is not in a text window — if it's on a frame or on the background or in a window that can't receive text — whatever you type is lost.

Click-to-type is a different way of setting the keyboard focus. See Chapter 4, "All about Keyboards" for more information.

- *The caret blinks.* The *insertion point* — the place in the text window that receives the next character you type — is marked by a small black blinking triangle. This triangle is known as the *caret*. When the keyboard focus is not in the window, its caret appears as an unblinking gray diamond. (This is not true in canvases or tty windows, which are described in Chapter 5, "All about Frames and Windows", in the *Reference* section.)

To see how the keyboard focus works, try these experiments: Before typing, make sure the keyboard focus is in the correct window.

- Move the pointer into the edit window, the large window in the Text Editor frame. Notice that its caret is blinking and its border is black. Be sure the pointer is in a text window before you start typing, or whatever you type will be lost. The pointer can be anywhere in the edit window.

- Type your first name. It appears at the upper left corner of the edit window, because that's where the caret was. The caret moves to a point just after your name.
- Move the pointer into the scratch window. (The pointer is almost as tall as the window — make sure the tip of the pointer is in the window.) The edit window loses its bold border. Its caret turns gray, changes to a diamond, and stops blinking.
- Type your last name, making sure the scratch window has the keyboard focus. Your name appears in front of the words `Scratch window`, with the caret at the end of your name.
- Move the pointer onto the clock icon so that the pointer shape becomes a target. Type your nickname. Look in all the windows and you won't see your nickname anywhere. When no window has the keyboard focus, what you type goes nowhere.

The Delete Key

It's time to type something else in the edit window, so let's get rid of what's there now. There are several ways to do this. (There are several ways to do almost everything in SunView. This makes it nice to work with because you get to use it just the way you prefer.)

`Delete` is the default key for erasing the character to the left of the insertion point. You'll find it on the right side of the alphanumeric part of the Type4 keyboard, above the `Return` key. Other keyboards have a `DEL` key — it's the same thing.

If you're not sure which keyboard you have, see Chapter 4, "All about Keyboards", for drawings of the various keyboards and instructions on how to label a Type2 or Type3 keyboard to match the Type4 keyboard references in this manual.

- Move the pointer into the edit window, where you typed your first name a few minutes ago.
- Press the `Delete` key once, and the last letter of your name disappears. Unless your name has only one letter, press the `Delete` key again until all of the name is erased.

You can change the backspacing key from `Delete` to `Back Space` with the Defaults Editor, described in Chapter 10 in the *Reference* section.

- Now erase your last name from the scratch window. Hint: Remember to move the keyboard focus first.

The File⇒ Pull-Right Menu

When you previewed the Text menu, you discovered that a pull-right menu can contain lower-level pull-right menus. The `File⇒` pull-right menu is three levels deep. It's called the `File⇒` pull-right menu because you bring it up by pulling the pointer to the right of the `File⇒` menu item in the Text menu.

Here's a way to use the **File**⇒ pull-right menu to erase all the text in a window.

- With the pointer in the edit window, type your name again. Now you're going to erase it by choosing a menu item.
- Bring up the Text menu.
- Highlight the **File**⇒ menu item by moving the pointer onto it.
- Move the pointer slightly to the right. The **File**⇒ pull-right menu appears, with **Save as New File** at the top and **Finishing Up**⇒ at the bottom.
- Choose **File**⇒**Empty Document**, as shown in the figure below. **File**⇒**Empty Document** is shorthand for "the **Empty Document** menu item in the **File**⇒ pull-right menu."

You choose a menu item in a pull-right menu the same way you choose **Open** and **Close** from the **Frame** menu: by moving the pointer over the item so that it's highlighted in inverse video, then releasing the mouse button.

Figure 1-10 *The File⇒ Pull-Right Menu*

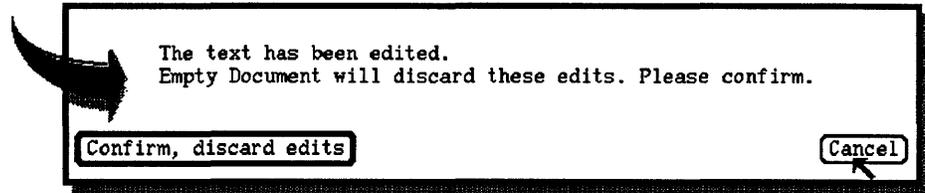


Empty Document tells the Text Editor to give you a fresh start:

- If you're creating a new file, the Text Editor throws away the editing you did in memory.
- If you're working on an existing file, the Text Editor throws away your edits (since the last time you saved the file) and forgets the file name if you make that choice. The file itself is still on the disk — **Empty Document** doesn't remove it.

Alert Windows

When you choose **Empty Document**, and have an existing file in the Text window, the Text Editor alerts you to the fact that you have made some changes in the text window that you haven't saved. It beeps and brings up the following message:

Figure 1-11 *An Alert Window*

The arrow alerts you to read what's in the box. The lozenge-shaped *buttons* at the bottom of the alert window give you the opportunity to choose an action. All alert windows in SunView work this way.

- The **Confirm, discard edits** button tells the Text Editor to empty the window and forget the file without saving the changes you've made. This button, the *yes* choice, has a double outline. The pointer moves automatically to the button with the double outline to make it easy to respond *yes*.
- The **Cancel** button tells the Text Editor to cancel the **Empty Document** request. This gives you an opportunity to save your changes.

You choose an action in an alert window by *pushing a button*. Since you don't want to keep your name, push the **Confirm** button.

Here's how to push a button in an alert window:

- Move the pointer over the button you want to push. In this case, **Confirm, discard edits** is the *yes* choice, so the pointer has already been automatically moved there.
- Click the LEFT mouse button. The alert window disappears, and your name is gone from the text window.

1.5. Selecting Text

You've learned how to *erase* text. There are also ways to *delete* text. The difference is that deleted text is saved on the Clipboard. (More about the Clipboard in a few pages.)

You can delete a selection by using:

- The **Edit**⇒ pull-right menu (described below).
- The **(Cut)** function key (described in the next lesson).
- An *accelerator* — a menu choice made from the keyboard. (These useful shortcuts are described in the *Reference* section in Chapter 9, "All about Accelerators").

We can't really discuss selections until there's some text to select, so let's enter some.

- Move the pointer into the edit window and type the following limerick. Press **[Return]** at the end of each line. Press **[Tab]** to begin an indented line.

```
All Hail, Electronic Mail!

There once was a workstation fan
Who networked, but only on LAN--
    Till he chanced to discover
    An Ethernet lover--
Now he's linked by e-mail to Japan.
```

Don't worry if you made some typing errors. You're about to learn how to make editing corrections. When you're done, the edit window looks something like the following figure.

Figure 1-12 *The Electronic Mail Limerick*

```
textedit - (NONE), dir: /usr/nutmeg/sages
Scratch window
All Hail, Electronic Mail!

There once was a workstation fan
Who networked, but only on LAN--
    Till he chanced to discover
    An Ethernet lover--
Now he's linked by e-mail to Japan.
```

Selecting a Word

The caret (the blinking black triangle) is now positioned after the period following the word Japan. Whenever you enter text from the keyboard, the caret moves to a point just after the last character you entered.

Suppose you want to change Japan to Siam. You can press the **[Delete]** key repeatedly until Japan is deleted and then type Siam. But that won't work if you want to change a word at the beginning of the text.

You can change any part of the text by selecting it, deleting it, and typing the replacement. To select a word, you move the pointer over the text and click **LEFT** twice.

To change a word anywhere in the text:

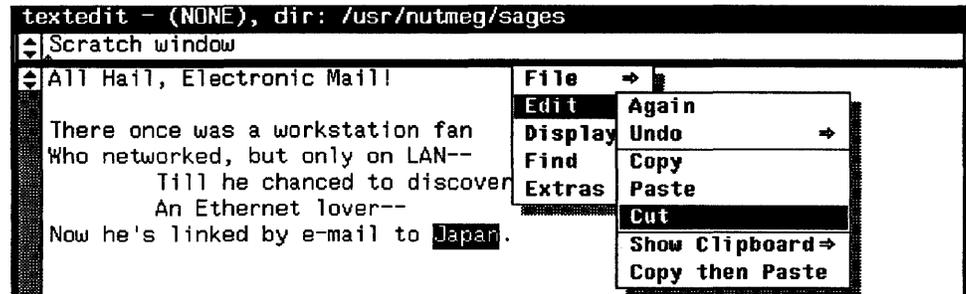
- Move the pointer so that the head of the arrow is anywhere in the word.
- *Double-click* the **LEFT** mouse button to *select the word*. The selection is highlighted.

Double-click means click twice in rapid succession. You *click* to select a character, *double-click* to select a word, *triple-click* to select a line, and *quadruple-click* to select all the text in a document.

- Bring up the Text menu and choose Cut from the Edit⇒ pull-right menu. The word is removed from the window.
- Type the replacement.

Now try replacing Japan with Siam by following the instructions you just read.

Figure 1-13 *Selecting and Cutting a Word*



Extending the Selection

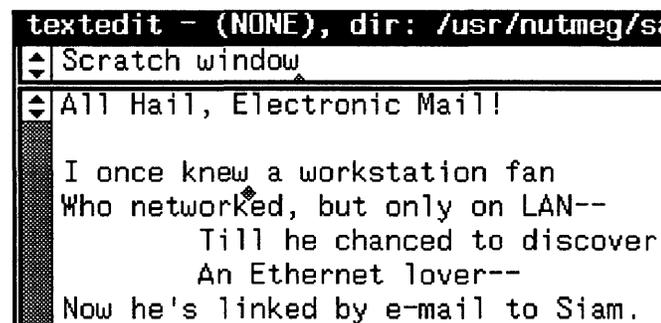
The MIDDLE mouse button is used to *extend* a selection by adjusting its length. If you selected a *word* by double-clicking the LEFT mouse button, the selection is extended in units of a word. If you selected a *line* by triple-clicking the LEFT mouse button, the selection is extended in units of a line.

To get an idea of how extending a selection works:

- Select the word *There* at the beginning of the limerick by double-clicking.
- Move the pointer to anywhere in the word *was* and click the MIDDLE mouse button once. The selection *There once was* is highlighted.
- Delete the selection by choosing **Cut** from the **Edit**⇒ pull-right menu.
- Type *I once knew* to replace the deleted text. The result is shown in the following figure.

The MIDDLE mouse button is used to extend a selection.

Figure 1-14 *Result of Cutting and Typing*



You can also extend a selection by *wiping* the pointer through the text in either direction while holding down the MIDDLE mouse button. The adjustment becomes final when you release the mouse button. You can adjust a selection as often as you like by wiping or clicking MIDDLE.

More About Selections

When you move the insertion point by pointing and clicking LEFT, you automatically select a single character.

- Try clicking LEFT various places in the text. A single character is highlighted each time. The character can be a space, a Return character (which looks like a space), or a Tab character (which looks like several spaces).

Whether you select a character, a word, or a line, the caret appears at the end closest to where you clicked. To see how this works:

Where you click determines which end of the selection gets the caret.

- Select the word `workstation` by pointing to the beginning of the word and double-clicking LEFT. The caret appears before the `w`.
- Select the word `networked` by pointing to the end of the word and double-clicking LEFT. The caret appears after the `d`.

Imagine a vertical line precisely in the center of the text to be selected. If you click anywhere to the right of the center, the right end of the selection gets the caret, and vice versa.

- Try double-clicking LEFT in a word and see how close you can get to the center before the caret switches sides.
- Try the same experiment with a single character by single-clicking LEFT.

There are also some very useful editing shortcuts that work with *secondary selections*, which are temporary selections you make while holding down a function key. They're covered in the *Reference* section in Chapter 6, "All about Selections".

Inserting Text

If you're just going to delete the word you selected, you won't care which end the caret is on. But if you're going to insert text instead of deleting, you can use the position of the caret to help you.

The *caret* shows the position of the *insertion point*.

To move the insertion point to the beginning of the line, you can carefully move the pointer in front of the first word and click, or you can point anywhere in the first half of the line and triple-click, which may seem a lot easier. Try it both ways and see which you prefer.

Change the first line of the limerick to read, `Gee, I once knew a workstation fan:`

- Move the pointer somewhere near the beginning of the first line and triple-click the LEFT mouse button.
- Type `Gee, .`

In SunView, you can choose the way that's easiest for you. Once you've used SunView for a while, you may decide to change to a different style of working. That's why it's a good idea to read the *Reference* section and become familiar with all the options available to you.

Extending the Selection Backward

You've seen how to extend a selection forward, but you can extend it backward just as easily.

- Double-click **LEFT** to select the word `fan` in the first line of the limerick.
- Move the pointer to the word `workstation` and click **MIDDLE** to extend the selection backward. The phrase `workstation fan` is highlighted.

More About the Edit⇒ Pull-Right Menu

You've seen how to choose **Edit⇒Cut** to delete text. **Copy** and **Paste** are related to **Cut**. They all use the Clipboard.

The Clipboard

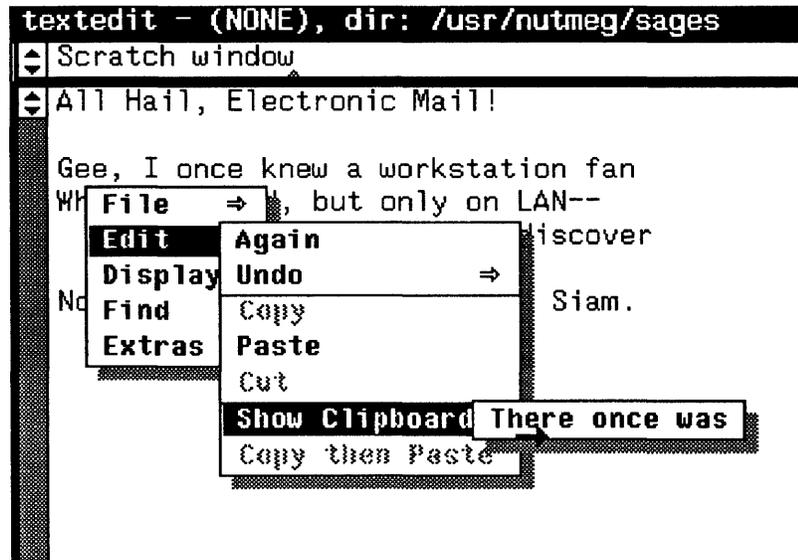
The *Clipboard* is a temporary storage area where text windows place text that you copy or cut. Remember the way your typing was lost when there was no keyboard focus? When you delete text by choosing **Cut** from the **Edit⇒** pull-right menu, it isn't lost. Instead, the deleted text is placed on the Clipboard.

If the text on the Clipboard is long, **Edit⇒Show Clipboard** shows only the first several characters.

- To look at the Clipboard, preview **Show Clipboard⇒** from the **Edit⇒** pull-right menu. (You don't actually have to choose **Show Clipboard⇒** by releasing the mouse button in order to see the Clipboard contents. You can simply preview the contents by pulling right on the menu item.)

Show Clipboard⇒ displays the last selection you cut or copied. If you've been following the lesson, you should see the words `There once was`, as in the following figure.

Figure 1-15 Showing the Clipboard



To put text on the Clipboard, select the text, then choose **Edit⇒Copy**:

- Select the word Ethernet.
- Choose **Edit⇒Copy**. The selection is copied onto the Clipboard.
- Now choose **Edit⇒Show Clipboard⇒**. There it is.

Copy, Paste, and Cut

Edit⇒Copy puts the selection on the Clipboard. **Edit⇒Cut** deletes the selection and puts it on the Clipboard. **Edit⇒Paste** inserts the contents of the Clipboard at the insertion point.

To move text from one place to another:

- Select the text.
- Choose **Edit⇒Cut**. Now the deleted text is on the Clipboard.
- Click LEFT to set the insertion point where you want the text moved to.
- Choose **Edit⇒Paste**. The text on the Clipboard is pasted at the caret.

The first line of the limerick contains the phrase I once knew. Change it to once I knew:

- Cut once . (Include the space after once by extending the selection.)
- Paste in front of I knew. Now the line reads, Gee, once I knew....

You can paste repeatedly from the Clipboard.

Now paste the contents of the Clipboard before the word knew so the line reads Gee, once I once knew

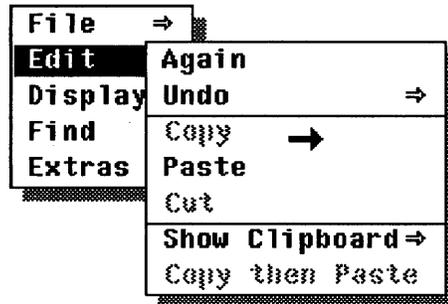
- Move the caret to the right of the space between I and knew.
- Paste.

The Clipboard won't change until you choose **Cut** or **Copy** again.

Some Menu Items Require a Selection

Open the **Edit**⇒ pull-right menu. Notice that **Copy** and **Cut** appear in light gray letters. And they're not highlighted when you preview the menu. Usually a menu item is grayed because it requires a selection and there isn't one currently. For example, after a **Paste** operation there's no current selection.

Figure 1-16 *Copy and Cut Require a Selection*



Select some text — you might try quadruple clicking to select the entire limerick — and preview the **Edit**⇒ pull-right menu again. Now **Copy** and **Cut** are available because there's a selection.

1.6. Saving a New File

Now that you've made a wonderful limerick even better, wouldn't you like to keep it so you can read it for inspiration on stressful days?

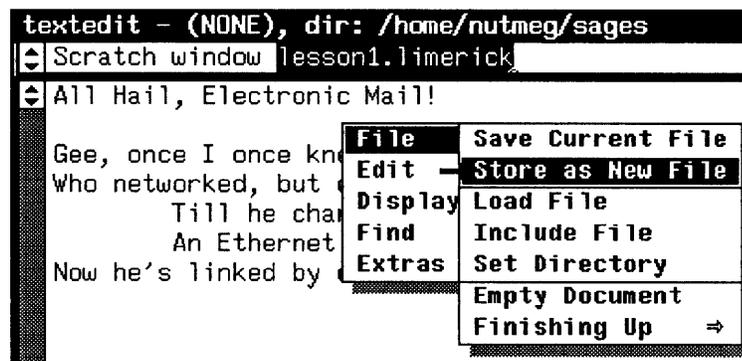
To save the contents of a text window:

- Type the filename of the new file. Let's name it `lesson1.limerick`. The Text Editor's scratch window is handy for that purpose, as shown in the following figure.
- Select the filename you typed. You'll need to extend the selection with the MIDDLE mouse button to select the whole name.

NOTE: Be sure to move the pointer back into the edit window. If the pointer is in the scratch window, the contents of the scratch window, not the edit window, will be saved. That's probably not what you want.

- In the edit window, bring up the Text menu and choose **File**⇒**Store as New File**.

Figure 1-17 *Saving a New File*



Look at Figure 1-18. The name of the file you saved now appears on the left side of the frame header. This is the file your text will be saved to when you choose **Save Current File** later in this lesson.

Figure 1-18 *Filename in the Frame Header*

```

textedit - lesson1.limerick, dir: /home/nutm
Scratch window lesson1.limerick
All Hail, Electronic Mail!
Gee, once I once knew a workstation fan
Who networked, but only on LAN--
    Till he chanced to discover
    An Ethernet lover--
Now he's linked by e-mail to Siam.

```

About the Directory

On the right side of the frame header in the preceding figure, you see `dir:` followed by the name of the directory where the limerick file was saved. That's the *default directory* for the Text Editor frame. The Text Editor uses the default directory when you load or store a file with a partial pathname.

On your workstation, the default directory is your own home directory. If you want to save the file to another directory, you can do it in one of two ways:

- You can type the full pathname of the file in the scratch window, beginning with a slash (/), and select that pathname for **Store a New File**. If you do this, the default directory will still be your home directory.
- If you want to change the default directory, you can select a new directory name and choose **File⇒Set Directory**. The new name will appear on the right side of the frame header.

For more information about setting the directory, see the **File⇒Set Directory** menu item in Chapter 8, "All about Menus".

Undoing Edits

The Text Editor lets you know when the current file has been edited.

- Select `All Hail` in the limerick title.
- Choose **Edit⇒Cut**.
- Type `Oh Swell` to replace words you cut.

The frame header now contains the word `(edited)` after the filename.

As soon as you change the text in a text window after saving the file, the frame header shows that the file has been edited as shown in the following figure.

NOTE: This can serve as a reminder to save your file. It's a good idea to do that fairly often.

Figure 1-19 *The File Has Been Edited*

```

textedit - lesson1.limerick (edited), dir:
Scratch.window lesson1.limerick
Oh Swell, Electronic Mail!
Gee, once I once knew a workstation fan

```

The Text Editor (or any application that uses the Text menu) lets you change your mind by undoing edits.

To remove the words you just typed:

- Choose **Undo**⇒**Undo Last Edit** from the **Edit**⇒ pull-right menu. The line disappears. The word (edited) in the frame header also disappears.

Undo Last Edit undoes the most recent edit action. It can undo:

- **Paste, Cut, or Again** in the **Edit**⇒ pull-right menu.
- Typing, including pressing the **Delete** key. Everything you typed since the last selection is undone. (Remember, clicking to set the insertion point makes a one-character selection.)

CAUTION It's possible to accidentally undo a great deal of typing at one fell swoop, so be careful. You can't undo Undo!

1.7. The SunView Menu

The location of the pointer determines which menu comes up when you hold down the RIGHT mouse button. If the pointer is in a text window, the Text menu comes up. If it's on a frame or an icon, the frame menu comes up. If it's on the gray SunView background, the SunView menu comes up.

Figure 1-20 *The SunView Menu*

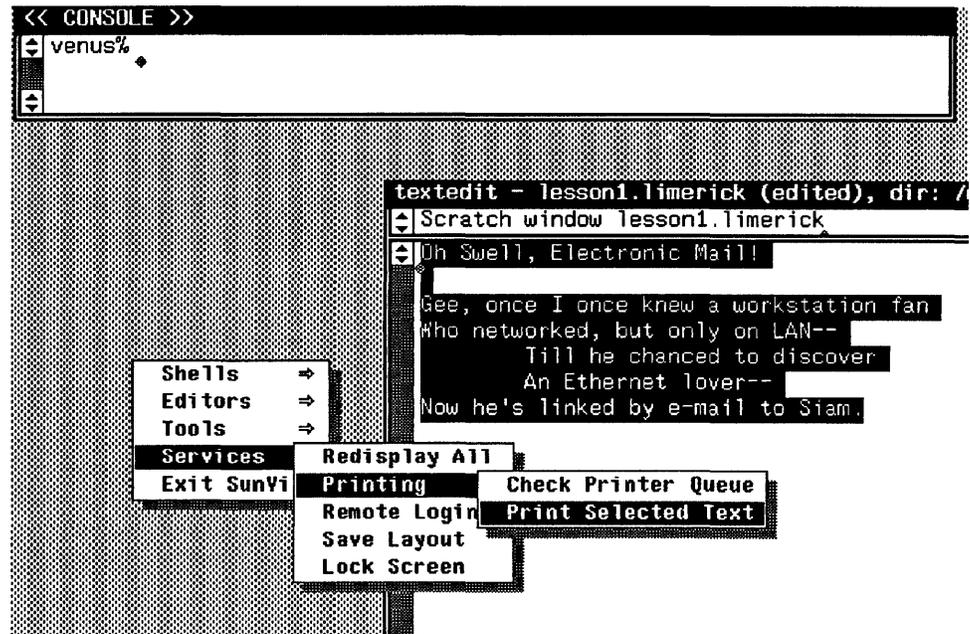
The SunView menu lets you start all the tools you now have — Mail Tool, Text Editor, Command Tool, Console, and Clock — and provides several other handy tools and services.

NOTE: You should never have more than one Console tool running.

Printing from the SunView Menu

One of the services in the SunView menu gives you an easy way to print the selection.

Figure 1-21 *Printing the Selection*



To print the limerick:

- Select the limerick by quadruple-clicking in the edit window. The entire document is highlighted.
- Choose **Print Selected Text** from the **Services**⇒**Printing**⇒ pull-right menu. Selection printed will appear in the Console window — don't make another selection until you see this message.
- Go to your local printer and pick up your poem. (If you don't know which printer has been assigned to your workstation, ask your system administrator.)

Locking the Screen

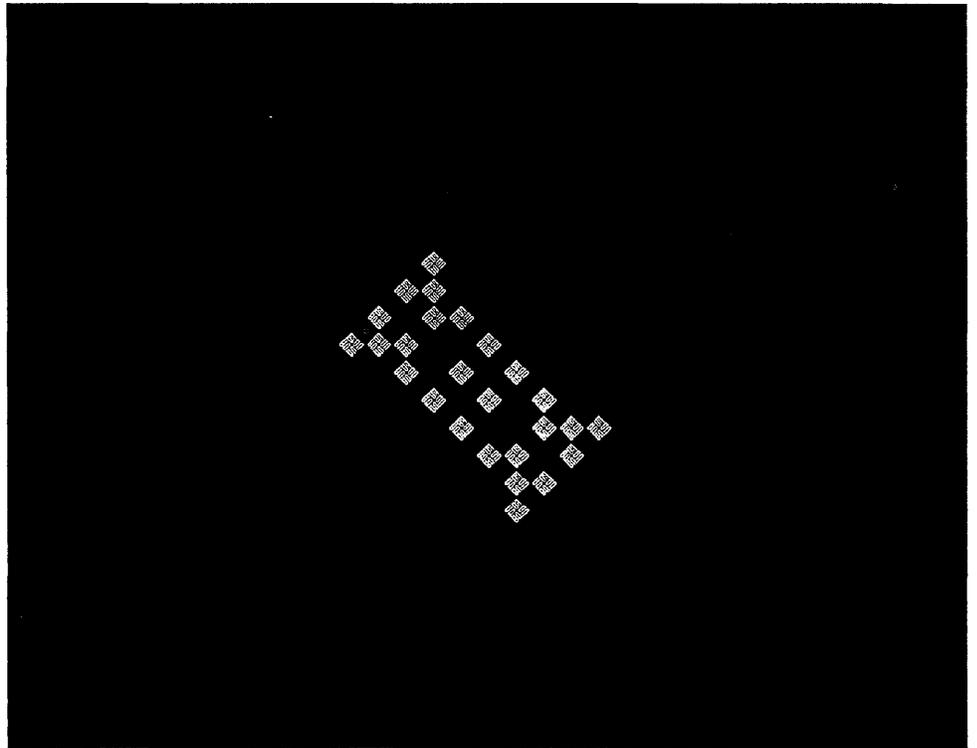
Ready to take the limerick and run? Before you head out the door, there's just one more thing to do. If you leave your workstation screen up for long periods of time without changing the display, the image will "burn" into the tube and make the screen hard to read. That's one reason why it's a good idea to lock the screen before leaving work. **Lock Screen** displays a moving image that protects the screen.

Another reason to lock the screen is to protect your privacy. You can leave all your frames open in SunView, and **Lock Screen** will keep idle passersby from reading their contents.

When you're ready to end this lesson:

- Bring up the Text Editor's Text menu and choose **File⇒Save Current File** from the **File⇒** pull-right menu. Always save your work when you're going to be away from your workstation for a while.
- Choose **Empty Document** from the **File⇒** pull-right menu. This isn't usually necessary, but it gives you a clean start for the next lesson.
- Choose **Close** from the Frame menu to prepare for the next lesson. Normally you would keep frames open.
- Bring up the SunView menu.
- Choose **Lock Screen** from the **Services⇒** pull-right menu. Your screen gets dark and a moving image appears. Look at the example in the following figure.

Figure 1-22 *The Locked Screen*



At the beginning of the next lesson, you'll learn how to unlock the screen.

SunView Accelerated

In the first SunView lesson, you learned how to perform window operations using menus. This lesson teaches you how to use accelerators that make SunView operations faster and easier. You will learn how to:

- Use SunView shortcuts.
- Load a file in the Text Editor.
- Make frames larger or smaller and move them around.
- Look at different parts of a document in a window by scrolling.
- Find and replace text.
- Use panels to make choices and perform actions.

2.1. Introduction to Accelerators

Many SunView actions can be initiated in more than one way. Some alternative methods are known as *accelerators* because they're faster than the standard technique. There are accelerators that let you perform menu operations from the keyboard, accelerators that let you move the caret from the keyboard, and accelerators that use the mouse and the keyboard instead of a menu.

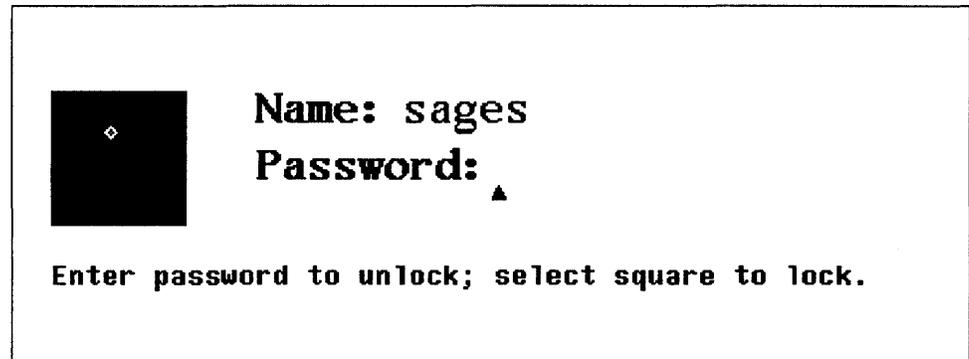
You can start using a few accelerators right away and add more to your repertoire as you become accustomed to SunView. In this chapter you'll learn how to use a few of the powerful accelerators that are available. There are many more, and they're covered in the *Reference* section in Chapter 9, "All about Accelerators".

2.2. Unlocking the Locked Screen

You ended the first lesson by locking the screen to ensure privacy while preserving the state of your windows. To unlock the screen:

- Click any mouse button or press any key except **F1** or **Caps**. The screen changes from black to white and a password prompt appears, as shown in the following figure.

Figure 2-1 *Unlocking the Screen*



NOTE: If you change your mind and want to relock the screen, click **LEFT** in the black box.

- To unlock the screen, type your password, then press the **Return** key. The words `Validating login...` appears.

If your password was correctly typed, the SunView screen appears, just as you left it. If you made a typing error, Lock Screen gives you another chance to enter the correct password.

2.3. Working with Several Frames

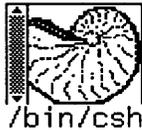
In the previous lesson, you had two frames open at the same time: the Console frame and the Text Editor frame. They didn't overlap, and the Console frame was up in the corner, out of the way. Often in SunView you will have several frames open at the same time, and working with them requires some window management techniques.

Let's get some frames open and work with them.



- Move the pointer to the Text Editor icon.
- Choose **Open** from its Frame menu.

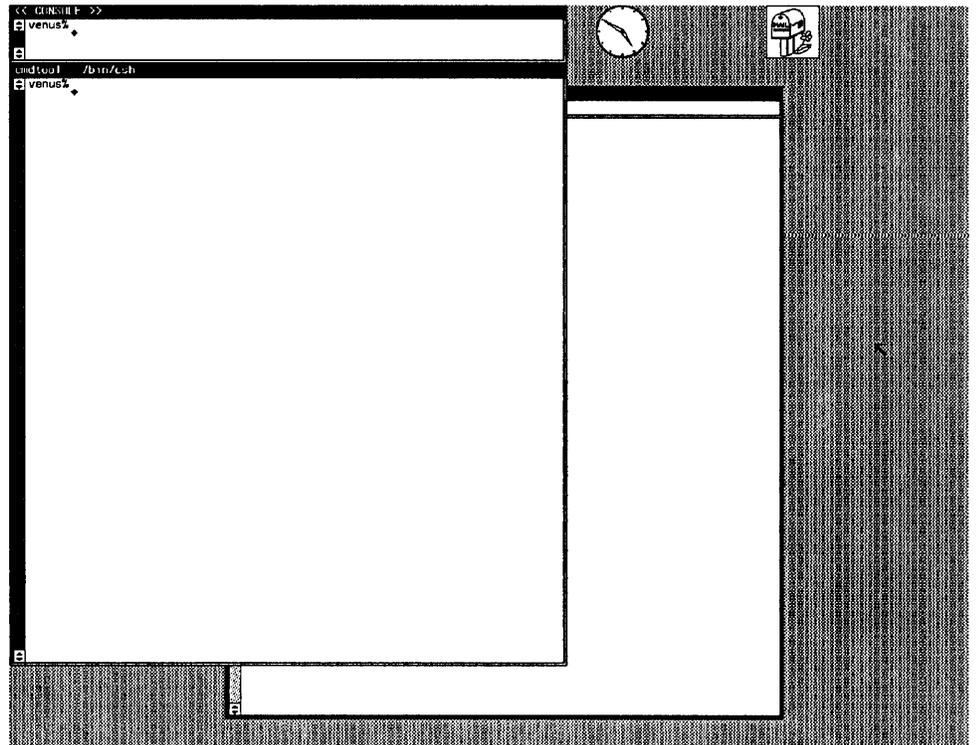
Now open the Command Tool using *a point-and-click accelerator for Open*:



- Move the pointer to the Command Tool icon.
- Click the LEFT mouse button. You can open any frame by clicking LEFT on its icon.

The SunView screen now contains three open frames, as shown in the following figure: the Console frame, a Text Editor frame, and a Command Tool frame.

Figure 2-2 *Three Open Frames*



The Command Tool frame overlaps the Text Editor frame. You can type in any open frame regardless of whether it's overlapped, by moving the pointer into its window.

- Move the pointer into each of the open windows, including the Text Editor's scratch window, one at a time. Notice how the keyboard focus moves with the pointer — the window border darkens — even if the window is partly hidden.

“Back” and “Front”

The **Back** and **Front** menu items in the Frame menu let you manage the overlap of frames.

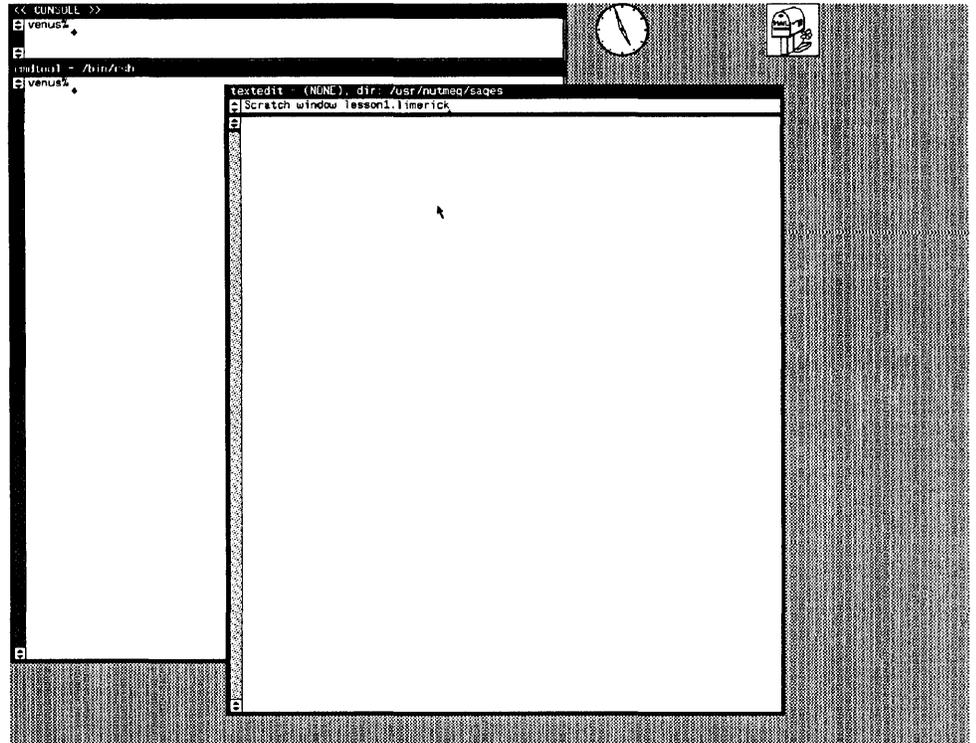
- **Back** hides the overlapping part of a frame by moving the frame to the back.
- **Front** exposes the overlapped part of a frame by moving the frame to the front.

To change the overlap of a frame, move the pointer onto the frame border and choose **Back** or **Front** from the Frame menu.

NOTE: It's possible for a frame to be completely hidden. To rescue a small frame from the back, send all the visible frames to the back by choosing **Back** for each one, and the hidden frame will appear.

Try exposing the Text Editor frame, which is partly hidden:

- Move the pointer onto the frame border of the Text Editor frame.
- Choose **Front** from the Frame menu. The overlap changes, as shown in the following figure.

Figure 2-3 *Exposing a Partly Hidden Frame*

You could have performed the same action by choosing **Back** from the Command Tool's Frame menu. If you have several layers, though, keep in mind that **Front** sends a frame all the way to the top, and **Back** sends it all the way to the bottom.

Function Keys

At the left side of your keyboard are the ten main SunView *function keys*. These keys act as accelerators that enable you to perform various menu operations without opening a menu. Some additional keys on the top and right side of the keyboard perform special functions in SunView; they're described in Chapter 9, "All about Accelerators".

Table 2-1 *The Left Function Keys*

Key	Menu Item	Menu
Stop	none	—
Props	Props	Frame
Front	Front, Back	Frame
Open	Open, Close	Frame
Find	Find⇒Find Selection⇒Forward Find⇒Find Clipboard⇒Forward	Text Text
Again	Edit⇒Again	Text
Undo	Edit⇒Undo⇒Undo Last Edit	Text
Paste	Edit⇒Paste	Text
Copy	Edit⇒Copy	Text
Cut	Edit⇒Cut	Text

Among the function keys described in Table 2-1, “The Left Function Keys”, you’ll find accelerators for **Copy**, **Paste**, and **Cut** as well as **Open** and **Front**. Some of the function keys accelerate the **Edit⇒** and **Find⇒** pull-right menus in the Text menu, while others accelerate the Frame menu. The keys are marked with the names of the lowest-level menu items.²

Back and Front Accelerators

Pressing a function key is the equivalent of opening a menu and choosing an item. Some function keys *toggle* just as some menu items toggle. The **Open** function key toggles to Close, and the **Front** function key toggles to Back.

- Move the pointer anywhere in the Text Editor frame (not just the frame border) and press the **Front** function key. The Text Editor frame moves to the back, behind any overlapping frames.
- Press **Front** again with the cursor in the Text Editor window. The Text Editor frame comes forward again.

Moving and Resizing Frames

Move⇒ and **Resize⇒** in the Frame menu let you change the location and size of a frame. You’ll learn how to do this from the menu and by using accelerators.

When you move or resize a frame, an empty rectangle known as the *bounding box* shows you the new position. As long as you continue to hold down the mouse button, you can continue to adjust the position of the bounding box. The frame moves to the position of the bounding box when you release the mouse button.

² There are two reasons why your function keys may not look or work as described in this section: either you don’t have a Type4 keyboard, or your function key functions have been remapped (customized).

Chapter 4, “All about Keyboards” shows you the differences among Type2, Type3, and Type4 keyboards, and tells you how to customize — or uncustomize — your keyboard. Left-handed people can customize the keyboard layout so that the functions can be accessed from the right side of the keyboard.

Unconstrained is the default in the **Move**⇒ and **Resize**⇒ pull-right menus. Lets you adjust both dimensions, height and width, or left and right. In unconstrained resizing or moving, the corner or side you have the pointer near becomes the portion of the frame that will be changed. You move the pointer near a part of the frame border or corner and, holding down the MIDDLE mouse button, make the change.

A **Constrained** resize lets you adjust only one dimension, either height or width. A **Constrained** move only lets you change the location either left-to-right or up-and-down. The direction the frame moves depends on where you position the pointer. If you press MIDDLE and the pointer is in the frame on a side, the frame moves sideways. If you press MIDDLE and the pointer is in the frame on the top or bottom edge, the frame moves up and down.

2.4. Rearranging the Screen

Here are the general rules for resizing a frame:

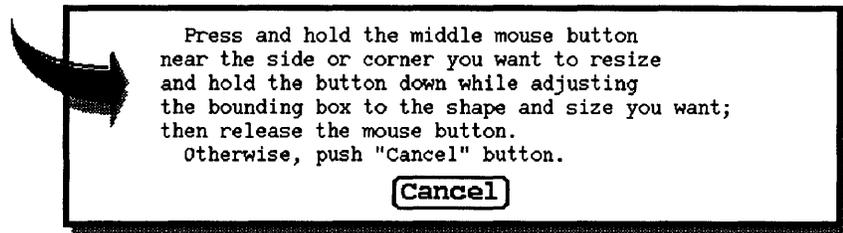
- Choose **Unconstrained** (the default) or **Constrained** from the **Resize**⇒ pull-right menu in the Frame menu.
- Move the pointer near the frame border or corner, then hold down the MIDDLE mouse button.
- Stretch or shrink the bounding box with the pointer while continuing to hold down the MIDDLE mouse button. (Reminder: The MIDDLE mouse button is for *extending*, *adjusting*, and *moving* SunView objects.)
- Release the mouse button when is the way you want it.

Step One

Let's rearrange the screen so that it looks like the one shown a few pages ahead in Figure 2-8, "Rearranging the Screen—Final Step". You can try this on your own, or you can follow the detailed instructions below.

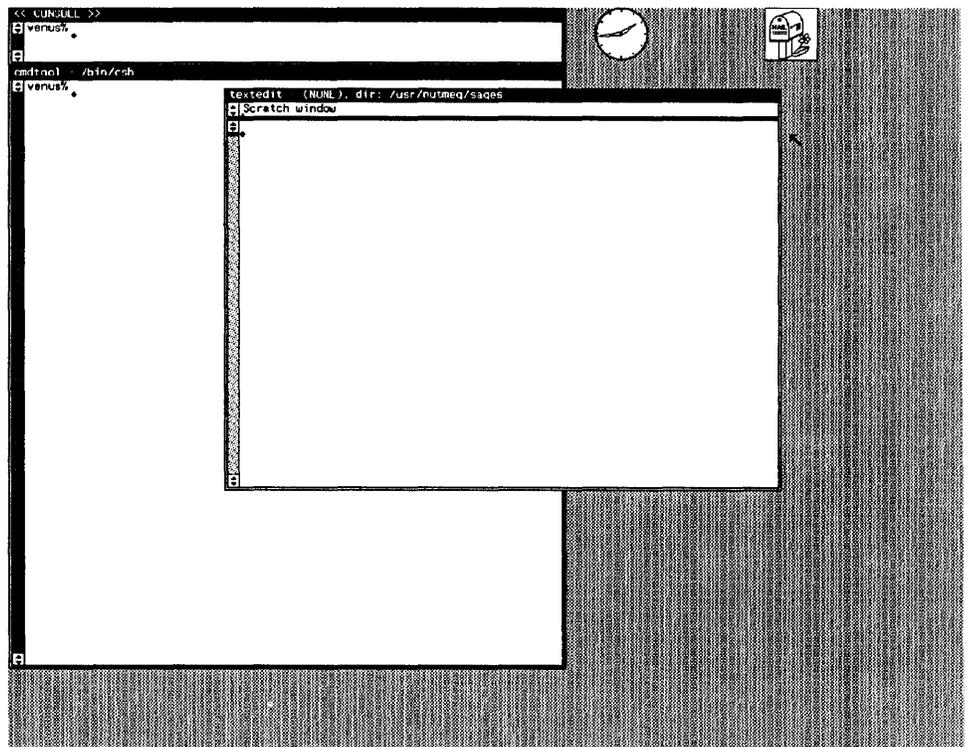
- Choose **Resize⇒Constrained** from the Text Editor's Frame menu. The following alert window appears:

Figure 2-4 *Resize⇒Alert Window*



- Following the instructions in the alert window, place the pointer near the bottom edge of the Text Editor frame and shrink it by about one-third, as shown in the following figure.

Figure 2-5 *Rearranging the Screen—Step 1*



Step Two

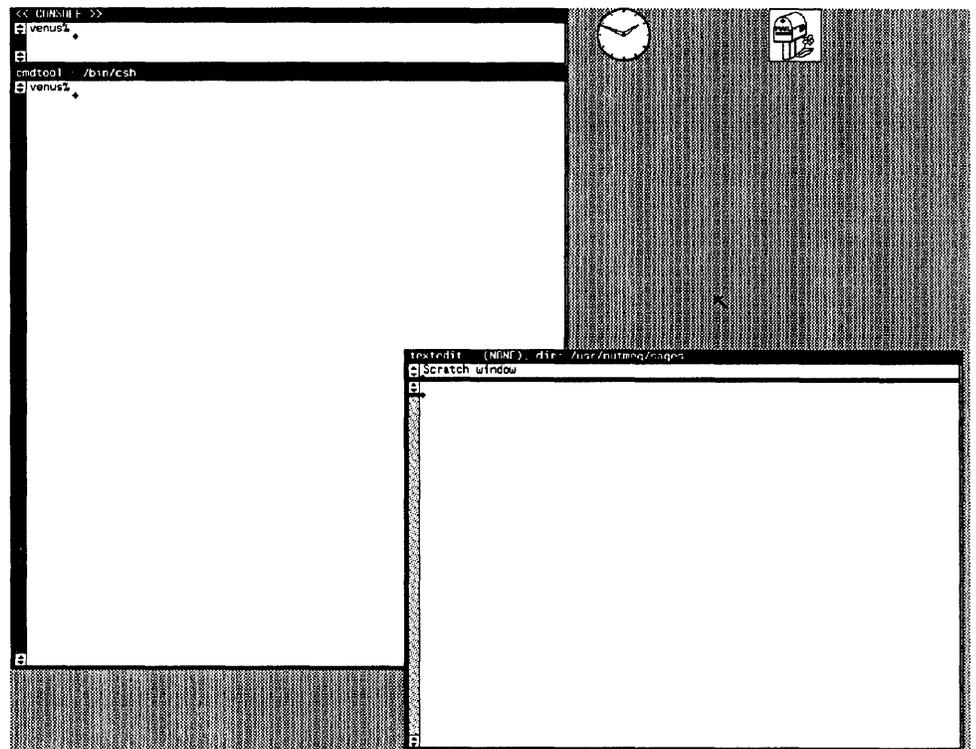
To move an icon or open frame to a new position on the screen:

- Choose **Move⇒Unconstrained** or **Move⇒Constrained** from the Frame menu.
- Move the pointer onto the icon or near a side or corner. Hold down the MIDDLE mouse button and move the bounding box.
- When the bounding box is in the location where you want the frame or icon to be, release the mouse button.

Here's the second step in detail:

- Choose **Move⇒** from the Text Editor's Frame menu. (**Unconstrained** is the default.)
- Follow the instructions in the alert window. Move the pointer near a corner or a side, or on top of an icon.
- Hold down the MIDDLE button and move the bounding box down to the bottom right corner of the screen, as shown in the following figure. Then release the button.

Figure 2-6 *Rearranging the Screen—Step 2*



Step Three

The accelerator for **Move**⇒ is to place the pointer on the frame and drag the frame while holding down the MIDDLE mouse button.

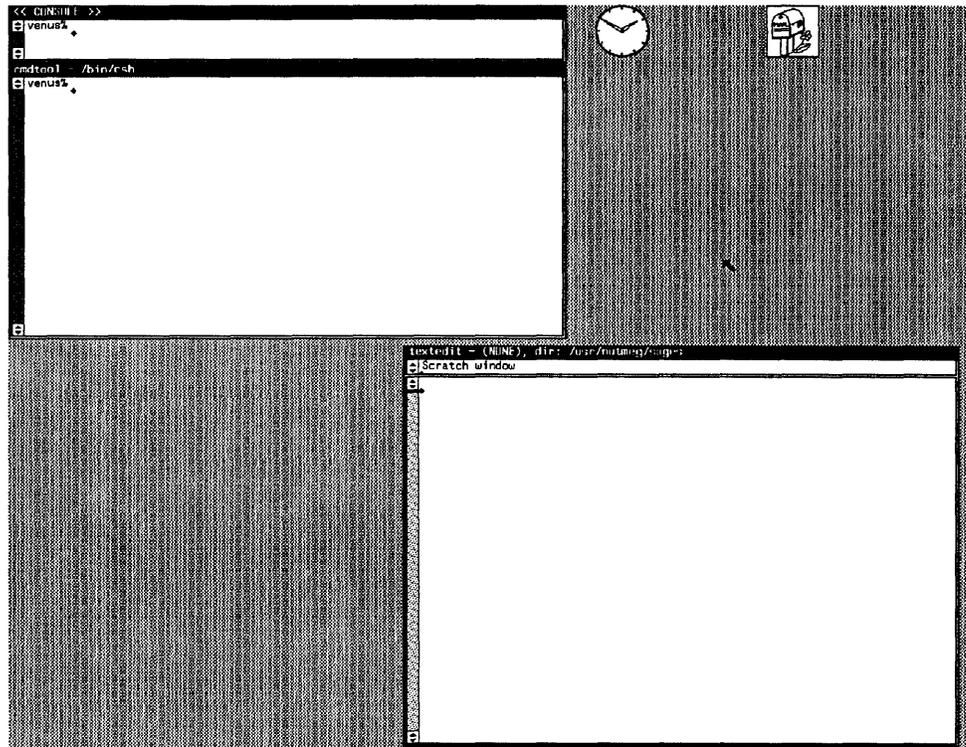
The accelerator for **Resize**⇒ is to place the pointer on the frame and drag the frame while holding down the **Control** key and the MIDDLE mouse button.

NOTE: Moving and resizing with accelerators is constrained if you grab the frame border near the middle of a side, unconstrained if you grab it near a corner.

Now let's resize and move the Command Tool frame, using accelerators instead of the Frame menu:

- Hold down the **Control** key while you resize the Command Tool frame.
- Move the pointer to the middle of the bottom edge of the Command Tool frame and hold down the MIDDLE mouse button to grab the frame.
- Drag the bottom edge upward until it's above the Text Editor's frame header, as shown in the following figure.
- Release the **Control** key and the mouse button.

Figure 2-7 *Rearranging the Screen—Step 3*

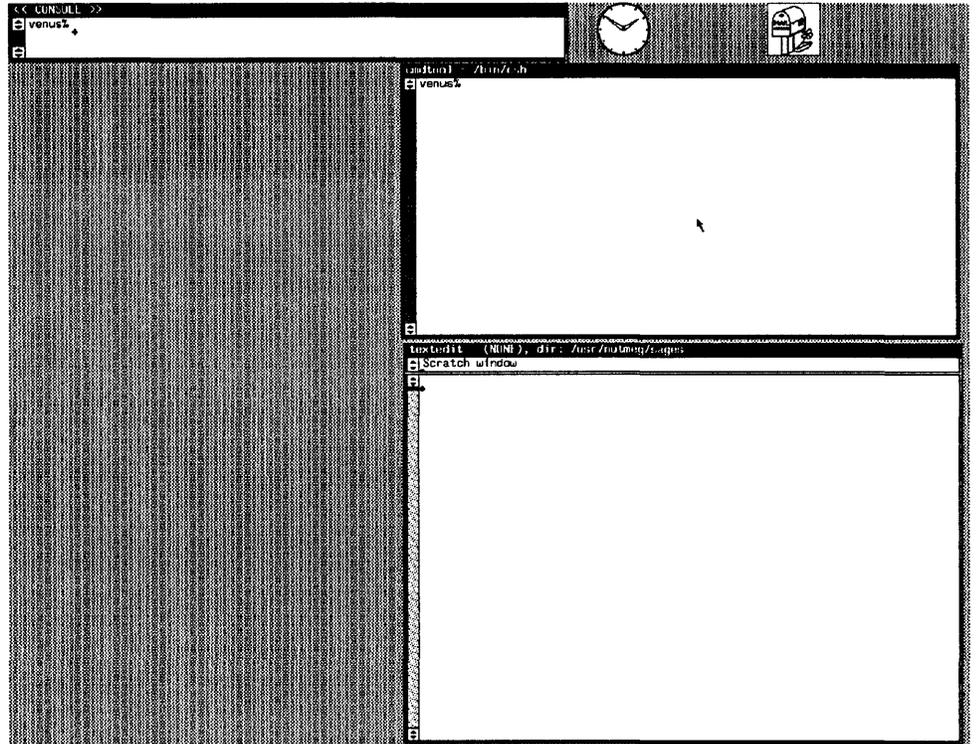


Final Step

All that's left to do now is to use an accelerator to move the Command Tool frame to its final position.

- Move the Command Tool frame to the right by grabbing the middle of the right border and dragging to the location shown in the following figure.

Figure 2-8 *Rearranging the Screen—Final Step*



One advantage of the new arrangement is that the windows no longer overlap. This means less redrawing of the screen, therefore faster response time. In addition, you now have room for another frame or two. As you gain confidence in using SunView, you'll often find it handy to work with several open frames.

Moving Icons

You can't resize icons, but you can move them just the way you move frames.

- For practice, try moving the Clock icon around with the Move accelerator. You may decide you'd prefer to keep it in another location.

Zoom and FullScreen

There are two additional menu items in the **Resize**⇒ pull-right menu: **Zoom** and **FullScreen**. **Resize**⇒**Zoom** expands a frame vertically to the height of the screen. **Resize**⇒**FullScreen** expands a frame to the full size of the screen. Both of these menu items toggle to **UnZoom**, which restores the frame to its previous size and location. **Zoom** and **UnZoom** have the same accelerator:

- Hold down the **(Control)** key while you click **LEFT** in the frame border. The shorthand notation for this keyboard-mouse combination is **(Control)**-**LEFT**.

To try these out, resize the Console frame to full height:

- Choose **Resize**⇒**Zoom** from the Console's Frame menu.

Then change it back to its original size and location:

- Click **(Control)-LEFT** in the Console frame's border. The **(Control)-LEFT** accelerator zooms a normal frame or, as in this case, unzooms a previously zoomed frame.

2.5. Loading a File

Let's load the limerick from the first lesson into the Text Editor's edit window. If you don't remember its name, you can list the directory in the Command Tool window.

- List all the files in the directory that contain the letters `lim` by typing the following command:

```
venus% ls *lim*
```

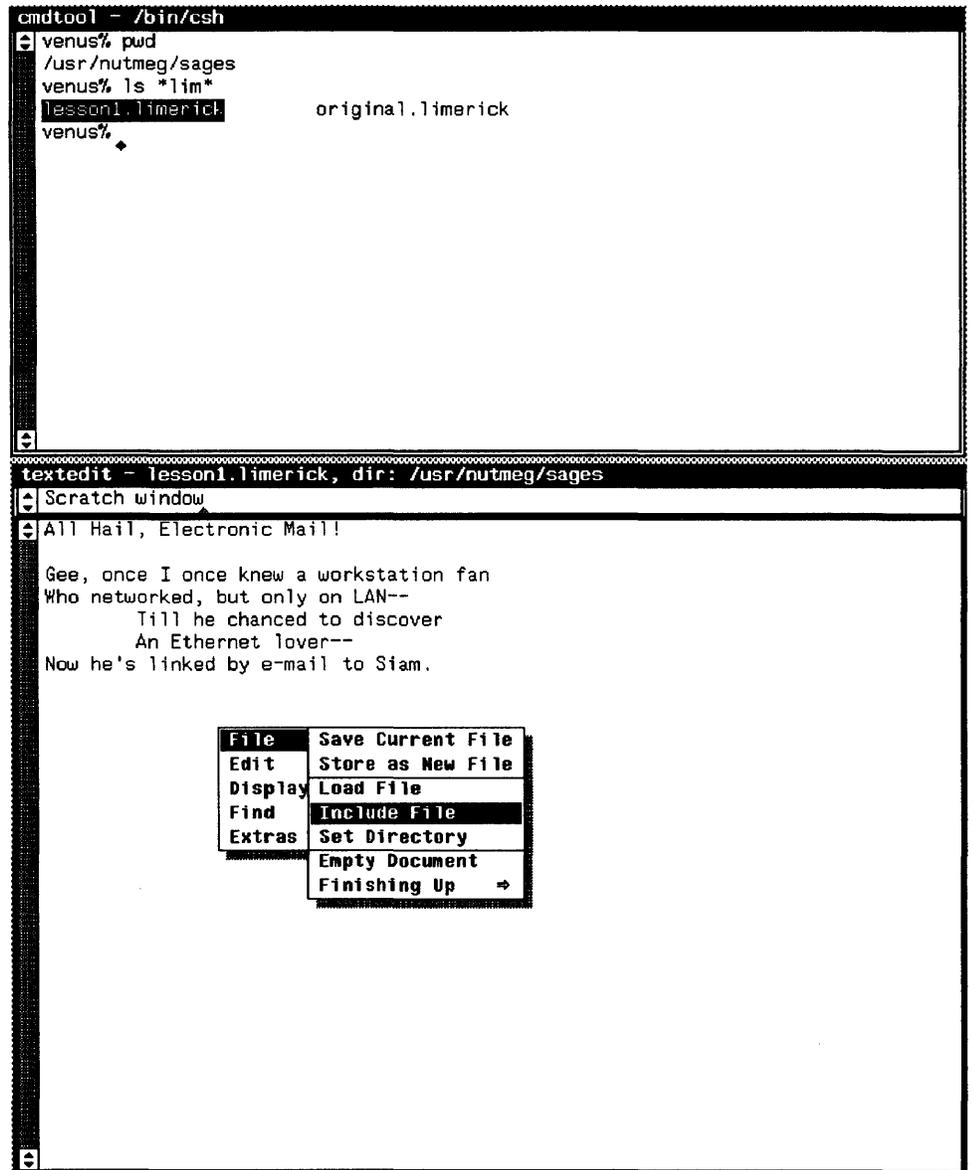
NOTE: Before choosing **Load File**, select the filename. The selection can be in any window.

The name of the file is `lesson1.limerick`. To load the file:

- Select its name in the Command Tool window. **Reminder:** The name is considered two words, so you'll have to extend the selection. Move the pointer into the Text Editor's edit window.
- Bring up the Text menu and choose **File**⇒**Load File**. The file is loaded, and the file name appears in the frame header.

In order to learn scrolling, you need to fill up the edit window with text. An easy way to do that is to put more copies of the limerick file into the window by choosing the **File**⇒**Include File** menu item, as shown in the next figure, or by copying and pasting in the window. You'll have a chance to try both methods. Unlike **Load File**, which clears the window before it loads the file, **Include File** inserts the specified file into the window at the insertion point (the caret) *in addition* to whatever else is in the window.

Figure 2-9 Loading the Limerick File



Add some additional copies of the limerick by following these instructions:

- Select the filename `lesson1.limerick` if it isn't already selected.
- Choose **File**⇒**Include File** from the edit window's Text menu. Another copy of the limerick appears.
- Quadruple-click the LEFT mouse button to select the entire document.
- Press the **(Copy)** function key to copy the selection onto the Clipboard. Now there are two limericks in the document and two on the Clipboard.
- Press the **(Paste)** function key to copy the Clipboard into the document. Now there are four copies of the limerick in the document.

- Repeat the last step twice. Now there are eight copies of the limerick in the document.

There's more text in the document than you can see in the edit window. You could make the window taller by choosing **Resize**⇒**Zoom** from the Frame menu, but that wouldn't help if you had a really large file. The SunView way to see all the text in a document is by scrolling.

2.6. Scrolling

The vertical bar on the left side of the edit window is called the *scrollbar*.³ Every text window contains a scroll bar. The Command Tool has a scrollbar also. The scrollbar lets you make all of a document visible by moving it through the window — it's similar to the way you wind film past the lens of a camera. The difference is that you can only wind the film forward, one frame at a time. Using the scrollbar, you can move a document backward or forward through the window a line at a time or a windowful at a time.

There are several additional scrolling techniques that let you show a specific part of the document, and there are scrolling accelerators. This lesson covers the basics; the advanced techniques are described in Chapter 7, "All about Scrolling".

Since the scratch window in the Text Editor is a text window, it has a scrollbar too. Because the window is only one line high, there isn't room to show the entire scrollbar, so only a scroll box is displayed, as shown in the following figure. The scroll box has the same function as the scroll boxes at either end of the edit window's scrollbar. The function of the scroll boxes is described in the next section.

Figure 2-10 *Scrollbar in the Scratch Window*



Using the Scroll Boxes



The scroll boxes at either end of the scrollbar are identical in function. They're in both places for convenience, so you don't have to move the pointer very far to reach one.

The upward- and downward-facing triangles in the scroll box indicate that you can move the text up or down in the window.

³ This chapter describes the scrollbar's standard appearance. There are many interesting ways to customize the scrollbar. See Chapter 7, "All about Scrolling", and Chapter 10, "The Defaults Editor", in the *Reference* section for more information.

Scrolling by the Line

- Move the pointer into one of the scroll boxes in the edit window's scrollbar. The pointer changes to a diamond shape, indicating that you can scroll in either direction.
- Hold down the LEFT mouse button. The document scrolls *upward* in the window, and the pointer changes to an upward-facing arrow. Release the mouse button after you've scrolled several lines.

NOTE: The LEFT mouse button scrolls lines *upward*. The RIGHT mouse button scrolls lines *downward*.

When you *hold down* the mouse button in a scroll box, the document scrolls continuously, a line at a time, until you release the mouse button.

- With the pointer in the scroll box, click the RIGHT mouse button. The document scrolls *downward* in the window by one line, and the pointer briefly changes to a downward-facing arrow.

When you *click* the mouse button in a scroll box, the document scrolls one line.

Scrolling by the Window

- With the pointer in either scroll box, click the MIDDLE mouse button. The text scrolls upward by one windowful, and the pointer changes briefly to a right-facing triangle.

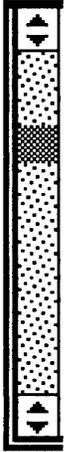
NOTE: The MIDDLE mouse button scrolls upward by the windowful. With the **(Shift)** key pressed, the MIDDLE mouse button scrolls downward by the windowful.

- With the pointer in either scroll box, hold down the **(Shift)** key while clicking the MIDDLE mouse button. The text scrolls downward by one windowful, and the pointer changes briefly to a right-facing triangle.

When you *click* the MIDDLE mouse button in a scroll box, the text scrolls one windowful.

When you *hold down* the MIDDLE mouse button in a scroll box, the text scrolls continuously, a windowful at a time, until you release the mouse button.

The Bar and the Bubble



Scrolling Directly to a Location

Besides the scroll buttons, the scrollbar contains two additional parts:

- The *bar* is the tall light-gray rectangle bounded on either end by scroll boxes.
- The *bubble* is the dark gray part of the bar. The bubble's *size* relative to the size of the bar tells you what percentage of the document is visible. The larger the document, the smaller the bubble because the less of the document you can see at one time. The bubble's *position* in the bar tells you the location of the text in the window relative to the whole document. When the bubble is at the top of the bar, the text in the window is at the beginning of the document; when the bubble is at the bottom of the bar, the text in the window is at the end of the document.

You can't change the bubble directly — it's a "read-only" indicator; like the mercury in a thermometer, it reports on its environment.

Sometimes you want to scroll directly to a particular location in a document. For example, you may remember where the bubble was when the text you want was showing.

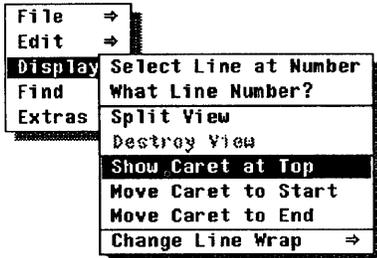
To scroll directly to a relative location in the document, move the pointer into the bar at the point that represents the relative location of the text in the document. The pointer changes to a right-facing arrow. Click the MIDDLE mouse button. The text appears in the window and the bubble reflects the new location.

- Move the pointer into the bar just under the top scroll box. Click the MIDDLE mouse button. This is an easy method for getting to the start of a document.
- Move the pointer to the bottom of the bar, just above the bottom scroll box. Click the MIDDLE mouse button. You see the end of the document.

NOTE: Scrolling doesn't change the location of the caret.

Scrolling in the Command Tool window works just the way it does in the Text Editor. Try it out and see.

2.7. Displaying and Moving the Caret



Occasionally when you're scrolling, you may lose track of where the caret is. The **Display**⇒ pull-right menu in the Text menu contains some handy ways to find or move the caret and display the line containing it in the window.

- **Display**⇒**Show Caret at Top** displays the caret near the top of the window. This is handy if you just want to see where the caret is. Try it.
- **Display**⇒**Move Caret to Start** positions the caret at the beginning of the document and displays the first line in the document at the top of the window. Try it.
- **Display**⇒**Move Caret to End** positions the caret at the end of the document and displays the last line in the document at the bottom of the window. This is handy if you want to add text at the end of the document. Try it.

Now you know a number of ways to move around in a document of any size.

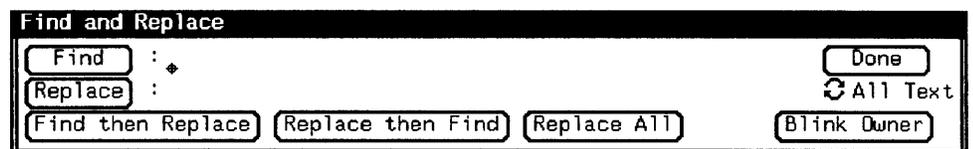
2.8. Finding and Replacing Text

You know how to use the Text menu or the function key accelerators to paste text into a document. The **Find and Replace** frame does that, and more. It lets you look for a specified set of characters and replace one or some or all occurrences.

The **Find and Replace** frame is a panel that the Text Editor brings up when you choose **Find**⇒**Find and Replace** from the Text menu. A *panel* is another type of window. The panel in the **Find and Replace** frame contains *buttons* like the buttons in an alert window, a *cycle item* that lets you make choices, and *text items* where you can type and do limited editing.

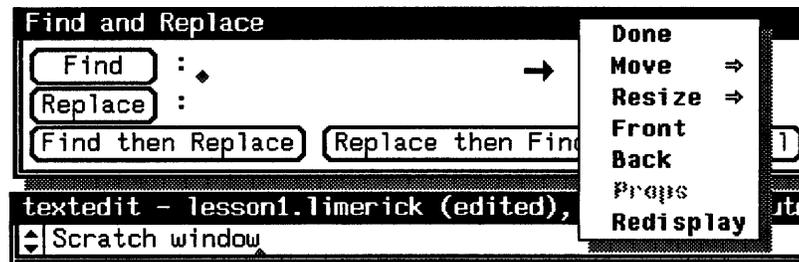
- Bring up the Text menu and choose **Find**⇒**Find and Replace**. The **Find and Replace** frame, shown in the following figure, appears above the Text Editor frame.

Figure 2-11 *The Find and Replace Frame*



Like other frames, the **Find and Replace** frame has a frame menu, but it isn't quite the same as the Text Editor's frame menu.

- Preview the two frame menus. Notice that the **Find and Replace** frame's Frame menu doesn't have **Quit** at the bottom, and it has **Done** instead of **Close** at the top.

Figure 2-12 *Find and Replace Frame Menu*

Because there's no **Quit** item in this frame, you can't leave the Text Editor tool from here. And you can't close the **Find and Replace** frame to an icon. When you choose **Done**, the **Find and Replace** frame disappears, but the Text Editor frame remains.

In most other respects, the **Find and Replace** frame is like any other frame. It usually comes up just above its *owner* — the frame containing the window from which it was chosen — but you can resize it or move it to a new location.

If the **Find and Replace** frame doesn't appear where you expect it, you can easily find it. Just choose **Find and Replace** again, and the **Find and Replace** frame will blink.

Pushing Panel Buttons

A button in a panel represents an action. To choose the action, move the pointer onto the button and click the LEFT mouse button. This is known as *pushing the button*. It's just like responding to an alert message, which you learned how to do in Chapter 1.

- Push the **Blink Owner** button in the **Find and Replace** frame.

Blink Owner causes the frame that owns this **Find and Replace** frame to highlight rapidly three times. When there's more than one **Find and Replace** frame on the screen, **Blink Owner** lets you find out which owner each pop-up frame belongs to.

- In the edit window, choose **Find⇒Find and Replace** again.

The **Find and Replace** frame blinks to let you know it's already available.

Replacing Text

Let's replace all the occurrences of `workstation`.

- Select the word `workstation` anywhere in the document.
- Choose **Find⇒Find and Replace**.

NOTE: A text item in a panel is usually preceded by a colon (:).

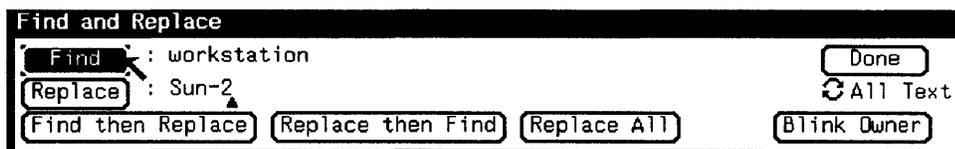
If text is selected when you choose **Find⇒Find and Replace**, the selection is automatically pasted into the **Find:** text item. (You could also click LEFT in a text item to set the insertion point and type the text you want. Or you could put the text on the Clipboard and paste it into the text item in the panel.)

NOTE: To move from one text item to the next in a panel, press the **Return** key or the **Tab** key.

When the **Find and Replace** frame puts the selection in the **Find:** text item, it also moves the insertion point into the **Replace:** text item. Just as in the edit window, the insertion point — the caret — is a blinking black triangle when the panel has the keyboard focus and a gray diamond when it doesn't.

- Move the pointer into the **Find and Replace** panel. Make sure the !68 caret is in the **Replace:** text item.
- Type Sun-2. The panel text items in the **Find and Replace** frame now look like those in the following figure.

Figure 2-13 *Finding and Replacing Text*



NOTE: After you push a panel button, the button turns gray while the action is taking place.

- Push the **Find** button (not the **Find** function key). An occurrence of `workstation` is now highlighted in the edit window to indicate that it has been found. It looks just like a normal selection, and in fact it is.
- Push the **Replace** button to change `workstation` to `Sun-2`. The selection is replaced, and the caret in the edit window moves to the right of the replacement.
- Move the pointer into the panel and change the **Replace:** text item from `Sun-2` to `Sun-3`. The caret is positioned after `Sun-2` because that was the last text you typed in the panel.
- Press the **Delete** key once to erase the `2`, then type `3`. Now the **Replace:** text item contains `Sun-3`.
- Push the **Find** button again. The next occurrence of `workstation` is selected.
- Push the **Replace then Find** button. The selection changes from `workstation` to `Sun-3` and yet another occurrence of `workstation` is found.
- Move the pointer into the panel and change `Sun-3` to `Sun-4` by erasing the `3` and typing `4`.
- Push the **Replace All** button. The remaining occurrences of `workstation` are replaced by `Sun-4`.
- Push the **Find** button.

Since `workstation` no longer exists in the edit window, the window blinks and beeps to tell you the **Find:** text item wasn't found.

- Scroll through the document in the edit window to see the changes you made.

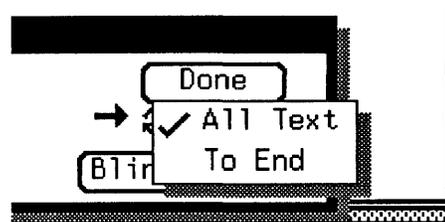
Menus in Panels

Panel items often have menus associated with them.

Cycle items, like the **All Text/To End** item in the **Find and Replace** panel, always have menus. The current choice is checked in the menu and appears to the right of the cycle item.

- **To End** means search from the caret to the end of the document.
- **All Text** (the default) means search from the caret to the end of the document, then from the beginning of the document back to the caret.

Figure 2-14 *A Cycle Item's Menu*



NOTE: Menus in panels work just like menus in other windows.

To preview a menu in a panel, move the pointer to the item and hold down the **RIGHT** mouse button. To dismiss the menu without choosing an item, move the pointer off the menu so that no menu item is highlighted (moving the pointer leftward usually works best), and release the mouse button.

- There's a second menu in the **Find and Replace** panel. Can you find it? Move the pointer onto each button and other item in the panel, and preview to see if there's a menu behind it.

The **Find** button contains a menu that lets you specify **Forward** or **Backward** as the direction of the search. **Forward** is the default.

Other tools with panels include the Mail Tool and the Defaults Editor. When you start to use them, check for menus so you won't miss out on some nifty features.

- You've finished the part of the lesson that uses the **Find and Replace** frame, so click **LEFT** on the **Done** button to put the frame away.

2.9. Wrapping Long Lines of Text

The limerick you've been working with has very short lines. If you will be working with documents that contain text in paragraph form, or that contain long lines, you'll want to know about the **Display⇒Change Line Wrap⇒** pull-right menu in the Text menu.

First let's get some long lines into the document by replacing all the Return characters with spaces. Since the **(Return)** key has a special function — moving the caret between the **Find:** and **Replace:** text items — there's no way to search for the Return character using the **Find and Replace** frame, but you can use other find features to do this search.

- Click **LEFT** after the last character in the first line of the limerick to select the Return character.
- Choose **Find⇒Find Selection⇒** to find the Return character in the next line.
- Press the **(Cut)** function key, then type a space. Cutting puts the Return character on the Clipboard.

Now that you've replaced the first Return character with a space, you can easily repeat the sequence of actions.

- Press the **(Again)** function key. (Or choose **Edit⇒Again** from the Text menu.)

Each time you press **(Again)**, the next replacement occurs. **(Again)** lets you repeat a series of edit operations.

- Press **(Again)** repeatedly until the window blinks and beeps. That's the signal that **Find** can't find any more Return characters.

Wrap at Character

Now the document is a single long line. You can type normal text — letters and memos — this way, without inserting Return characters except at the end of a paragraph. The Text Editor displays text on a new line in the window when the current line is about to overflow.

NOTE: The **(Find)** function key is an accelerator for **Find⇒Find Selection⇒**.

- Try getting rid of the Tab characters the same way you got rid of the Return characters, using **(Find)**, **(Cut)** and **(Again)**. (When there's a selection — in this case, when you select a Tab character — pressing the **(Find)** function key is the equivalent of choosing **Find⇒Find Selection⇒Forward**.) The edit window should then look like the one in the following figure.

Figure 2-15 *A Document with Character Wrap*

```

textedit - lesson1.limerick (edited), dir: /usr/nutmeg/sages
Scratch window
All Hail, Electronic Mail! Gee, once I once knew a Sun-4 fan Who networked, but
only on LAN-- Till he chanced to discover An Ethernet lover-- Now he's linked
by e-mail to Siam. All Hail, Electronic Mail! Gee, once I once knew a Sun-2 fa
n Who networked, but only on LAN-- Till he chanced to discover An Ethernet lov
er-- Now he's linked by e-mail to Siam. All Hail, Electronic Mail! Gee, once I
once knew a Sun-3 fan Who networked, but only on LAN-- Till he chanced to disco
ver An Ethernet lover-- Now he's linked by e-mail to Siam. All Hail, Electronic
Mail! Gee, once I once knew a Sun-4 fan Who networked, but only on LAN-- Till
he chanced to discover An Ethernet lover-- Now he's linked by e-mail to Siam.
All Hail, Electronic Mail! Gee, once I once knew a Sun-4 fan Who networked, but
only on LAN-- Till he chanced to discover An Ethernet lover-- Now he's linked
by e-mail to Siam. All Hail, Electronic Mail! Gee, once I once knew a Sun-4 fa
n Who networked, but only on LAN-- Till he chanced to discover An Ethernet lov
er-- Now he's linked by e-mail to Siam. All Hail, Electronic Mail! Gee, once I
once knew a Sun-4 fan Who networked, but only on LAN-- Till he chanced to disco
ver An Ethernet lover-- Now he's linked by e-mail to Siam. All Hail, Electronic
Mail! Gee, once I once knew a Sun-4 fan Who networked, but only on LAN-- Till
he chanced to discover An Ethernet lover-- Now he's linked by e-mail to Siam.

```

Wrap at Word

The text in the document *wraps* at the right edge of the window. If a word doesn't fit on a line, it wraps the remaining characters around to the next line. That's kind of hard to read because words at the end of the line may be broken in strange places, so let's change the way the line wraps:

- Choose **Display**⇒**Change Line Wrap**⇒**Wrap at Word**. The window changes to look like the following figure. Notice that words are no longer split between lines. If a word is too long to fit at the end of a line, it is moved to the beginning of the next line.

Figure 2-16 *A Document with Word Wrap*

```

textedit - lesson1.limerick (edited), dir: /usr/nutmeg/sages
Scratch window
All Hail, Electronic Mail! Gee, once I once knew a Sun-4 fan Who networked, but
only on LAN-- Till he chanced to discover An Ethernet lover-- Now he's linked
by e-mail to Siam. All Hail, Electronic Mail! Gee, once I once knew a Sun-2 fan
Who networked, but only on LAN-- Till he chanced to discover An Ethernet
lover-- Now he's linked by e-mail to Siam. All Hail, Electronic Mail! Gee, once
I once knew a Sun-3 fan Who networked, but only on LAN-- Till he chanced to
discover An Ethernet lover-- Now he's linked by e-mail to Siam. All Hail,
Electronic Mail! Gee, once I once knew a Sun-4 fan Who networked, but only on
LAN-- Till he chanced to discover An Ethernet lover-- Now he's linked by
e-mail to Siam. All Hail, Electronic Mail! Gee, once I once knew a Sun-4 fan
Who networked, but only on LAN-- Till he chanced to discover An Ethernet
lover-- Now he's linked by e-mail to Siam. All Hail, Electronic Mail! Gee, once
I once knew a Sun-4 fan Who networked, but only on LAN-- Till he chanced to
discover An Ethernet lover-- Now he's linked by e-mail to Siam. All Hail,
Electronic Mail! Gee, once I once knew a Sun-4 fan Who networked, but only on
LAN-- Till he chanced to discover An Ethernet lover-- Now he's linked by
e-mail to Siam. All Hail, Electronic Mail! Gee, once I once knew a Sun-4 fan
Who networked, but only on LAN-- Till he chanced to discover An Ethernet
lover-- Now he's linked by e-mail to Siam.

```

The actual text hasn't changed. Only its appearance on the screen is different.

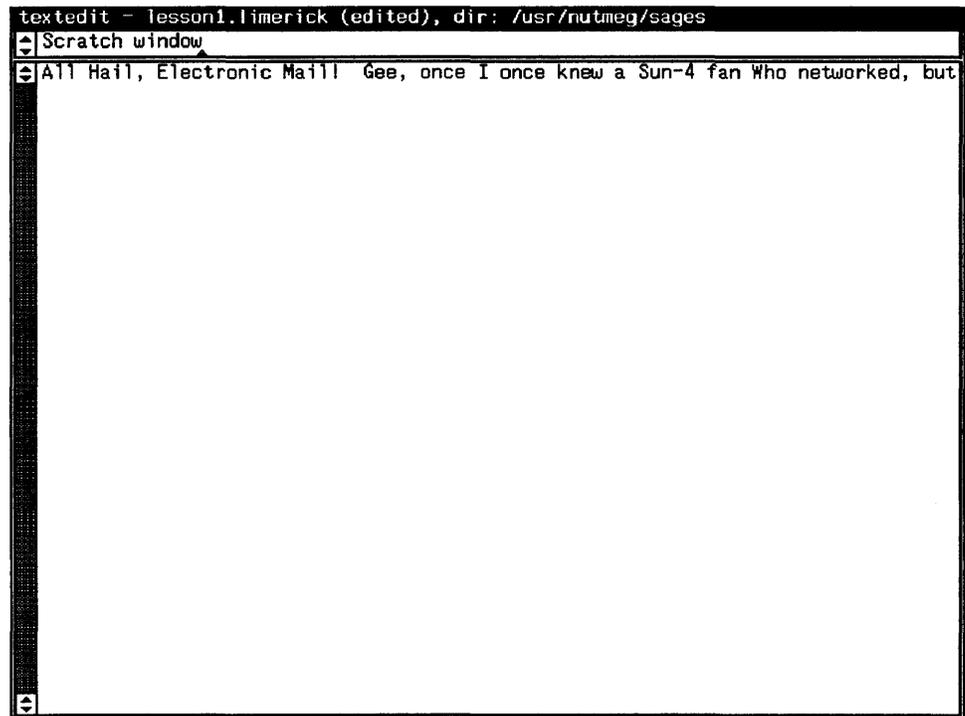
NOTE: In Mail Tool and certain other tools, and in some SunOS commands such as `lpr`, **Wrap at Word** looks great in the window but may give unsatisfactory results if you print the contents. For better printed output, control the line length yourself by pressing the **Return** key at the end of each line, as you did when typing the limerick.

Clip Lines

Instead of wrapping text, you can *clip* it so that whatever doesn't fit on a line of the screen doesn't show. It appears to be truncated on the right. Clipping doesn't actually snip off the part of the text that isn't shown — it just hides it.

- Choose **Display⇒Change Line Wrap⇒Clip Lines**. Only the first few words of the document appear in the window.

Figure 2-17 *A Document with Clipped Lines*



Remember that we took out all the Return characters, so the Text Editor regards the document as one long line. It has clipped off the part that couldn't be displayed. If you widen the frame with **Resize⇒**, you'll see more of the line. Or you can change the line wrap back to the way it was:

- Choose **Wrap at Character** from the **Display⇒Change Line Wrap⇒** pull-right menu to go back to the default setting.

It's time to end this lesson. If you'd like to save the document in its present form, you can either choose **File⇒Save Current File** or type a new name in the scratch window, select it, and choose **File⇒Store as New File**.

- Preview the **File⇒Finishing Up⇒** pull-right menu to see some other interesting ways to save a file.

2.10. Exiting SunView

The normal way of ending a work session is to lock the screen. You don't have to go back to the SunOS shell, since you can execute commands from a Command Tool. Occasionally, though, you may want to restart SunView in order to change the startup options.

To exit and restart SunView:

- Move the pointer over the gray background.
- Bring up the SunView menu and choose **Exit SunView**.
- Follow the instructions for starting SunView at the beginning of Chapter 1, "Getting Started with SunView".

2.11. A Word to the Wise

You now know all of the SunView basics plus some fancy twists. But there's a lot you can do to make your SunView environment more efficient and personally pleasing. You'll miss out on many attractive features unless you read the *Reference* section, especially the chapter that describes the Defaults Editor.

To whet your appetite, here are some features you might like to try:

- *Stay-up menus* let you preview menus without having to keep holding down the mouse button. You click to bring the menu up and click again to choose a menu item or take the menu down.
- *Pending-delete* is a way of selecting text so that you can replace it without having to delete it first. This feature makes the Text Editor work more like a personal computer's word processor.
- *Icons* can be displayed in the SunView menu instead of toolnames. The **Clock**⇒ pull-right menu in the SunView menu shows you how this looks.
- *If you're left-handed*, you can rearrange the standard function keys and add special functions of your own.
- *Split View* in the Text menu lets you divide a window into two or more sections and scroll them individually.
- *Reduced mousing* is possible by using keyboard accelerators instead of the mouse to move the caret around.
- **Find Marked Text** is a pop-up frame that you bring up from the **Find**⇒ pull-right menu. It finds text that's enclosed in certain standard delimiters, like parentheses and quotation marks.
- The **Extras**⇒ pull-right menu contains lots of formatting shortcuts for source programs and other text.

If nothing on this list caught your attention, you probably have everything you need to be happy in SunView right now. Go to it!

If you saw something you'd like to try, read on. And happy exploring.

PART TWO: REFERENCE

Starting and Exiting SunView

3.1. Starting SunView

To start SunView from a SunOS command line, type `sunview` in response to the command prompt:

```
venus% sunview
```

The directory you're in when you start SunView is the directory that SunView tools come up in. This manual shows SunView starting from your home directory, but you can start SunView in any directory. For more details, follow the instructions in Chapter 1, "Getting Started with SunView". These instructions tell you how to bring up SunView with the standard options.

There are various ways to modify the SunView display, as described under the `sunview` command in the *SunOS Reference Manual*. Two of the more interesting options are shown below.

Customizing the Background Pattern

The SunView background doesn't have to be plain gray. Using the `-background` option, you can display a picture on the screen to serve as the background.

- To display a background picture, type the following command line:

```
venus% sunview -background filename
```

The *filename* file contains the picture you want to use as a background. This file is a standard *rasterfile*, a graphics file that you can produce using `screen-dump`, the Icon Editor, or other image-producing software.⁴

For example, suppose you have a file called `celtics` that contains the background pattern shown in the next figure.

⁴ For more information about `screendump`, see the `screendump` command in the *SunOS Reference Manual*.

- To start SunView with the `celtics` file as the background, type the following command:

```
venus% sunview -background celtics
```

The frames and icons will be displayed over the `celtics` background.

Figure 3-1 *A Cheering Background*



Background in Inverse Video

You may prefer *inverse video* — black windows on a dark gray background, with text and frame borders in white — rather than the default of white windows and black text on a light gray background. Some people think inverse video is easier on the eyes.

- To display the custom background screen in inverse video, add the `-i` (for *inverse*) option when you start SunView:

```
venus% sunview -i -background celtics
```

Here's what inverse video looks like:

Figure 3-2 *The Customized Screen in Inverse Video*



You don't have to have a special background to use inverse video. You can use the `-i` option all by itself to get the standard SunView screen in inverse video.

3.2. Saving the Layout

If you like the way you've set up your screen and don't want to go to the trouble of starting tools and moving and resizing frames every time you start SunView, you can have SunView recreate your personal environment automatically.

The **Save Layout** menu item in the SunView menu saves a map of the screen in the `.sunview` file in your home directory. (If the file already exists, its contents are replaced.) The next time you start SunView, the information in the `.sunview` file is used to recreate your environment just the way you like it. Or you can save several layouts and specify the one you want — see the topic “Managing Multiple Frames” in Chapter 5, “All about Frames and Windows” for details.

3.3. Locking the Screen

Locking the screen displays a changing pattern that protects the privacy of your open windows and also prevents the image from burning into the screen when you leave your workstation unattended for a long period of time.

- To lock the screen, choose **Services⇒Lock Screen** from the SunView menu.
- To unlock the screen, click any mouse button or press any keyboard key except **(F1)** or **(Caps)**, then type your password and press **(Return)**.

For more information on locking and unlocking the screen, see the information on **Lock Screen** in the *Lessons* section and in Chapter 8, "All about Menus".

3.4. Exiting SunView

When you quit SunView, the contents of all windows are forgotten unless you save them. (By contrast, when you close a frame, the contents of its windows are remembered and restored when you open its icon.) The screen layout (current icon and frame locations) is also forgotten unless you saved it by choosing **Services⇒Save Layout** from the SunView menu.

- To quit the SunView windows environment and return to the SunOS command line, choose **Exit SunView** from the SunView menu.

Emergency Exit

Sometimes you want to leave SunView under catastrophic conditions, and you can't access the SunView menu.

To exit SunView in an emergency:

- Move the pointer over the gray background. This emergency technique works only if the pointer is not over a frame or icon.
- Type **(Control-D)** followed by **(Control-Q)** to bail out from the keyboard.

NOTE This is not the recommended way to exit SunView. Use the SunView menu if you can.

All about Keyboards

This chapter tells you:

- How to tell which model keyboard you have.
- How to label the function keys using the stickers provided with your Sun documentation.
- How to notify the Defaults Editor that you're a left-handed user of the keyboard and mouse.
- How to change your keyboard focus to Click-to-Type.
- How to "map" (customize) the unused function keys.

The use of the keyboard for menu and mouse shortcuts is described in Chapter 9, "All about Accelerators".

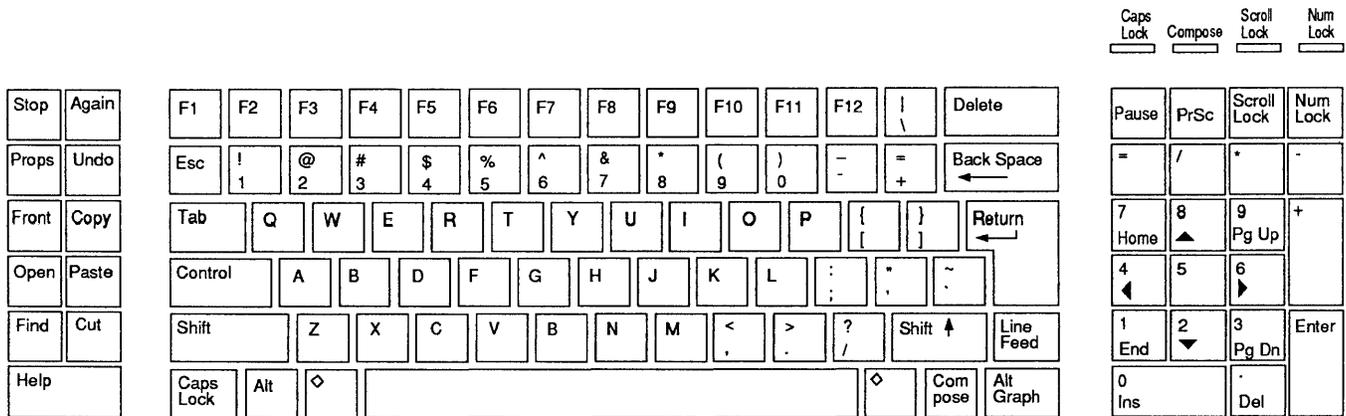
4.1. Which Sun Keyboard Do You Have?

Diagrams of the Type4, Type3, and Type2 keyboards are provided on the following pages. Compare the diagrams to your keyboard to determine which type you have.

This manual assumes you have a Type4 keyboard. If you have another type, use the labels provided with your documentation to label your keyboard so that it looks like a Type4 keyboard. (See the next topic for details.) Then you'll be able to follow the keyboard references in this manual.

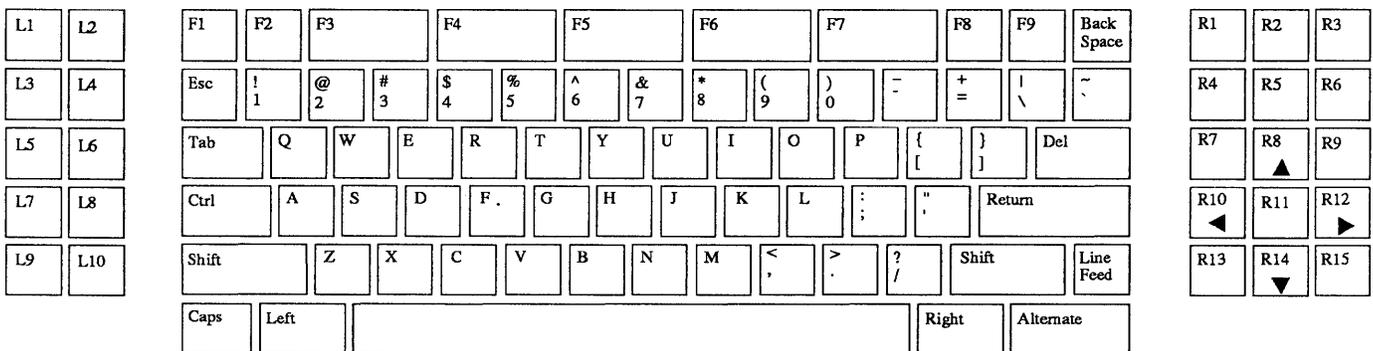
If you're a left-handed keyboard and mouse user, you'll need labels even if you have a Type4 keyboard.

Figure 4-1 The Type4 Keyboard



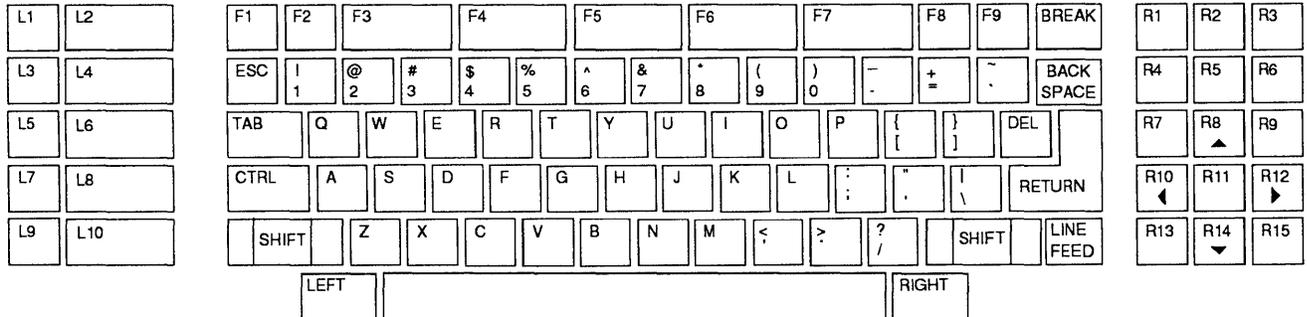
In addition to the **Help** key, you can also request Help by using the **Meta-/** keys. There are two meta keys marked by diamonds (♦) that are located immediately to the left and right of the space bar. The “diamond” (♦) labels are for the **Meta** keys — the keys on either side of the space bar. (Remember **Meta-/** means holding down a **Meta** key and the **/** key together.) Although a new Help facility is now available for most windows, frames, and panels, the Help text must be supplied by someone at your installation. Refer to *SunOS 4.1 Release Notes* for more details.

Figure 4-2 The Type3 Keyboard



Use the **Meta-/** keys to request Help. There are two **Meta** keys normally marked **Left** and **Right** located immediately to the left and right of the spacebar. (Remember **Meta-/** means holding down a **Meta** key and the **/** key together.) The “diamond” (♦) labels are for the **Meta** keys — the keys on either side of the space bar. On the Type3 keyboard, these keys are marked Left and Right. Although a new Help facility is now available from most windows, frames, and panels, the Help text must be supplied by someone at your installation. Refer to *SunOS 4.1 Release Notes* for more details.

Figure 4-3 The Type2 Keyboard



Use the **(Meta-)** keys to request Help. There are two **(Meta)** keys that are normally marked **(Left)** and immediately **(Right)** located and right of the space bar. (Remember **(Meta-)** means holding down a **(Meta)** key and the **(/)** key together.) The “diamond” (♦) labels are for the **(Meta)** keys — the keys on either side of the space bar. On the Type2 keyboard, these keys are marked LEFT and RIGHT.

Although a new Help facility is now available from most windows, frames, and panels, the Help text must be supplied by someone at your installation. Refer to *SunOS 4.1 Release Notes* for more details.

4.2. How to Label the Keyboard

This section applies only to Type2 and Type3 keyboard users and to left-handed Type4 keyboard users. The Type4 keyboard is already labeled correctly for right-handed users.

NOTE: Your workstation model number may differ from your keyboard model. For example, it's possible to use a Type2 keyboard with a Sun-3™ workstation, and vice versa.

Other keyboards must be labeled with the stickers that come with your Sun documentation. The labels make your keyboard look like a Type4 keyboard so that you can follow the instructions in this manual and related manuals.

Here's how to label your keyboard (if you're left-handed, read the next topic before following these instructions):

- Compare your keyboard with Figures 4-1 through 4-3 to find out what kind of keyboard you have.
- Follow the labeling diagram that corresponds to your keyboard type — see figures 4-4 through 4-6.
- Move the pointer over the gray background as you affix the labels to the keys so you won't accidentally cause a function key operation to be performed.
- Affix the appropriately sized and marked labels provided with your Sun documentation.

You can remove and restick the labels, if necessary, without damaging the labels or the keyboard.

4.3. Left-Handed Keyboard Use

The Defaults Editor, described in Chapter 10, lets you specify which side of the keyboard you want the function keys on.

If you like moving the mouse with your right hand, you don't have to change anything. Right-handed people use the keys located on the left of the keyboard as their function keys.

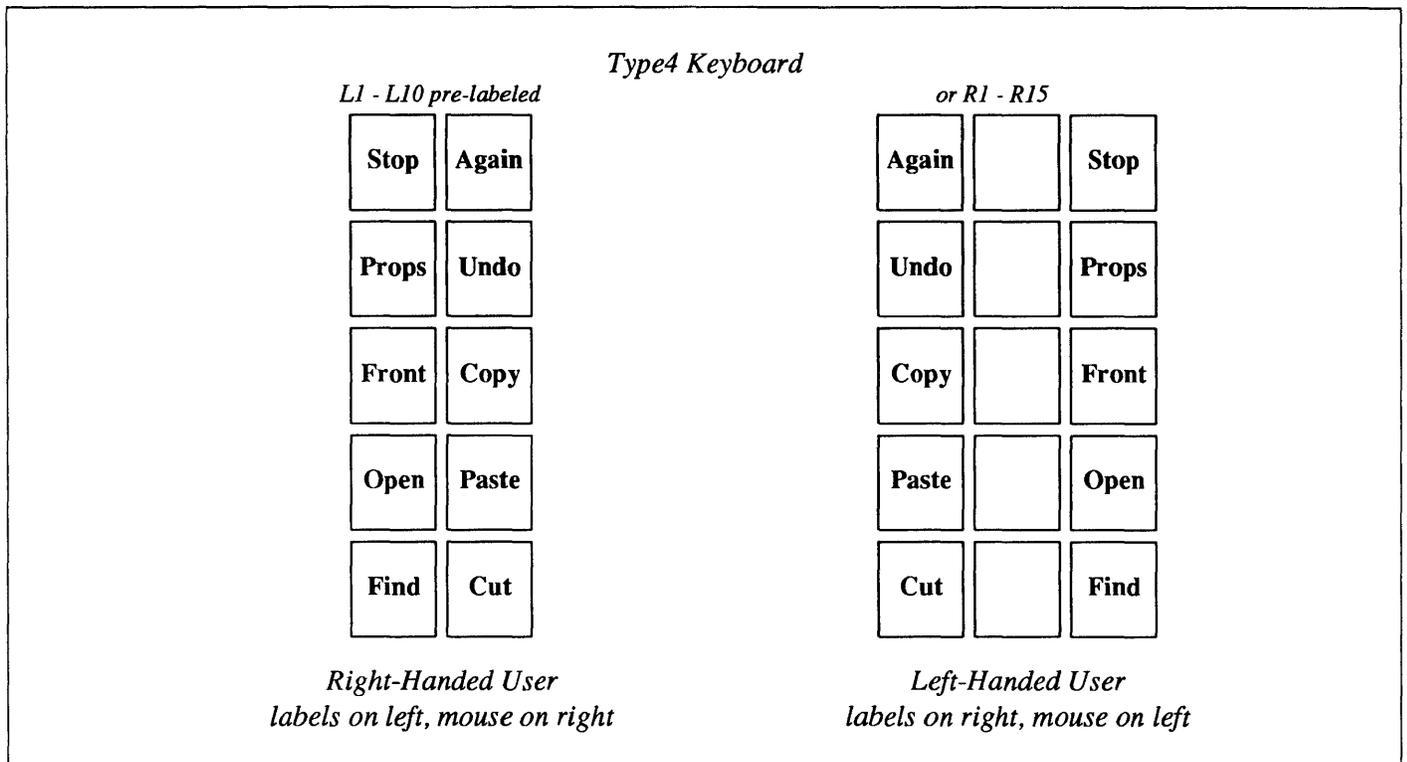
However, left-handed people sometimes prefer having function keys on the right side of the keyboard and moving the mouse on the left. They use keys in the right-hand keypad, **R1** through **R15**, as their function keys.

When you set up your system for left-handed use, the LEFT and RIGHT mouse buttons switch so that the mouse button you click with your index finger is always LEFT whether you're right-handed or left-handed.

To set up the keyboard and mouse for left-handed use:

- Label the keyboard, as described above. Put the function key stickers on the right side of the keyboard.
- Notify the Defaults Editor that you're using left-handed mousing and function keys on the right by setting the default value for the `Left_Handed` default line in the **Input** category to `Yes`.

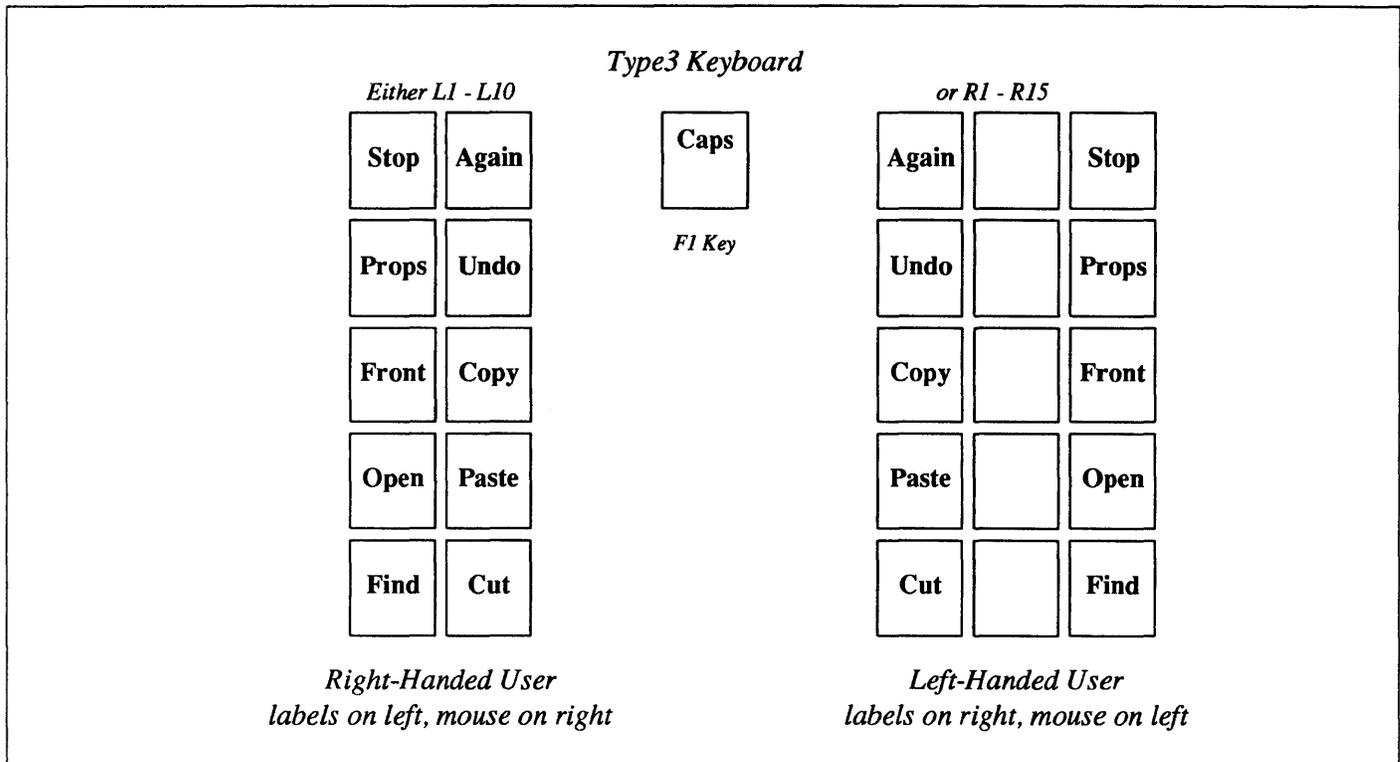
Figure 4-4 How to Label the Type4 Keyboard



The Type4 function keys on the left have names printed on them. If you're right-handed, that's all you need.

If you plan to use a left-handed setup, label the keys as shown on the right side of the diagram.

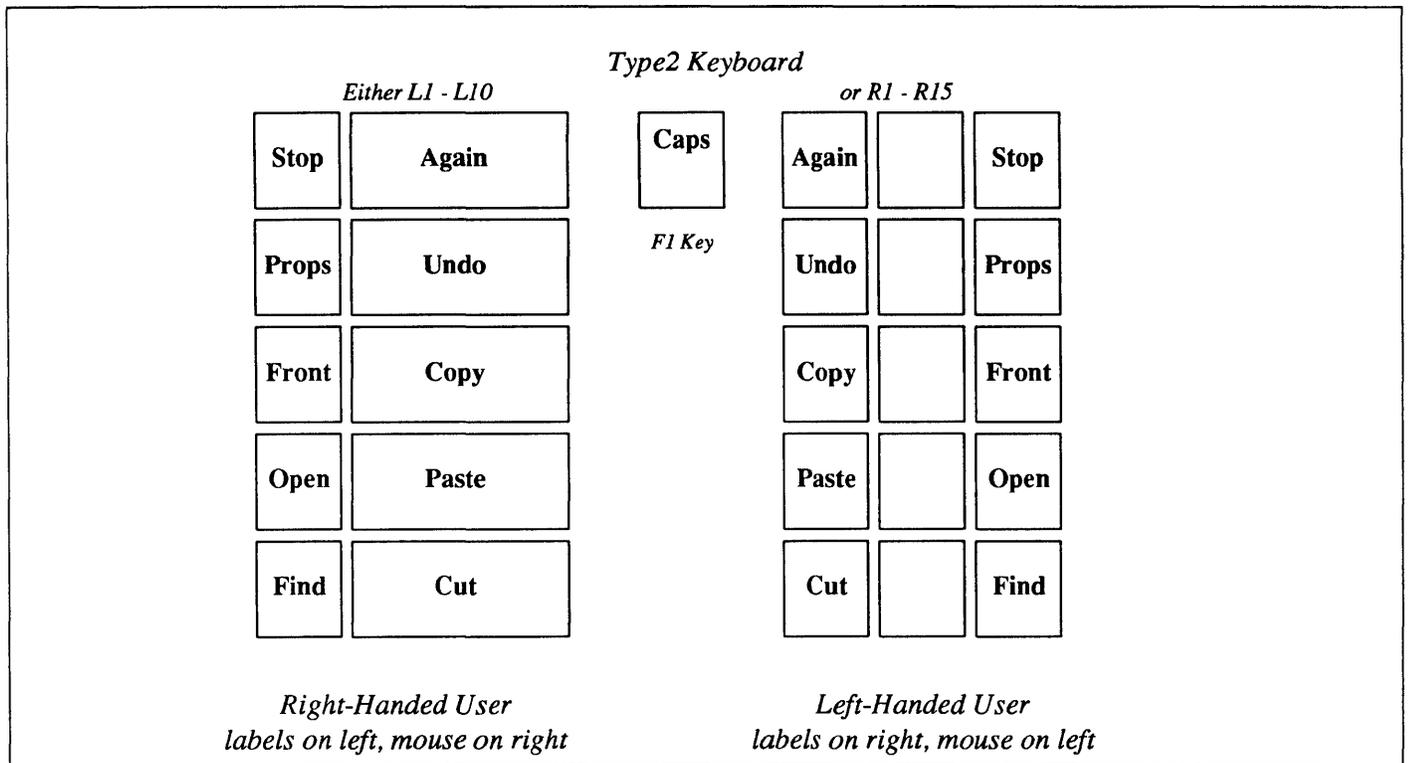
Figure 4-5 How to Label the Type3 Keyboard



The Type3 keyboard has a pre-labeled **Caps** key, but the **F1** key acts as a **Caps** key for individual windows, so put the **Caps** label on the **F1** key also.

The “diamond” (◆) labels are for the **Meta** keys — the keys on either side of the space bar. On the Type3 keyboard, these keys are marked Left and Right.

Figure 4-6 How to Label the Type2 Keyboard



The even-numbered function keys between **[L1]** and **[L10]** on the left side of the Type2 keyboard are narrower than the odd-numbered function keys, so use the narrow labels for those keys.

The “diamond” (◆) labels are for the **[Meta]** keys — the keys on either side of the space bar. On the Type2 keyboard, these keys are marked LEFT and RIGHT.

The Type2 keyboard has no built-in **[Caps]** key. Put the **[Caps]** label on the **[F1]** key.

4.4. Click-to-Type Keyboard Focus

The window that receives whatever you type is said to have the *keyboard focus*. Here's how you can tell which window has the keyboard focus:

- **The window border thickens.** When a window receives the keyboard focus, its border changes to a bold black line. When the keyboard focus is not in the window, its border is a thin line.
- **The caret blinks.**⁵ The caret represents the *insertion point* — the place in the text window that receives the next character you type or paste. When the keyboard focus is in a window, its caret is a small black blinking triangle just below the line of type. When the keyboard focus is not in a window, its caret appears as an unblinking gray diamond.

Normally you set the keyboard focus — choose the window you want to type in — by moving the pointer into the window. This is called *Pointer-in-Window* or *Mouse-to-Type* keyboard focus, and it's the default in SunView.

Click-to-Type is an alternative method of setting the keyboard focus. With Click-to-Type, you click the LEFT or MIDDLE mouse button inside the window you want to have the keyboard focus, without pressing any function key. Then, no matter where you move the pointer, the keyboard focus stays in the window you clicked in. In fact, you can even type into a frame that's been closed to an icon when you have Click-to-Type.

Why Use Click-to-Type?

When copying text from one window to another with Pointer-in-Window keyboard focus, you must be careful to press the (Paste) key with the pointer in the destination window before moving the pointer to the source window to select the text to copy. Then you have to remember to move the pointer back to the destination window before continuing to type.

Similarly, when using Dbx Tool, you can't move the pointer into the source window to select a breakpoint location and immediately type a command — you have to return the pointer to the command window first.

Instead, you can use Click-to-Type keyboard focus. Click-to-Type allows keyboard input and mouse input to have different destinations.

When you use Click-to-Type, the destination for keyboard input isn't changed until you click with the pointer in another window:

- You click LEFT in a new window to change the keyboard focus, set the caret, and make a selection.
- You click MIDDLE in a new window to change the keyboard focus while retaining the current selection.

With Click-to-Type, for example, you can move the pointer outside the text window where you're typing in order to push buttons in a control panel or make a secondary selection in another window. Then you can continue typing without having to move the pointer back into the text window.

⁵ You can fine-tune performance by turning off the `Text/Blink_caret` option in the Defaults Editor. If you do, the caret will be a black unblinking triangle.

How to Get Click-to-Type

You can permanently change the type of keyboard focus through the Defaults Editor:

- To change keyboard focus to Click-to-Type, set the `Click-to-Type` default value in the `SunView` category to `Enabled`.
- To change back to Pointer-in-Window, set the `Click-to-Type` default value back to `Disabled`, the `SunView` default.

If you want to change the type of keyboard focus temporarily to see which kind you like best, run `swin` in a Command Tool. The `swin` command changes the type of keyboard focus for all windows immediately.

- To set Click-to-Type keyboard focus, type:

```
venus% swin -c
```

- To go back to Pointer-in-Window (Mouse-to-Type) keyboard focus, type:

```
venus% swin -m
```

4.5. Type4 Keyboards

The Type4 keyboard has some capabilities not available for Type2 or Type3 keyboards.

International Key Caps

The Type4 keyboard can be relabeled with a special set of plastic key caps that show the characters set of a particular language. There are a number of these sets available for different international languages.

The changes done at the factory to support a language other than English are:

- Replacing selected plastic key caps that show the characters now assigned to various keys.
- Modifying the keyboard translate tables to reflect the new character set.

These changes are normally done to the workstation before it is shipped, so a user need not worry about them.

If it is necessary to modify the keyboard translation tables, refer to the *SunOS Reference Manual* for a detail description in `keytables(5)`. The correct translation table is then loaded with the `loadkeys(1)` command.

Key Clicks

When each key is struck on a Type4 keyboard a short clicking sound is heard in addition to the sound of impact as a key strikes. You can use the `click(1)` command documented in the *SunOS Reference Manual* to turn this feature on or off. You can use the following commands.

```
# turn on
venus% click -y
# turn off
venus% click -n
```

4.6. Mapping Functions to Keys

You can change the way text windows and tty windows respond to some of the function keys and other non-alphanumeric keys by associating them with functions or text filters. This is known as *mapping* or *binding filters to keys*.

Text windows are remapped through a `.textswrc` file. Tty windows are remapped through a `.ttyswrc` file. In order to cause remapping, the files must be in your home directory.

Which keys can be remapped depends on whether you're operating SunView in the normal right-handed mode or have set `Input/Left_Handed` to `Yes` in the Defaults Editor.

Tty Windows and the `.ttyswrc` File

With a `.ttyswrc` file in your home directory, you can assign alternate functions to keys and change the default page mode of your tty windows.

The table below shows the keys that are *not* available to be remapped.

Table 4-1 *Keys Not Available for Remapping in `.ttyswrc`*

<i>Right-Handed Keyboard</i>	<i>Left-Handed Keyboard</i>
L1-L10, F1	L1, F1 R1, R4, R7, R10, R13, R3, R6, R9, R12, R15.
With arrow keys enabled: R8, R10, R12, R14.	With arrow keys enabled: R2, R5, R8, R14

If your file tries to remap a key that can't be remapped, you'll see an error message like this:

```
.ttyswrc error: keyname cannot be mapped.
```

This message means that the `.ttyswrc` file contains function key specifications that conflict with the standard SunView function key assignments. Be careful about trying to assign new functions to the function keys used by text windows or to the arrow keys. When you start a SunView tool, SunView takes control of your function keys, disallowing any other functions you may have assigned to those keys.

Enabling Page Mode

Page mode refers to the mechanism in a tty window that lets you look at one windowful at a time. It's described under the **Enable Page Mode** menu item in the Tty menu, in Chapter 8, "All about Menus".

By default, page mode is disabled in tty windows until you choose **Enable Page Mode**. However, you can set the default to "enabled" by including the following

line in your `.ttyswrc` file.

```
set pagemode
```

Format of the `.ttyswrc` File

The format of the `.ttyswrc` file is:

#	Starts comments line.
set pagemode	Enables page mode.
map <i>key</i> <i>text</i>	Inputs <i>text</i> when <i>key</i> is pressed.
mapo <i>key</i> <i>text</i>	Outputs <i>text</i> when <i>key</i> is pressed.

Key can be any function key not shown in Table 4-1, "The Type4 Keyboard". *Text* may contain escape strings, such as `\E` for escape, `\n` for newline, and `^X` for **Control-X**. See `termcap` in the *SunOS Reference Manual* for the format of escape strings recognized by tty windows.

A tty-based program can send special escape sequences, shown in Figure 4-7, to perform certain operations. These escape sequences may also be sent using the `mapo` function shown above. The escape sequences are received by the tool that owns the tty window, not by the command shell that's running in the tty window.

Figure 4-7 *Tty Escape Sequences*

<code>\E[1t</code>	– open
<code>\E[2t</code>	– close (become iconic)
<code>\E[3t</code>	– move, with interactive feedback
<code>\E[3;TOP;LEFTt</code>	– move, to TOP LEFT (pixel coordinates)
<code>\E[4t</code>	– stretch, with interactive feedback
<code>\E[4;WIDTH;HTt</code>	– stretch, to WIDTH HT size (in pixels)
<code>\E[5t</code>	– front (expose)
<code>\E[6t</code>	– back (hide)
<code>\E[7t</code>	– refresh
<code>\E[8;ROWS;COLSt</code>	– stretch, to ROWS COLS size (in characters)
<code>\E[11t</code>	– report if open or iconic by sending <code>\E[1t</code> or <code>\E[2t</code>
<code>\E[13t</code>	– report position by sending <code>\E[3;TOP;LEFTt</code>
<code>\E[14t</code>	– report size in pixels by sending <code>\E[4;WIDTH;HTt</code>
<code>\E[18t</code>	– report size in characters by sending <code>\E[8;ROWS;COLSt</code>
<code>\E[20t</code>	– report icon label by sending <code>\E]Llabel\E\</code>
<code>\E[21t</code>	– report frame header by sending <code>\E]llabel\E\</code>
<code>\E]l<text>\E\</code>	– set frame header to <i><text></i>
<code>\E]I<file>\E\</code>	– set icon to the icon contained in <i><file></i> ; <i><file></i> must be in <code>iconedit</code> output format
<code>\E]L<label>\E\</code>	– set icon label to <i><label></i>
<code>\E[>OPT;...h</code>	– turn OPT on (OPT = 1 => pagemode), e.g., <code>\E[>1;3;4h</code>
<code>\E[>OPT;...k</code>	– report OPT; sends <code>\E[>OPT1</code> or <code>\E[>OPTk</code> for each OPT
<code>\E[>OPT;...l</code>	– turn OPT off (OPT = 1 => pagemode), e.g., <code>\E[>1;3;4l</code>

The following aliases use escape sequences. You can type them from the keyboard, or you can put them into your `~/ .cshrc` file:

```
# dynamically set the frame header of the tool:
alias header 'echo -n "[l\!*^\\"'
# dynamically set the label on the icon:
alias iheader 'echo -n "[L\!*^\\"'
# dynamically set the image on the icon:
alias icon 'echo -n "[I\!*^\\"'
```

For example, the header alias lets you put the current working directory into the frame header by typing:

```
venus% header pwd
```

Sample .ttyswrc File

The following figure contains an example of a `.ttyswrc` file that you can copy into your home directory and modify.

Figure 4-8 *Example of a .ttyswrc File*

```
venus% cat .ttyswrc
#
#   ttywindow startup file
#
set      pagemode
# Move:
mapo    T3      E[3t
# Stretch:
mapo    T4      E[4t
# Refresh:
mapo    T5      E[7t
# Move (non-interactive) to top left:
mapo    T6      E[3;1;1t
# Stretch in chars (non-interactive) to half high:
mapo    T7      E[8;10;80t
# Stretch in chars (non-interactive) to normal size:
mapo    T8      E[8;34;80t
# Mail:
mapi    R1      dt
mapi    R2      s +planning\ndt\n
mapi    R3      s +bugs\ndt\n.sp
# Editing:
# Bracket word in italic escapes while in vi:
mapi    R4      iEeaE
# Bracket word in bold escapes while in vi:
mapi    R5      iEeaE
venus%
```

Text Windows and the .textswrc File

Whenever you bring up a text window, SunView looks for a `.textswrc` file in your home directory. This file lets you associate filters to unused function keys. (Filters are programs or routines.) When a text selection has been made and a function key pressed, the selected text is piped through the filter assigned to that key. The output is then piped back into the text at the caret. If the selection is pending-delete, the output from the filter will replace it.

A simpler alternative is to add filters to the Extras⇒ pull-right menu by modifying the `.text_extras_menu` file, as described in Chapter 8, “All about Menus”.

Function Keys That Cannot Be Mapped

The keys listed in the following table may not be remapped.

Table 4-2 *Keys Not Available for Remapping in .textswrc*

<i>Right-Handed Keyboard</i>	<i>Left-Handed Keyboard</i>
L1-L10, F1	L1, F1 R1, R4, R7, R10, R13, R3, R6, R9, R12, R15.
With arrow keys enabled: R8, R10, R12, R14.	With arrow keys enabled: R2, R5, R8, R14.
With new accelerators enabled: R7, R11, R13.	

To change arrow keys and new accelerators defaults, use the `Compatibility/New_keyboard_accelerators` and `Input/Arrow_Keys` options in the Defaults Editor.

CAUTION Do not remap the arrow keys without setting `Input/Arrow_Keys` to `No`.

Special Filters

There are a number of special filters, documented in `textedit_filters(1)`, that are provided especially for SunView users.

- `insert_brackets`,
- `remove_brackets`,
- `align_equals`,
- `shift_lines`.

Note, however, that any shell command that uses characters as input and output can be used as a text subwindow filter.

Format of the .textswrc File

List the sample file, `/usr/lib/.textswrc`, to see an example of filters used with a right-handed keyboard. You can copy the sample file into your home directory and modify it.

Here is the basic format of each `.textswrc` file entry.

```
key-name FILTER
command-line
```

A function is assigned to one of the function keys by including in the `.textswrc` file a statement like:

```
/*
 * Note that:
 * insert_brackets /* */ does NOT work
 * double quotes are needed
 */
KEY_TOP(10) FILTER
insert_brackets "/* " " */"
```

This example shows how to include C language comment markers around a piece of text. You could enter this example into your `.textswrc` file, and save the file. Then, to use this added function, bring up a new `textedit` tool. The changes only become effective when a tool is started. Then select the text pending-delete and press the key. The text will be replaced with a copy of itself surrounded with `/*` and `*/`. Note the C-like syntax of comments in `.textswrc`.

The following example enters the octal value for characters in the desired string.

```
KEY_TOP(10) FILTER
insert_brackets "\057*\040" "\040*\057"
```

You might want to add a filter to remove comments:

```
KEY_TOP(10) FILTER
remove_brackets "/* " " */"
```

Several filters shown below are handy for `troff` users.

For `troff` *italics*

```
R(1) FILTER
insert_brackets "\fI" "\fP"
```

Several `troff` command pairs,
each on it's own line

```
KEY_TOP(12) FILTER
insert_brackets ".BS\n.LS\n" "\n.LE\n.BE"
```

The next group shows a variety of parentheses and quotes used.

Parentheses

```
/*
 * Note: insert_brackets "(" ")" also works, and
 *       insert_brackets ( )      also works
 */
```

```
KEY_RIGHT(4)    FILTER
insert_brackets \( \)
```

```
/*
 * Note: insert_brackets "\" "\"" does NOT work
 *       surrounding double quotes are not needed
 */
```

Quotes

```
KEY_RIGHT(5)    FILTER
insert_brackets \" \"
```

```
/*
 * Note: insert_brackets "\\\" "\\\" does NOT work
 *       surrounding double quotes are not needed
 */
```

```
KEY_RIGHT(9)    FILTER
insert_brackets \\\" \\\"
```

Finally the next example uses the `ls` command to obtain the listing of the current directory and pipe it into the text subwindow after some formatting.

```
KEY_TOP(10) FILTER
ls | insert_brackets "*List*\n" "*End*" | shift_lines 4
```

This can be done with `awk` or `sed` scripts to great advantage.

```
KEY_TOP(10) FILTER
awk -f ~/me/mydirectory/myscript
```

For example,

```
KEY_RIGHT(5)    FILTER
insert_brackets \( \)
```

When you press **(R5)**, the `insert_brackets` command is executed putting parentheses around a pending-delete selection.

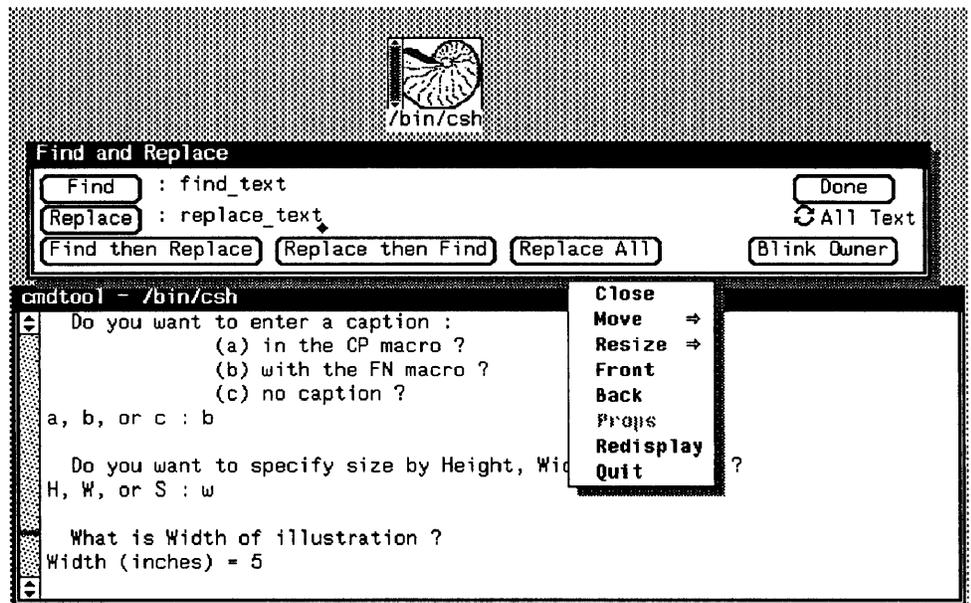
The filter you specify must appear in a directory accessible by your search path. If the command is not found, or if there is some other failure in the invocation of the filter, no change occurs in the document.

All About Frames and Windows

5.1. Anatomy of a Frame

Every tool that runs in SunView has one or more frames. An open frame contains at least one window. A closed frame appears as an icon. The following figure shows a Command Tool frame open and another Command Tool frame closed to an icon. The **Find and Replace** frame is a *pop-up frame* belonging, in this case, to the open Command Tool frame. (You can bring up a **Find and Replace** frame from any text window or command window.)

Figure 5-1 *The Parts of a Frame*



Point anywhere in the frame border, then hold down the RIGHT mouse button to bring up the Frame menu.

The perimeter of the frame, known as the *frame border*, is where you bring up the Frame menu.

Frames contain *windows*, where you edit text, type commands, push buttons, draw pictures, or do whatever the application that owns the frame lets you do. There are five types of windows:

- Text windows — for displaying and editing text

- Panels — for controlling applications
- Command windows — for running SunOS commands
- Tty windows — for emulating a tty dialog
- Canvases — for displaying images

The types of windows are described later in this chapter.

When the pointer is in a window, the window can receive input from the keyboard and is said to have the *keyboard focus*. The window that has the keyboard focus has its border thickened to a bold black line.

The Frame Header

The wide black stripe at the top of the frame is called the *frame header*.

The left side of the frame header contains the name of the tool that “owns” the frame. The rest of the frame header contains valuable information about the frame. For example, the Text Editor frame header tells you:

- The name of the current file — or (NONE)
- Whether the current file has been edited since it was last saved
- The default directory — the directory that is searched to store, load, or include a file with a relative pathname.

As soon as you edit in the Text Editor's edit window, (edited) appears in the frame header. If the Text Editor frame is closed, a greater-than symbol (>) appears in front of the filename on the icon if you've done some editing you haven't yet saved.

Pop-Up Frames

Pop-up frames are usually used for temporary tasks like finding and replacing text. Pop-up frames have **Done** instead of **Quit** in the Frame menu. If its Frame menu contains **Done**, the pop-up frame is actually a *subframe*. Unlike a regular frame, a subframe can't be closed to an icon. The **Find and Replace** pop-up frame is actually a subframe, but the **Reply to or Compose Mail** pop-up frame is a regular frame that you can close to an icon.

5.2. Text Windows

Text windows let you display and edit text.⁶

Many applications are built on text window capabilities, such as Text Editor for files, Command Tool for command interpretation, Mail Tool for mail operations, and Dbx Tool for debugging.

The text window is where you bring up the Text menu, which provides operations for loading and saving files, editing, changing how text is displayed, and finding and replacing text.

⁶ Documentation on the programming interface for text windows is found in the *SunView Programmer's Guide*.

CAUTION Nothing prevents you from editing a file in more than one window at the same time, but this is not a good idea! If you change the file in one window and then go on editing in another window, you will destroy the integrity of the file. This is true whether you change the file using Text Editor, vi, or by any other means in text windows, command windows, or tty windows.

Scratch Windows

Some applications contain a special kind of text window called the *scratch window*, which provides a work area for the application. The scratch window has all the editing capabilities of a regular text window, since it has its own Text menu, but some of the file capabilities are disabled.

You can't choose Load File in a scratch window, and you can't choose Set Directory. These features, which update the frame header, are reserved for regular text windows.

You *can* choose Store to Named File to save the contents of the scratch window. When you want to retrieve them, choose Include File from the File⇒ pull-right menu in the Text menu.

Another limitation of the scratch window is that it doesn't change the frame header when you save or include a file. This is actually an advantage, since you probably are much more concerned about the state of the document you're editing than about the electronic scratch pad.

Changing the Scratch Window Size

You can specify the number of lines in the Text Editor's scratch window by using the Defaults Editor. The default is one line.

You can also increase the number of lines in the scratch window with the Resize accelerator.

- Move the pointer to the bottom edge of the scratch window. The pointer changes to a target.
- Hold down the **(Control)** key.
- Hold down the MIDDLE mouse button and drag the scratch window downward.
- When the scratch window is big enough, release the mouse button and the **(Control)** key.

Changing Direction with the Shift Key

The **(Shift)** key often lets you reverse an action or apply the action in the opposite direction. For example: **(Shift-Find)** means search in the Backward direction. **(Shift)** in scrolling means scroll with respect to the bottom of the window, rather than the top. **(Shift-Delete)**, **(Shift-Control-W)**, and **(Shift-Control-U)** erase the *next* (not previous) character, word, or line.

Mousing Ahead

You don't have to wait for an operation to be completed before you start the next one; you can *mouse ahead*. As you become more adept at working in SunView, you will probably start to mouse ahead more and more.

However, you may not get the results you expect when you mouse ahead, since the text window doesn't immediately figure out the location where you click — it finishes processing the first operation first. By the time it processes the click, the display may have changed, causing a different selection from the one you thought you made when you moused ahead.

You can cut a word in one line, then mouse ahead and cut a word in the next line before the display has been updated to reflect the first cut. It doesn't matter when the display of the first line is updated — there won't be any effect on the next line.

If you mouse ahead and make a cut *later in the same line*, the first cut causes the rest of the line to move left and your second selection may be misinterpreted.

To mouse ahead safely in a text window, do the operations bottom to top and right to left. Then you can be sure of the results.

Filtering Selections

A *filter* is a command that transforms text. For example, `indent` and `capitalize` are text filters. There are three ways to filter text in SunView:

- You can add a text-filtering command to the **Extras**⇒ pull-right menu, which already contains a standard set of text filters. See Chapter 8, “All about Menus”, for information on how to add filters to the **Extras**⇒ pull-right menu.
- You can use the `get_selection` command on a command line to pass the selection to a text filter such as a spelling checker. `get_selection` sends the selection to a filter and inserts the resulting output at the caret. If it's a pending-delete selection, the output from the filter replaces it. See Chapter 6, “All about Selections”, for a discussion of pending-delete selections and the `get_selection` command.
- You can associate a text filter with an unused function key. See Chapter 4, “All about Keyboards” for information on how to “map” a function key to a filter.

5.3. Command Windows

A *command window* is a text window that includes a command interpreter and lets you bring up a Text menu. Command Tool displays a command window.

The Command Line

The *command line* is the line that is communicated to the command interpreter. It's usually the last line of text in the command window. When you press the **(Return)** key, the text on the command line is sent to the command interpreter, which runs the command you requested. The command that's running often sends messages back to the command window. This conversation is known as a *dialog*. Generally the command line begins with a *prompt* like the `venus%` prompt seen in this manual. The command interpreter prompts you when it's ready to receive the next command (but you can type ahead).

The text on the command line in a command window can be edited just the way you edit in a text window — you can cut, paste, copy, and use secondary selections. (In a tty window, by contrast, editing is limited.)

The text that's *not* on the command line is normally read-only. You can't edit it unless you specifically ask to do so by using the **Cmd Modes**⇒ pull-right menu in the Text menu. **Cmd Modes**⇒, which isn't found in text window Text menus, has special features specific to command windows — it lets you turn editing and scrolling on and off.

Changing Command Modes

The Text menu in a command window contains a pull-right menu that doesn't appear in text window Text menus: the **Cmd Modes**⇒ pull-right menu.

Normally you can edit only the command line. The **Cmd Modes**⇒ pull-right menu lets you change the transcript in the command window by editing any text in the window. The *transcript* is a record of the interactive dialog between the user and the shell.

You can also run tty-based programs in a command window. Well-behaved tty-based programs, like `vi`, cause the scrollbar to disappear when they start, and restore it when they're done. If you want to run a program that doesn't do this for you automatically, the **Cmd Modes**⇒ pull-right menu lets you disable and reenable scrolling in the command window. See Chapter 8, "All about Menus", for more information on **Cmd Modes**⇒.

5.4. Panels

A panel is a window that lets you control an application by means of various *panel items* such as panel buttons, choice items, and panel text items.

Blink Owner

Panel buttons are used to specify an action. Nothing happens until you "push" the panel button by clicking LEFT or choosing from its menu. The panel button turns gray while the action is taking place.

Choice items are used to set options. You can set a number of choice items before choosing an action. There are many ways to depict a choice in a panel, including:



- Cycle items.



- Choice buttons.



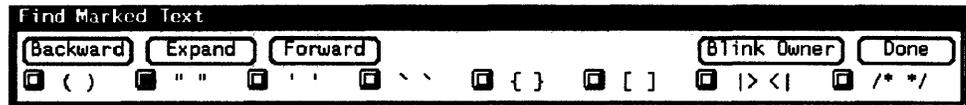
- Check boxes.



- Sliders. When you are dragging a slider, the slider will continue to move even if the pointer slides off the slider until you release the button. This action takes place as long as the mouse stays within the panel or window the contains the slider.

The way choices are shown depends on the application. For example, the **Find Marked Text** pop-up frame (shown in the figure below) contains choice buttons, while the **Defaults Editor Control** panel uses cycle items.

Figure 5-2 *Choice Buttons in the Find Marked Text Frame*



Editing Panel Text Items

You can edit a panel text item in much the same way you would any other text window. You can copy and paste between text windows, command windows, and panel text items. You can use the quick-editing features of secondary selections with **Copy**, **Paste**, and **Cut**.

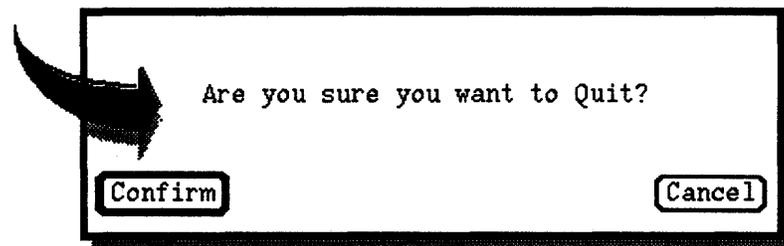
For more information about secondary selections, see Chapter 6, "All about Selections". For more information about text-handling differences among windows and panels, see Table 6-3, "Text Handling in Windows and Panels".

Alert Windows

Alert windows are actually panels and work the way other panels do. A large black arrow sweeping in from the left readily identifies a window as an *Alert* window.

Alert windows contain a message and one or more *buttons* that you push.

Figure 5-3 *Alert Buttons*



The arrow alerts you to read the question or message in the window, and respond accordingly. Often the buttons are **Confirm** and **Cancel**, but there may be more than two, depending on the message.

The button that represents the *yes* choice has a double outline. The pointer is automatically moved over this button for you.

Accelerators for Alert Buttons

If the alert button you want is labeled **Confirm**, or another obvious *yes*. You can press the **[Return]** key instead of pushing the button.

The alert button that's activated by the **[Return]** key is accented by a double outline, like the **Confirm** button in the preceding figure.

If the alert button you want is labeled **Cancel**, or another obvious *no*, you can press the **[Stop]** function key instead of pushing the button.

5.5. Tty Windows

Tty (for *teletype*) windows don't give you all the bells and whistles of SunView, but some applications can't run in any other environment. Shell Tool provides an environment for applications that need a tty window. You can also request a tty window in Command Tool by choosing **Cmd Modes⇒Disable Scrolling** from the Text menu.

A tty window is essentially a command window without scrolling. Editing in the tty command line is limited. The only deletion allowed in the command line is backspacing from the end of the line. You can make secondary selections in a tty window and you can store the primary selection on the Clipboard, as in text windows. You can also copy text between tty windows and text windows.

The Block Cursor

The tty window has its own insertion point, called a *block cursor*. It's a black rectangle when the tty window has the keyboard focus. When the focus is elsewhere, it's an outline of the cursor's rectangle. Like the caret in a text window, the block cursor represents the insertion point. Unlike the caret, the block cursor never blinks.

When you run a command in Command Tool that requires a tty window, the change is usually made for you automatically, and when the command has completed, it changes the window back. For example, when you run `vi` in Command Tool, the scrollbar disappears. When you type `:q`, the scrollbar returns.

Scrollbar Not Restored

If you need to run a non-standard program that doesn't do its own setting up, you can change Command Tool to a tty window yourself by choosing the **Cmd Modes⇒Disable Scrolling** from the Text menu. To change back, choose **Enable Scrolling** from the Tty menu.

5.6. Canvases

A *canvas* is a window that displays unformatted images. It's used for drawing. For example, the Icon Editor contains a canvas where you can draw and erase images by using the mouse. For more information about the Icon Editor, see Chapter 11, "All about Tools" and the `iconedit` command in the *SunOS Reference Manual*.

5.7. Managing Multiple Frames

Deciding how to arrange your SunView screen is as personal as deciding how to arrange your living room. Everyone has a slightly different idea of what works best. (And a few people have *radically* different ideas.) A few things to keep in mind are discussed below.

Move⇒ and **Resize⇒** in the **Frame** menu let you rearrange frames to suit your needs.

Back and **Front** in the **Frame** menu expose and hide overlapping frames.

Speed vs. flexibility.

Services⇒Save Layout in the **SunView** menu saves the current layout of the screen.

Redisplay All in the **SunView** menu refreshes the entire screen. **Redisplay** in the **Frame** menu refreshes the display of the frame.

You can shape frames to suit your needs under different circumstances. The **Console** is an example of this. It's quite small, and it's been placed out of the way. You will probably have at least one other **Command Tool** open, and you will want it large enough so you can comfortably list directories and display files.

Overlapping frames are very flexible in **SunView**. You can move the keyboard focus to a frame that's overlapped and type, paste, save or store the file, do all the operations that the application lets you do, even if you can't see everything because part of the window is hidden.

There's a price to pay, though. Overlapping frames take longer to redraw when you open or close one of them or move one of them. If you're concerned about speed, minimize your use of overlapping frames.

Once you get an arrangement that's convenient, it's nice to have a way to get back to it if you happen to change it temporarily, or if someone uses your workstation, or....

Save Layout remembers the current arrangement by storing it in the `.sunview` file, which is read when you start **SunView**. If you have favorite screen arrangements for different tasks, you can save one, rename it `.sunview_debugging`, save another, rename it `.sunview_writing`, and so on.

Then, the next time you restart **SunView**, choose the arrangement you want by specifying the `-s` option and the filename:

```
venus% sunview -s .sunview_writing
```

It may happen that the screen display is corrupted by a malfunctioning application (for instance, when you're testing a new **SunView** program). You can request a redisplay to refresh the entire screen or an individual frame. You probably won't have to do this very often, but it's there when you need it.

All about Selections

This chapter covers the following information:

- The mouse and the pointer.
- The caret and the block cursor.
- Selecting text and extending a primary selection.
- Erasing text.
- Copying, pasting, and cutting selections.
- Pending-delete selections.
- Quick editing with secondary selections.
- The `get_selection` command.

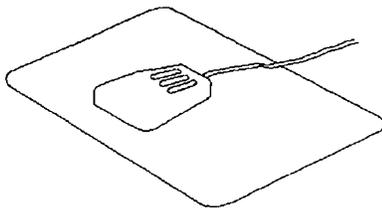
When you select text, you identify it to the system so that it can be recognized for use in editing and other operations.

6.1. Mouse, Pointer, Caret, and Block Cursor

Here's a quick summary of terms.

The mouse is used to move the pointer on the screen. To make a selection, you move the pointer with the mouse and click `LEFT`. The caret marks the insertion point in a text or command window. You set the caret by making a selection. The block cursor marks the insertion point in a tty window. In a tty window the block cursor is always at the end of the command line.

About the Mouse



The *mouse* translates physical motion into electronic signals so that you can move the *pointer* on the screen. For example, you use the mouse to point to text in a window that you want to select.

The mouse sits on a *pad*, a silver plate with a blue grid. The *tail* of the mouse is a wire connected to the back of your keyboard. The mouse should be positioned so that its short side is parallel to the long side of the pad, with the red light facing down and the three lozenge-shaped mouse buttons facing up.⁷

⁷ This description of the optical mouse doesn't apply to the mechanical mouse. If you have a mechanical mouse with a rolling ball on the bottom, you don't have to align the mouse to the pad.

Grip the sides of the mouse with your thumb and little finger, with your fingers resting on the mouse buttons. Move the mouse in circles on the pad and the pointer makes circles on the screen.

About the Pointer

You position the pointer over SunView objects to make selections, choose menu items, push panel buttons, and so on.

The pointer's shape changes as you move it. The shape gives you information about where the pointer is positioned and what action is taking place.

Each pointer has a *hot spot* — the place on the pointer that actually does the pointing. For example, the hot spot in an arrow pointer is the tip of the arrow, and the hot spot in a target pointer is the middle of the target. The pointers are designed so that you'll intuitively use the hot spot correctly.

Here are some standard SunView pointer shapes (some applications may use additional shapes).



The basic pointer shape, which you see when you bring up SunView, is the “northwest” arrow. This is the shape that appears in most windows and on the background.



The pointer changes to a target when it's over an icon, a frame header or any part of the frame border. You can't bring up the Frame menu unless you see a target.



When a tool is busy with an action, such as retrieving new mail in Mail Tool, the pointer in its frame may change to an hourglass. This means the tool won't respond to you until it's finished, but you can mouse ahead or type ahead. You can also continue working in other frames where the pointer shape is normal.



When you choose **Enable Page Mode** from the tty menu, the pointer in the tty window changes to a stop sign whenever there's more than one windowful of text to display. The stop sign means, “The display has stopped but there's more to come.” Also, if you type `Ctrl-S`, the cursor will change to the stop sign. To change the cursor back to the original shape type `Ctrl-Q`.



The right arrow pointer appears when you press `RIGHT` to bring up a menu.

Scrollbar Pointer Shapes



The pointer changes to a diamond when it's in a scroll box.



The pointer changes to a right-facing arrow when the pointer is in the scrollbar (anywhere except the scroll boxes).



When you click or hold down the LEFT or RIGHT mouse button, during scrolling the pointer changes to an up or down arrow to tell you the direction of scrolling.



When you click or hold down the MIDDLE mouse button in any part of the scrollbar, during scrolling the pointer changes to a right-facing triangle.

About the Caret

The *caret* is a symbol that appears in text windows, command windows, and panel text items to indicate where the next insertion will go. It sits just below a line of text and points upward *between adjacent characters*.



When the keyboard focus is in the window, the caret is black, triangle-shaped, and blinking, unless you have disabled blinking.



At other times, the caret is gray, diamond-shaped, and non-blinking.

The terms *caret* and *insertion point* can be used interchangeably. The caret is always positioned at the current insertion point. To set the insertion point in text, move the pointer to the location you want and click the LEFT mouse button. The caret moves to the new location and makes a one-character selection. You can extend the selection by moving the pointer to the end of the additional data and clicking the MIDDLE mouse button.

After you type or paste text:

- The text is where the caret was.
- The caret is at the right of the inserted text.
- There's no selection.

About the Block Cursor

The *block cursor* is found only in tty windows. It represents the insertion point, which in a tty window is always at the end of the command line.



Unlike the caret, the block cursor sits *on* the line, not below it, and it never blinks. Its shape is a tall rectangle — solid black when the keyboard focus is in the tty window, outlined when it's not.



To make a selection in a tty window, move the pointer to the location you want and click the LEFT mouse button. You can then extend the selection by moving the pointer to the end of the additional data and clicking the MIDDLE mouse button.

6.2. Primary Selections

Selections are a way of specifying operands for Text menu operations like **Load File**. Selections also act as operands for some function key operations, like **Cut**, and for some keyboard accelerators, like **⌘-S**. (See Chapter 4, “All about Keyboards”, for information on the Meta (⌘) keys.)

```
primary
selection
```

Primary selections are normal SunView selections. They're highlighted in inverse video (black on white) and last until you type, paste, or make another selection.

Here are the basic operations you need to know when working with text in SunView.

Selecting a Character

To select a single character in a text window or command window:

- Move the pointer over the character and click the LEFT mouse button. The one-character selection is highlighted. Notice that the caret is to the right of the character selected.

```
select a candidate
```

In a panel text item, the entire text item is selected when you click LEFT.

Extending the Selection

To select more than one character:

- Select the first or last character by clicking LEFT on it.
- *Extend* (or *adjust*) the selection by pointing to the other end of the span and clicking MIDDLE.

```
select a candidate
```

Alternatively, you can *wipe* to make and extend a selection. Many personal computers work this way. Wiping works only when there's no selection in the window.

To make a selection by wiping:

- Hold down the MIDDLE mouse button to start the selection.
- Move the pointer to the end of the text you want to select, and release the mouse button.

You can extend a selection as many times as you like, and you can adjust either end of the selection. When you hold down MIDDLE, the end of the selection closest to the pointer is adjusted. You can make a selection larger or smaller when adjusting it.

The Insertion Point

When you click the LEFT or MIDDLE mouse button, two things happen:

- One or more characters are selected.
- The insertion point is specified.

The *insertion point* is the place where the next character you insert will go. The *caret*, a blinking triangle, represents the current insertion point. When you make a selection, the caret appears at the end of the selection closest to the pointer.

The insertion point is always *between characters*. For example, to place the insertion point between the letters m and u in the word *mouse*, position the pointer anywhere between the middle of m and the middle of u, then click LEFT.



I can always get a pointer from my mouse.

To insert text, just type. Here's what happens if you type o:



I can always get a pointer from my mouse.

6.3. Basic Editing

The basic editing operations are:

- *Insert* — Type or paste.
- *Erase* — Remove text from the document.
- *Delete* — Remove text and place it on the Clipboard.
- *Duplicate* — Make a copy of text at another location.
- *Move* — Remove text from its present location and place it somewhere else.
- *Replace* — Delete text and move or duplicate other text to take its place.

Text windows let you do any kind of editing. Command windows, tty windows, and panel text items are subject to a few restrictions, as summarized in Table 6-3, “Text Handling in Windows and Panels” later in this chapter.

Inserting Text

To insert characters in a text window, simply place the caret where you want the characters to go and start typing. Unlike *vi*, for example, the Text Editor has no special insert mode — it is always in insert mode.

Another way to insert text is to paste the contents of the Clipboard at the caret by choosing **Edit**⇒**Paste** from the Text menu.

Erasing a Character

Erasing usually works on the text immediately to the left of the caret. To erase the previous character, press the **Delete** key.

You can erase the character immediately to the *right* of the caret by pressing **Shift-Delete**.

NOTE: **Shift-Delete** means hold down the **Shift** key while you press the **Delete** key.

The **Shift** key often *toggles* — reverses the direction of an operation — especially when used in conjunction with text function keys.

Erasing Words

To erase the word to the left of the caret, press **Control-W**.

To erase the word to the right of the caret, press **Shift-Control-W**.

Erasing Lines

To erase the current line leftward from the caret to the beginning of the line, press **Control-U**. If the caret is already at the beginning of the line, **Control-U** has no effect.

To erase the current line rightward from the caret to the end of the line, press **Shift-Control-U**. If the caret is already at the end of the line, **Shift-Control-U** has no effect.

Deleting Text

To delete text, select it, then choose **Edit⇒Cut** from the Text menu or press the **Cut** function key. You can also use the **⬥-X** keyboard accelerator. The use of the Meta (⬥) keys is described in Chapter 9, “All about Accelerators”.

Cut is a deleting operation. Unlike erasing, deleting “remembers” the text that you removed, so you can reinsert it. Notice that the **Delete** key does not store or remember the text.

Whenever you cut a selection, it's automatically stored on the Clipboard for later use. To see the contents of the Clipboard, choose **Edit⇒Show Clipboard⇒** from the Text menu. (For more information about displaying the Clipboard, see the topic “Show Clipboard⇒” in Chapter 8, “All about Menus”.)

Reinserting Deleted Text

Edit⇒Paste copies the contents of the Clipboard to the insertion point. To reinsert text you just deleted:

- Press the **Paste** function key, or choose **Edit⇒Paste** from the Text menu, or press **⬥-V**. The deletion is inserted at the caret.

Deleting Line Breaks

When you erase a *newline character*, you join two lines into one. The newline character is inserted in a document when you press the **Return** key or the **Line Feed** key.

6.4. Copying, Pasting, and Cutting

Editing frequently requires copying and pasting or cutting and pasting. You can perform these operations in any of the ways shown in the following table. Here are a few examples:

- When you have a long pathname in a command window that you don't want to retype, you can copy and paste.
- When you want to duplicate a format you've used earlier in a document, you can copy and paste in a text window.
- You can cut and paste in a Mail Tool composition window to move a paragraph from one place to another.

Copying doesn't affect the text you copy. **Cut** removes the selection from its current location. **Paste** puts it in a new location. Copying and cutting both put the selection on the Clipboard. **Paste** inserts the contents of the Clipboard at the caret.

Table 6-1 *Copy, Paste, and Cut — Alternative Methods*

<i>Menu Item</i>	<i>Function Key</i>	<i>Menu Accelerator</i>
Edit⇒Copy		
Edit⇒Paste		
Edit⇒Cut		

The examples in this chapter use the function keys to copy, paste, and cut, but you can use any of the methods shown in the table above. The Meta (♦) keys are described in Chapter 9, “All about Accelerators”.

Copying and Pasting a Selection

The **(Copy)** function key stores selected text on the Clipboard without deleting it from the document. The example shows how to copy the filename `original.limerick` from a listing in the command window to the command line in the same window. (The `head -1` command lists the first line of a file. See the *SunOS Reference Manual* for more information about the `head` command.)

To copy text from one location to another:

- Select the text you want to copy and press **(Copy)**.

```
cmdtool - /bin/csh
venus% cd ~
/usr/nutmeg/sages
venus% ls *lim*
lesson1.limerick      original.limerick      original.limerick%
venus% head -1
```

- Select to set the new insertion point if necessary.
- Press **(Paste)** to copy from the Clipboard to the insertion point.

```
cmdtool - /bin/csh
venus% cd ~
/usr/nutmeg/sages
venus% ls *lim*
lesson1.limerick      original.limerick      original.limerick%
venus% head -1 original.limerick
```

Cutting and Pasting a Selection

The **(Cut)** function key deletes selected text from the document and stores it on the Clipboard. The example shows how to delete the word *Cajun*, together with the space following it, and paste them in front of the word *catfish*.

To move a selection from one location to another:

- Make the selection and press **(Cut)**. The selection is deleted and placed on the Clipboard.

```
textedit - (NONE), dir: /usr/nutmeg/sages
Scratch window
Dear Diary,
This is the third day of my California Yuppie Diet.
Today I ate blue cornmeal pancakes for breakfast,
Cajun sushi for lunch, and catfish for dinner.
Tomorrow, will have sprouts and sun-dried tomatoes.
```

- Select to set a new insertion point if necessary.

```
textedit - (NONE), dir: /usr/nutmeg/sages
Scratch window
Dear Diary,
This is the third day of my California Yuppie Diet.
Today I ate blue cornmeal pancakes for breakfast,
sushi for lunch, and catfish for dinner.
Tomorrow, will have sprouts and sun-dried tomatoes.
```

- Press **(Paste)** to copy from the Clipboard to the insertion point.

```
textedit - (NONE), dir: /usr/nutmeg/sages
Scratch window
Dear Diary,
This is the third day of my California Yuppie Diet.
Today I ate blue cornmeal pancakes for breakfast,
sushi for lunch, and Cajun catfish for dinner.
Tomorrow, will have sprouts and sun-dried tomatoes.
```

Cutting and Typing

As soon as you've cut text, you're ready to start inserting a replacement. The example shows you how to change salad to soup.

To replace a selection by typing:

- Select the text you want to replace.

```
textedit - (NONE), dir: /usr/nutmeg/sages
Scratch window
Dear Diary,
California Yuppie Diet, day 10. Let's see...
tomorrow I'll have watercress salad
```

- Press **[Cut]**. The insertion point is now where the deleted text was.

```
textedit - (NONE), dir: /usr/nutmeg/sages
Scratch window
Dear Diary,
California Yuppie Diet, day 10. Let's see...
tomorrow I'll have watercress
```

- Type the replacement text.

```
textedit - (NONE), dir: /usr/nutmeg/sages
Scratch window
Dear Diary,
California Yuppie Diet, day 10. Let's see...
tomorrow I'll have watercress soup.
```

Copying Text Between Windows

You can copy or move a selection from one window to another window the same way you did within a single window. The example shows you how to copy `watercress salad` from a listing in a Command Tool window into a Text Editor document:

- Make the selection in one window and press **[Copy]**.

```
cmdtool - /bin/csh
venus% cat yuppie_food
pine nuts    gourmet granola    baby carrots
radicchio    watercress salad      purple potatoes
venus%

textedit - (NONE), dir: /usr/nutmeg/sages
Scratch window
Dear Diary,
California Yuppie Diet, day 10. Let's see...
tomorrow I'll have
```

- Move the pointer into the other window. Select to set the insertion point if necessary. (In this case, the caret is already where you want to **Paste**.)

```
cmdtool - /bin/csh
venus% cat yuppie_food
pine nuts    gourmet granola    baby carrots
radicchio    watercress salad      purple potatoes
venus%

textedit - (NONE), dir: /usr/nutmeg/sages
Scratch window
Dear Diary,
California Yuppie Diet, day 10. Let's see...
tomorrow I'll have
```

- Press **Paste**.

The screenshot shows two overlapping windows. The top window is a terminal titled 'cmdtool - /bin/csh'. It displays the command 'venus% cat yuppie_food' and its output: 'pine nuts gourmet granola baby carrots', 'radicchio watercress salad purple potatoes', and 'venus%'. The text 'watercress salad' is highlighted. The bottom window is a text editor titled 'textedit - (NONE), dir: /usr/nutmeg/sages'. It shows a 'Scratch window' with the text: 'Dear Diary,', 'California Yuppie Diet, day 10. Let's see...', and 'tomorrow I'll have watercress salad'. A mouse cursor is positioned over the text, and a small triangle indicates the insertion point.

You can also *move* text from one window to another in the same way, using **Cut** instead **Copy**.

6.5. Undoing Editing

Sometimes you realize that you've just edited something you wish you hadn't. The **Undo** operation is a way of restoring the text window to its previous state. **Undo** undoes whatever editing you did since the last time you either made a selection or performed a **Copy**, **Paste**, or **Cut**.

Each window keeps its own undo information — unlike the Clipboard, which all windows share.

- To undo the most recent editing, choose **Undo Last Edit** from the **Edit**⇒**Undo**⇒ pull-right menu in the Text menu, or press the **Undo** function key. The keyboard accelerator for **Undo Last Edit** is **⌘-U**.

CAUTION There is no way to undo an Undo.

A second **Undo Last Edit** undoes the changes you made since the previous insertion point change. You can continue to undo by “changed insertion point” intervals up one of the following limits, whichever comes first:

- The limit specified in the Defaults Editor (the default limit is 50 intervals).
- The last **Save Current File** or **Load File**.

You may want to undo all your edits at once, back to the last time you saved the file. For example, you save your document and then push the **Replace All** button in the **Find and Replace** frame. Afterwards you discover you inadvertently replaced with dental mint instead of mental dint throughout the document.

- To revert to the state of the file after loading or saving, whichever is most recent, choose **Undo All Edits** from the **Edit**⇒**Undo**⇒ pull-right menu.

6.6. Selecting by Multi-Clicking

Multi-clicking is a short-cut that lets you select a word, a line, or the whole document without having to extend the selection.

NOTE: You can double-, triple- or quadruple-click to select words, lines, or the entire document, respectively.

When you make a selection by multi-clicking, the caret appears at whichever end of the selection is closest to where the pointer was. For example, when you double-click anywhere to the left of the letter `u` in the word `petunia`, the caret appears before the letter `p`.

Using multi-clicking, you can easily position the caret before or after a word, or at the beginning of a line.

When you *click* LEFT, a *character* is selected.

To select a *word*:

- Point at any character in the word and *double-click* LEFT. The main attraction of word selection is speed and convenience — you don't have to carefully select the first and last characters in the word. A *word* is one or more alphanumeric characters plus underscore (`_`) bounded by non-alphanumeric characters, or vice versa. For example, `yankees/mickey_mantle` contains three words (`yankees`, `/`, and `mickey_mantle`), while `yogi_berra_and_joe_dimaggio_and_casey_stengel` contains only one word.

To select a *line*:

- Point anywhere in the line and *triple-click* LEFT.

To select the entire *document*:

- Point anywhere in the window and *quadruple-click* LEFT.

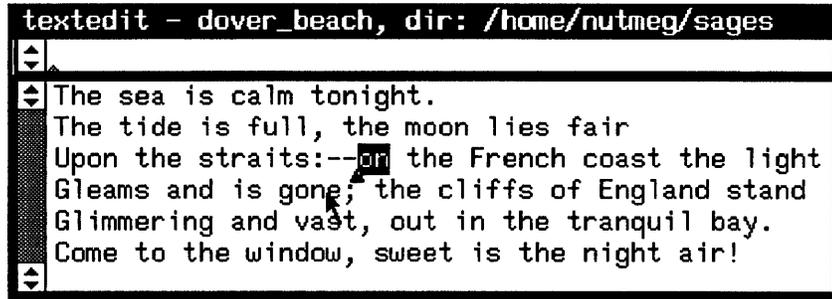
Whether two clicks are interpreted as a double-click or not depends on the time between clicks. You can adjust SunView's sensitivity to multi-clicking by changing the Defaults Editor's `Text/Multi_click_space` and `Text/Multi_click_timeout` settings.

Extending a Multi-Click Selection

When you select a word by double-clicking and extend the selection, the selection is extended in units of a word. Similarly, when you select a line by triple-clicking, the selection is extended in units of a line. This means you don't have to get too precise when pointing to the end of the extended selection, but it also means that the extension is only as precise as the multi-click unit (words or lines).

There *is* a way to adjust a multi-click selection precisely:

- Point to *the very end* of the selection — whichever end you want to adjust.
- Hold down the MIDDLE mouse button and wipe through the selection until it's the way you want it. You can wipe one character at a time.

Figure 6-1 *Extending a Multi-Click Selection*

As shown in the figure above, to select text that begins with the phrase *on the French coast* and ends with the phrase *and is gone*, point anywhere in the word *on*, double-click LEFT, then point anywhere in the word *gone* and click MIDDLE.

No matter how many times you clicked to make the selection, you extend by clicking MIDDLE *once*.

6.7. Secondary Selections

secondary selection

A *secondary selection* is a temporary selection you make while holding down a function key. The action takes place when you release the function key. This feature lets you do many Copy, Paste, Cut, and Find shortcuts with function keys.

The secondary selection is always *underlined*. It lasts only as long as you hold down the function key. During that time, you can adjust the secondary selection the same way you adjust a primary selection.

You can also scroll while making a secondary selection, as long as you continue to hold the function key down. That way, you can adjust the selection to encompass characters that aren't currently visible in the window.

With secondary selections, you can do editing operations that require both a source and a destination, such as duplicating and moving text. Secondary selections also let you perform an operation without changing the current primary selection and insertion point.

You've already learned one method for copying text from one place to another:

- Select some text and press **(Copy)**.
- Select the destination and press **(Paste)**.

For an easier way to duplicate text, use *secondary Paste*. This example changes *strawberry* to *strawberry vanilla*:

- Click LEFT to set the caret where you want the text copied.

```

textedit - ice_cream, dir: /usr
ice_cream
vanilla
coffee
strawberry
chocolate chip
coconut almond fudge
maple banana nut

```

- Hold down the **(Paste)** key while selecting the text to be copied. The text is underlined to indicate that it's a secondary selection.

```

textedit - ice_cream, dir: /usr
ice_cream
vanilla
coffee
strawberry
chocolate chip
coconut almond fudge
maple banana nut

```

- Release the **(Paste)** key. The secondary selection is pasted at the caret.

```

textedit - ice_cream (edited),
ice_cream
vanilla
coffee
strawberry vanilla
chocolate chip
coconut almond fudge
maple banana nut

```

NOTE: When you use secondary **Paste** to get text from another window, be sure the pointer is in the "to" window when you first press the **(Paste)** key; otherwise the keyboard focus will shift and you'll paste in the "from" window. The secondary selection is pasted into the window that has the keyboard focus at the time you press the function key.

You can use secondary selections to specify operands for **[Copy]**, **[Paste]**, and **[Cut]**, as well as for **[Find]**.

To delete text without changing the insertion point, use *secondary Cut*:

- Hold down the **[Cut]** key while selecting the text you want to delete.
- Release the **[Cut]** key. The text is deleted and placed on the Clipboard. The caret remains in its original position.

To set the insertion point for the destination *without losing the primary selection*, use *secondary Copy*:

- Make a primary selection.
- Hold down the **[Copy]** key while setting the caret by clicking LEFT.
- Release the **[Copy]** key. The primary selection is copied to the left or right of the secondary selection, depending on which end you clicked nearest to. The caret moves to the end of the insertion.

Finding a Secondary Selection

To search for a secondary selection, hold down the **[Find]** key while selecting text. The secondary selection is searched for when you release the function key, and the first occurrence that's found is highlighted. The search is in a forward direction, but you can reverse it by holding down **[Shift-Find]** while selecting text. This is a handy way to search without changing the Clipboard or the current primary selection. Note that the search begins at the caret, not at the secondary selection.

Stopping a Secondary Selection

When you start a secondary operation — for example, hold down **[Paste]** and select some text — and then decide you don't want to complete the operation, press the **[Stop]** key to reset the window.

NOTE: If a text window gets stuck so it makes only secondary selections, press the **[Stop]** key to reset the window.

6.8. Pending-Delete Selections



When you want to replace a selection immediately after you make the selection, *pending-delete selections* make your job easier. A pending-delete selection is replaced the next time you type or Paste.

Pending-delete primary selections are *highlighted in dark gray*. Think of the characters as being less visible because they are fading away, soon to be gone altogether.

A pending-delete selection can also be used as an ordinary selection: for example, to specify a filename for **Load File**.

- To make a primary selection pending-delete, hold down the **[Control]** key while selecting or adjusting the selection.

A pending-delete selection is actually deleted only when you insert or type something over it. When you make a new selection without performing any operation or inserting any characters the pending-delete selection isn't affected.

6.9. Quick Editing with Secondary Selections

The use of secondary and pending-delete selections provides a streamlined way of moving, copying, and replacing text. Here's a summary:

- Use pending-delete selections to mark text for deletion (hold down the **Control** key while making the selection).
- Use secondary **Paste** to move or copy text *there-to-here*, in other words, from somewhere else to the caret. When you're working in a tty window such as Shell Tool, secondary **Paste** is the equivalent of **Stuff**.
- Use secondary **Copy** to move or copy text *here-to-there*, in other words, from the caret to somewhere else.
- Use secondary **Cut** to delete text without affecting the caret.
- Use swapping to make the primary selection and the secondary selection change places.

In the techniques described below, *here* means where the caret is, *there* means elsewhere. **Control-Paste** means hold down the **Control** key and the **Paste** key at the same time.

These methods work for text and tty windows as well as panels, although tty windows do not support deletion.

The following table summarizes all the quick-edit techniques that use secondary selections. To use these techniques, first make the primary selection, then make the secondary selection. The techniques are described in detail following the table.

Table 6-2 *Quick Editing with Secondary Selections*

<i>To do this...</i>	<i>Make primary selection...</i>	<i>Make secondary selection using...</i>
<p>DUPLICATE HERE: Copy secondary selection to caret.</p> <p>MOVE HERE: Move secondary selection to caret.</p> <p>DUPLICATE AND REPLACE HERE: Copy secondary selection to replace primary selection.</p> <p>MOVE AND REPLACE HERE: Move secondary selection to replace primary selection.</p>	<p>normal or none (sets caret)</p> <p>normal or none (sets caret)</p> <p>pending-delete</p> <p>pending-delete</p>	<p>Paste</p> <p>Control-Paste</p> <p>Paste</p> <p>Control-Paste</p>
<p>DUPLICATE THERE: Copy primary selection to left or right of secondary selection without moving caret.</p> <p>MOVE THERE: Move primary selection to left or right of secondary selection without moving caret.</p> <p>DUPLICATE AND REPLACE THERE: Copy primary selection to replace secondary selection without moving caret.</p> <p>MOVE AND REPLACE THERE: Move primary selection to replace secondary selection without moving caret.</p>	<p>normal</p> <p>pending-delete</p> <p>normal</p> <p>pending-delete</p>	<p>Copy</p> <p>Copy</p> <p>Control-Copy</p> <p>Control-Copy</p>
<p>DELETE THERE: Delete secondary selection without affecting primary selection or caret.</p>	<p>normal, none, or pending-delete</p>	<p>Cut</p>
<p>SWAP: Exchange locations of primary selection and secondary selection.</p>	<p>pending-delete</p>	<p>Control-Copy , then regular Paste .</p>

Secondary Paste

Secondary **Paste** operations let you edit there-to-here, moving or copying text to the caret.

Duplicate Here

To copy text to the caret:

- Hold down **Paste** while selecting the text. The selected text is underlined.
- Release **Paste**. The text is copied at the caret.

Move Here

To move text from its original location to the caret:

- Press **Control-Paste** while selecting the text. The selected text is underlined and grayed.
- Release **Control-Paste**. The text is removed from its original location and pasted at the caret.

Duplicate and Replace Here

To copy text to replace the primary selection:

- Make the primary selection pending-delete (hold down **Control** while selecting).
- Press **Paste** while selecting the replacement text. The secondary selection is underlined.
- Release **Paste**. The primary selection is deleted and placed on the Clipboard. The secondary selection is inserted where the primary selection was.

Move and Replace Here

To move text from its original location to replace the primary selection:

- Make the primary selection pending-delete (hold down **Control** while selecting).
- Press **Control-Paste** while selecting the replacement text. The replacement text is underlined and grayed.
- Release **Control-Paste**. The primary pending-delete selection is placed on the Clipboard and the replacement text is moved from its former location to where the primary selection was.

Secondary Copy

Secondary Copy operations let you edit here-to-there. The location of the caret doesn't change.

Duplicate There

To copy the primary selection to another location without moving the caret:

- Make a primary selection.
- Hold down **[Copy]**, then click LEFT at the new location. The new location is underlined.
- Release **[Copy]**. The selection is copied to the new location — to the left or right of the underlined text, depending on which end you clicked nearest to. The caret stays where it was.

Move There

To move the primary selection to another location without moving the caret:

- Make the primary selection pending-delete (hold down **[Control]** while selecting).
- Hold down **[Copy]** while selecting the new location by clicking LEFT. The new location is underlined.
- Release **[Copy]**. The primary selection is moved to the new location — to the left or right of the secondary selection (the underlined text), depending on which end you clicked nearest to. The caret stays where the primary selection was. The Clipboard is not changed.

Duplicate and Replace There

To copy the primary selection to replace text at another location without moving the caret:

- Make a primary selection.
- Hold down **[Control-Copy]** while selecting the text to be replaced. The text to be replaced is underlined and grayed.
- Release **[Control-Copy]**. The primary selection is copied, replacing the secondary selection (the text to be replaced), and the secondary selection goes on the Clipboard.

Move and Replace There

To move the primary selection to replace text at another location without moving the caret:

- Make the primary selection pending-delete (hold down **[Control]** while selecting).
- Hold down **[Control-Copy]** while selecting the text to be replaced. The text to be replaced is underlined and grayed.
- Release **[Control-Copy]**. The secondary selection (the text to be replaced) is deleted and placed on the Clipboard. The primary selection moves to where the secondary selection was. The caret stays where the primary selection was.

Secondary Cut

Secondary Cut erases text anywhere in the document without changing the insertion point.

Secondary Cut is useful when you're typing along and notice something you want to delete. It lets you delete the text without moving the caret, so you can delete someplace else and go right on typing.

Delete There

To erase text without moving the caret:

- Press **[Cut]** while selecting the text to delete. The secondary selection is underlined and grayed.
- Release **[Cut]**. The secondary selection is deleted and placed on the Clipboard. The caret stays where it was.

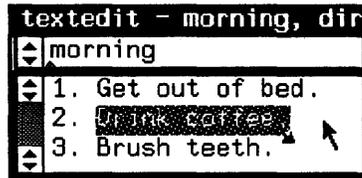
Swapping Text

Here's a fast way to swap text using secondary selections. The example shows you how to exchange items in a list.

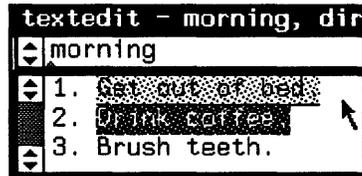
Swap Here and There

To exchange the primary selection with other text:

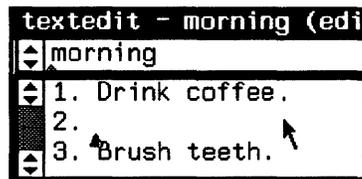
- Make the primary selection pending-delete (hold down **Control** while selecting). You don't have to worry about which item in the swap is which. Just pick either one to be the primary selection.



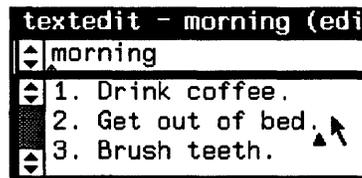
- Press **Control-Copy** while selecting the text to swap. The secondary selection is underlined and grayed.



- Release **Control-Copy**. The secondary selection is deleted and placed on the Clipboard. The primary selection moves to where the secondary selection was. The caret stays where the primary selection used to be.



- Press **Paste**. (This is regular **Paste**, not secondary **Paste**.) The text on the Clipboard — the former secondary selection — is inserted at the caret, completing the swap.



6.10. Text Handling Comparison

The following table contains a comparison of the different types of windows with respect to selections, text operations, and related issues.

Text windows and command windows are similar, since they share the Text menu. In general, whatever editing you can do in a text window can also be done on the command line in a command window. Whatever is not on the command line is normally "read-only," but you can edit all of the command window if you first choose **Enable Editing** from the **Cmd Modes**⇒ pull-right menu.

In tty windows and panels, you can do a subset of text window editing. In these windows you can insert or erase text only at the end of the command line or panel text item. Certain other restrictions apply, as shown in the table.

If you choose **Services**⇒**Remote Login**⇒**Command Tool** from the SunView menu or type the `rlogin` command in a command window, text handling in the command line follows the rules for tty windows in the following table. The rest of the command window follows the rules for command windows — that is, it's read-only unless you choose **Enable Editing**.

Table 6-3 *Text Handling in Windows and Panels*

<i>Description</i>	<i>Text Window</i>	<i>Command Window</i>	<i>Tty Window</i>	<i>Panel Text Item</i>
Insertion point	Caret	Caret	Block cursor	Caret
Menu	Text menu	Text menu with Cmd Modes ⇒	Tty menu	No default menu
Select text	1 or more chars	1 or more chars	1 or more chars	1 or more chars
Insert text	Anywhere	Anywhere in command line	At end of command line	Anywhere in line
Erase text	Anywhere	Anywhere in command line	From end of command line	Anywhere in line
Copy	1 or more chars	1 or more chars	1 or more chars	1 or more chars
Paste	Anywhere	Anywhere in command line	To end of command line	Anywhere in line
Cut	Anywhere	Anywhere in command line	No	Anywhere in line
Undo	Anywhere	Anywhere in command line	No	No
Pending-delete selection	Anywhere	Anywhere in command line	No	Anywhere in line
Secondary selection — source	1 or more chars	1 or more chars	1 or more chars	1 or more chars
Secondary selection — destination	Anywhere	Anywhere in command line	End of command line	Anywhere in line

6.11. Using `get_selection`

The `get_selection` command, which is described in the *SunOS Reference Manual*, gives you lots of nice ways to work with selections.

- To save the selection in a file (especially useful when you want to save part of a mail message):

```
venus% get_selection > filename
```

Piping is a way of stringing commands together. The *pipe* symbol is a vertical bar (`|`). The output of the command on the left of the pipe is passed as input to the command on the right of the pipe.

- To pipe the selection to a print command:

```
venus% get_selection | lpr -J"title for printing"
```

- To pipe the selection to a spelling checker:

```
venus% get_selection | spell
```

- To use the selection — for example, a directory name — on the command line:

```
venus% ls -a `get_selection`
```

The back quotes (`` ``) mean “not the thing, but the meaning of the thing.” In other words, the command line `ls -a `get_selection`` tells the command interpreter to evaluate ``get_selection`` before making the list.

NOTE: When using `get_selection` in a command, you must wait until the command completes before making another selection.

Whenever you type, the current selection is deselected if it's in the same window. Therefore you may lose the selection that you want `get_selection` to work with. You can get around the problem by copying the selection to the Clipboard and using it from there.

- To preserve the current selection, copy the selection to the Clipboard by pressing **Copy**, then type:

```
venus% get_selection clipboard ...
```

All about Scrolling

Often the document you're editing is too large to be entirely visible within the text window. You can resize the window, but most often you'll want to *scroll* the text through the window. To access parts of the document that aren't visible, you scroll to move the document relative to the window.

The information about scrolling in text windows in this chapter also applies to command windows, panels, and all windows with scrollbars that you'll encounter in this manual and in other SunView manuals.

7.1. Anatomy of the Scrollbar



At the left of each text window is a gray column called the *scrollbar*. The light gray area is called the *bar*.

The bar contains a darker gray area called the *bubble*. The bubble indicates the relative size and location of the visible text within the document. For example, if the bubble is about one-fourth the size of the bar, the document is about four windows in length.

At the top of the scrollbar is a square that contains up/down arrows. It's known as a *scroll box*. There's an identical scroll box at the bottom of the scrollbar.

Both scroll boxes operate the same way; the bottom scroll box is a convenience when you're editing near the bottom of the window.

The figures in this manual show the default scrollbar. However, you can change the appearance of the scrollbar, whether it's there all the time, what color it is, what side of the window it's on, how wide or narrow it is, and so on, by changing scrolling options in the Defaults Editor.

7.2. Basic Scrolling

Basic scrolling takes place with the pointer in a scroll box. Direct scrolling takes place with the pointer in the bar.

Scrolling Up or Forward

When you move the pointer into the scroll box, its shape changes to a diamond.

- To scroll the document *up* one line, click LEFT in the scroll box.

The pointer shape changes to an upward-pointing arrow while scrolling, and the document moves through the window.

The line closest to the start of the document disappears at the top of the window and a new line appears at the bottom.

You can think of this as *scrolling forward*.

Scrolling Down or Backward

- To scroll the document *down* one line, click RIGHT in the scroll box.

The pointer shape changes to a downward-pointing arrow while scrolling, and the document moves through the window.

The line closest to the end of the document disappears at the bottom of the window and a new line appears at the top.

You can think of this as *scrolling backward*.

Scrolling a Windowful at a Time

You can scroll at a faster pace by scrolling a windowful at a time. This is sometimes called “paging.”

- To scroll up or forward by one windowful, click MIDDLE in a scroll box.
- To scroll down or backward by one windowful, hold down the **(Shift)** key while you click MIDDLE in a scroll box.

Continuous Scrolling

By holding down the mouse button instead of clicking and releasing, you can automatically repeat any scroll box operation.

7.3. Direct Scrolling

By clicking the mouse buttons in the bar, rather than in a scroll box, you can scroll directly to anywhere in the document. The direction you scroll is controlled by the mouse buttons.

You can use direct scrolling to move a large distance in the document by thumbing, or to make small adjustments in the window by fine tuning.

Thumbing in the Bar

Thumbing lets you flip to a page in the document as you would flip through a dictionary or telephone book. You use the MIDDLE mouse button for thumbing.

- To thumb, click MIDDLE in the bar.

The actual distance scrolled depends on where you click in the bar. Click MIDDLE half-way down in the bar to scroll to the middle of the document, click MIDDLE in the top of the bar to scroll to the start of the document, and so on.

- To undo thumbing, click **(Shift)**-MIDDLE in the bar.

When you undo thumbing, the document scrolls to the last click-MIDDLE, the last Find that caused a scroll, or the last **(Shift)**-MIDDLE, whichever is most recent.

For example, you can thumb to another part of the document, paste or type, and undo thumbing to return to the previous location.

The **(Home)** and **(End)** keys (R7 and R13) are another way of scrolling directly. **(Home)** scrolls to the beginning of the document, and **(End)** scrolls to the end. However, **(Home)** and **(End)** also move the caret, while thumbing leaves the caret

at its current location.

Fine Tuning

You can use direct scrolling to adjust the placement of text in the window. For instance, you may want to see the lines that precede the line at the top of the window.

- To move the line at the top of the window to the pointer, click RIGHT in the bar.

Clicking RIGHT in the bar lets you see preceding lines. The further down in the bar you click, the more preceding lines you'll see.

- To move the line opposite the pointer to the top of the window, click LEFT in the bar.

Clicking LEFT in the bar lets you move a line directly to the top of the window and exposes more text at the bottom of the window.

Hold down the **(Shift)** key to fine tune with respect to the *bottom* of the window. This is especially useful when the upper portion of a window is overlapped by another frame.

- To move the line at the bottom of the window to opposite the pointer, click **(Shift)**-LEFT in the bar.
- To move the line opposite the pointer to the bottom of the window, click **(Shift)**-RIGHT in the bar.

(Shift)-LEFT undoes **(Shift)**-RIGHT, and vice versa.

All about Menus

This chapter describes all the standard SunView menus:

- The SunView menu.
- The Frame menu.
- The Text menu (including command window menu items).
- The Tty menu.

The menu items described in the chapter are in the same order as they appear in the menus, so you can use the menus themselves as an index to the chapter.

8.1. Menus in General

This section discusses features common to all menus in SunView.

How to Bring Up a Menu

You can bring up any menu by holding down the RIGHT mouse button. Which menu pops up depends on where the pointer is:

- The SunView menu pops up when the pointer is over the SunView background.
- The Frame menu pops up when the pointer is over any part of the frame border. (When the pointer is over the frame border, its shape changes to a target.)
- The Text menu pops up when the pointer is in a text window or a command window.
- The Tty menu pops up when the pointer is in a tty window or a command window with scrolling disabled.

When the menu appears, keep holding down the mouse button until you make a choice from the menu or dismiss it. The pointer changes to a right arrow.

- *To choose from a menu*, highlight the menu item you want by moving the pointer over it; then release the mouse button.
- *To dismiss a menu without choosing a menu item*, move the pointer to the left so that no menu item is highlighted; then release the mouse button.
- *To open a pull-right menu*, highlight the pull-right menu item and drag the pointer to the right. The pull-right menu appears.

Occasionally when you open a pull-right menu (say, near the edge of the screen), the pull-right menu completely hides its “parent” menu. To get back to the parent menu, dismiss the pull-right menu by moving the pointer to the left. If the parent menu is only partly hidden, moving the pointer anywhere on the parent menu dismisses its “child.”

Grayed Menu Items

If a menu item is displayed in gray — not black — letters, it's not available to be chosen and it doesn't highlight when you move the pointer over it. Usually a menu item is grayed because it requires a selection and there's no current selection. However, a menu item may be grayed for other reasons. For example, tools that have not implemented the **Props** menu item in the **Frame** menu display it grayed.

NOTE: Make sure the menu item you want is highlighted before you choose it. If you click **RIGHT** on a grayed menu item, your request is ignored.

Defaults in Pull-Right Menus

To choose the first menu item in a pull-right menu, you don't have to open the menu. That's because the first item in any pull-right menu is the default for that menu. Instead of opening the **File**⇒ pull-right menu and choosing **Save Current File** (the first item), you can simply choose the **File**⇒ menu item itself. In other words, with the standard settings in the Defaults Editor, the default for a pull-right menu item is always the first item in its menu. This is true on all levels of the menu.

For example, the defaults for the **Text** menu pull-rights are:

- **File**⇒**Save Current File**.
- **Edit**⇒**Again**.
- **Display**⇒**Select Line at Number**.
- **Find**⇒**Find and Replace** (a pop-up frame).
- **Extras**⇒**Format**.

And the defaults in the **Find**⇒ pull-right menu are:

- **Forward** for the **Find**⇒**Find Selection**⇒ menu.
- **Forward** for the **Find**⇒**Find Clipboard**⇒ menu.
- **Expand** for the **Find**⇒**Replace** |>field<|⇒ menu.

The nice thing about defaults is that you don't have to open the pull-right menu to choose the default menu item.

However, if you change the Defaults Editor settings for `Menu/Initial_Selection` and `Menu/Default_Selection` from `Default` to `Last_Selection`, the default for each pull-right menu becomes whatever you last chose from that menu. If that menu choice isn't possible — for example, if it requires a selection and there isn't one — no action is taken. **NOTE:** `Last_Selection` is very nice if you're careful and experienced, but it can get you into real trouble. If you use this option, be sure you

know what the default is when you choose a pull-right menu without opening it.

Stay-Up Menus

If you get tired of holding the mouse button down while you look at a menu, you can change the `Menu/Stay_up` option in the Defaults Editor from `False` to `True`. Then your menus will appear when you *click* the RIGHT mouse button and disappear when you click it again. With stay-up menus, you click in a highlighted menu item to choose it.

Menus in Panels

Many panel items have menus associated with them. These work just like the standard SunView menus. To bring up a menu in a panel, move the pointer over a panel item and hold down the RIGHT mouse button. The current choice in the menu may be indicated by a check mark.

8.2. The SunView Menu

The SunView menu contains tools and services. A *tool* is a shell, editor, or other application that runs in SunView. To request a service, choose its menu item. To start a tool, choose its menu item.

For additional information about tools in the SunView menu, see Chapter 11, “All about Tools”, and the *SunOS Reference Manual*.

The “Shells⇒” Pull-Right Menu

A *shell* is a window that accepts SunOS commands. The Shells⇒ pull-right menu contains menu items for the following SunView shells:

Command Tool
Shell Tool
Graphics Tool
Console

To bring up a shell, just choose its menu item. You can have several shells open at the same time.

The “Editors⇒” Pull-Right Menu

The Editors⇒ pull-right menu contains menu items for tools that create and revise documents, icons, and fonts, plus a tool that lets you change SunView defaults. The following editors are included:

Text Editor
Defaults Editor
Icon Editor
Font Editor

The “Tools⇒” Pull-Right Menu

The Tools⇒ pull-right menu contains menu items for additional important applications, including:

Mail Tool
Dbx (debug) Tool
Performance Meter⇒
Clock⇒

You can choose either a round or rectangular clock face from the Clock⇒ pull-right menu, which contains icons instead of menu-item names.

The “Services⇒” Pull-Right Menu

The Services⇒ pull-right menu contains a variety of useful options.

“Redisplay All”

Choose **Redisplay All** to refresh the screen. This is useful if the screen image appears to have become corrupted.

“Printing⇒”

The **Printing⇒** pull-right menu lets you check the status of print jobs and print a selection.

- **Check Printer Queue**, the default, lists printer status in the Console window.

- **Print Selected Text** sends the current selection to the printer.

NOTE: Don't change the selection until you see the `Selection` printed message in the `Console` window.

“Remote Login⇒”

The **Remote Login⇒** pull-right menu lets you bring up a new `Command Tool` or `Shell Tool` and connect to a remote host. Type the name of the remote host in response to the prompt.

```
Hostname? remote_host_name
```

When you've finished the session on the remote host, type `exit` to end the session. The `Command Tool` or `Shell Tool` quits automatically.

“Save Layout”

Choose **Services⇒Save Layout** from the `SunView` menu to save the current `SunView` screen layout in the `~/ .sunview` file.

The `.sunview` file.

The `.sunview` file contains the locations of your open and closed frames. Whenever you restart `SunView`, the startup screen is laid out according to the `.sunview` file. For example, if a frame was open when the `.sunview` file was created, it will be opened whenever you start `SunView`.⁸

If the `.sunview` file doesn't exist, the standard startup screen is produced.

Save Layout executes the `toolplaces` command, described in the *SunOS Reference Manual*.

“Lock Screen”

Choose **Lock Screen** to display a randomly moving Sun logo on a black background. This prevents the screen from burning in a constant image. Locking the screen also provides privacy.

To unlock the screen, click any mouse button or press any keyboard key except `(F1)` or `(Caps)`, then type your password and press `(Return)`.

“Exit SunView”

Choose **Exit SunView** to return to the `SunOS` shell.

If there's a serious system problem and you can't bring up the `SunView` menu, you may be able to “bail out” — this works only if the pointer is over the gray background, not over a frame or icon:

- Type `(Control-D)` followed by `(Control-Q)` for an emergency exit of `SunView`. (It's not a good idea to exit `SunView` this way unless you have to.)

⁸ For compatibility with previous releases, if the `.sunview` file isn't found, `SunView` looks for a `.suntools` file from which to restore the screen layout.

8.3. Customizing the SunView Menu

You can modify the SunView menu to add or subtract tools and services. You can also display tools in iconic form, the way the **Tools⇒Clock⇒** pull-right menu is displayed. The standard SunView menu is built from the `/usr/lib/rootmenu` file.

To construct your own SunView menu, you need to:

- Construct a new file containing the menu commands.
- Specify the name of the new file to the Defaults Editor.
- Exit SunView.
- Restart SunView.

When SunView comes up again, you can bring up your new SunView menu.

After you've created the new SunView menu file and restarted SunView, you can make additional changes to the file and see them immediately — you don't have to restart SunView again.

Making a `.rootmenu` File

The following figure shows you the standard SunView menu file.

Figure 8-1 SunView Menu File in /usr/lib/rootmenu

```

textedit - sunview_rootmenu, dir: /usr/nutmeg/sages
#
#
#
@(#)rootmenu 15.7 87/08/19 SMI
#
#
#
sunview root menu
#
#
#
"Shells"          MENU
  "Command Tool"  cmdtool
  "Shell Tool"    shelltool
  "Graphics Tool" gfxtool
  "Console"       cmdtool -C
"Shells"          END
"Editors"         MENU
  "Text Editor"   textedit
  "Defaults Editor" defaultsedit
  "Icon Editor"   iconedit
  "Font Editor"   fontedit
"Editors"         END
"Tools"           MENU
  "Mail Tool"     mailtool
  "Dbx (debug) Tool" dbxtool
  "Performance Meter" MENU
    "Percent CPU Used"      perfmeter -v cpu
    "Ethernet Packets"     perfmeter -v pkts
    "Swapped Jobs"         perfmeter -v swap
    "Disk Transfers"       perfmeter -v disk
  "Performance Meter"     END
  "Clock"                 MENU
    </usr/include/images/clock.icon>  clock
    </usr/include/images/clock.rom.icon> clock -r
  "Clock"                 END
"Tools"                 END
"Services"              MENU
  "Redisplay All"       REFRESH
  "Printing"           MENU
    "Check Printer Queue" sh -c "echo; echo '-----'; echo
'Printer queue'; lpq; echo '-----'^G"
    "Print Selected Text" sh -c "get_selection | lpr ; echo 'Selection
printed'.^G"
  "Printing"           END
  "Remote Login"       MENU
    "Command Tool"     cmdtool csh -c "echo -n 'Hostname? '; exec rlogin $<
$<"
    "Shell Tool"       shelltool csh -c "echo -n 'Hostname? '; exec rlogin
$<"
  "Remote Login"       END
  "Save Layout"        sh -c "mv -f ~/.sunview ~/.sunview-;toolplaces>.sunview; ech
o 'Screen layout saved (Previous layout in .sunview-)"
  "Lock Screen"        lockscreen
"Services"             END
"Exit SunView"        EXIT

```

NOTE: The format of the date on the top line would be 10.14 88/06/09 SMI on a SPARCstation 1.

The name of each menu item is surrounded by double quotation marks (""). Each pull-right menu begins with a MENU statement and ends with an END statement.

NOTE: The SunView menu file must not contain any blank lines. Begin an otherwise empty line with a pound sign (#). It's okay to start a line with a Tab character or space, as long as the rest of the line isn't blank.

For each menu item, supply the name of the menu item in quotes followed by the command you want SunView to execute when the menu item is chosen. If the command has options, you can specify them right after the command name.

The command consists of one or more shell commands. Let's say you have written a program called `peacenow` and you want it to be the default in the **Editors**⇒ pull-right menu. To include your program and its options in the **Editors**⇒ menu, follow these instructions:

- Open `/usr/lib/rootmenu` and add a line before the "Text Editor" line that looks like this:

```
"Peace Tool"    ~/peacenow -make_love -not_war
```

- Save the changed file in your home directory as `~/rootmenu`. (You can use another filename if you prefer — you have to tell the Defaults Editor which name you're using in any case.)
- Bring up the Defaults Editor, cycle to the **SunView** category, and change the `Rootmenu_filename` text item to `~/rootmenu`.
- Push the Defaults Editor's Save button to save the change.
- Exit SunView and restart.

Now you can open the SunView menu and choose **Peace Tool**, which will run with the options you specified.

Icons in the SunView Menu

The "Clock" MENU definition in Figure 8-1 displays icons, not menu item names. You can change the standard SunView menu to display the icons of all the tools it contains, or you can create your own icons using the Icon Editor.

The standard icons are found in the `/usr/include/images` directory.

To replace a menu item name with its corresponding icon:

- List `/usr/include/images` to find the icon you want (or make your own). All of the filenames ending in `.icon` are icons.
- Replace the menu item name in your SunView menu file (`~/rootmenu`, described above) with the name of the icon enclosed in `<>` brackets; for example:

```
</usr/include/images/textedit.icon>
```

Look at Figure 8-1 and see how the icon is specified for `Clock`.

```
</usr/include/images/clock.icon>
```

- Save the `~/rootmenu` file.
- Bring up the SunView menu to see the new icon.

8.4. The Frame Menu

The Frame menu is used to open or close, move, resize, expose, hide, and redisplay frames. It also lets you quit the tool that owns the frame. You can bring up the frame menu over an open frame or over an icon.

- To bring up the Frame menu, move the pointer over the frame header or any other part of the frame border so that the pointer changes to a target. Then hold down the RIGHT mouse button.

“Close” and “Open”

When you bring up a tool, its frame is open and the Frame menu contains the **Close** menu item. **Close** closes the frame to an icon.

The tool can continue to process while in iconic form. For instance, you can compile a program by typing a compiler command in a Command Tool and then closing the Command Tool — the compilation continues.

When the Text Editor is closed, its icon shows the name of the file being edited.

Open toggles to **Close**. That is, when the frame is closed, the Frame menu you bring up over its icon contains **Open** instead of **Close**.

Open opens the frame and places it on the screen in the location it occupied when you last closed it. **Open** does *not* clear the window. If the tool was inactive while closed, the window looks just the way it did before you closed it. If the tool was active, the results of the activity, such as a compilation, are reflected in the window.

Here are the accelerators for **Open** and **Close**:

- To open an icon using the mouse, move the pointer over the icon and click the LEFT mouse button.
- To open an icon using a function key, move the pointer over the icon and press the **(Open)** function key.
- To close a frame, move the pointer anywhere over the frame (including the windows and the frame border) and press **(Open)**. (The **(Open)** function key toggles to **Close**.)

The “Move⇒” Pull-Right Menu

The **Move⇒** pull-right menu lets you specify whether the frame moves on one axis (constrained) or both (unconstrained). All moving and resizing is done using the MIDDLE mouse button.

When you choose **Move⇒**, an alert window displays a message that tells you how to move the frame or icon.

The Bounding Box

When you move or resize a frame, the *bounding box* — an outline the size and shape of the frame — shows you where the frame will be located. The bounding box moves or changes size as you move the pointer while holding down the MIDDLE mouse button. When you release the mouse button, the frame moves to the location of the bounding box.

When you hold down the MIDDLE mouse button, the pointer shape changes to show you the kind of move or resize you’ve requested — the solid outline

represents the frame and the dotted outline represents the bounding box. The pointer shows whether you're moving or resizing, constrained or unconstrained, as described below.

“Move⇒Unconstrained”

Choose **Move⇒Unconstrained**, the default, to move a frame or icon freely on the screen.

Here's the accelerator equivalent for **Move⇒Unconstrained**:

- Grab the frame border *near a corner* by moving the pointer over it, then hold down the MIDDLE mouse button. Continue to hold down the MIDDLE mouse button while you drag the bounding box to the new location. Then release the mouse button.

“Move⇒Constrained”

Choose **Move⇒Constrained** to move a frame or icon in only one direction — left-and-right or up-and-down, but not both. If you want frames precisely lined up, it's helpful to constrain the move.

Here's the accelerator for **Move⇒Constrained**:

- Grab the frame border *near the middle of a side* by moving the pointer over it, then holding down the MIDDLE mouse button. Continue to hold down the MIDDLE mouse button while you drag the bounding box to the new location. Then release the mouse button.

The “Resize⇒” Pull-Right Menu

The **Resize⇒** pull-right menu lets you change the dimensions of a frame.

When you choose **Resize⇒**, an alert window appears with instructions for you to follow. The bounding box shows you the new shape before you release the mouse button.

Resize⇒ is only for frames. You can't change the size of an icon.

“Resize⇒Unconstrained”

Choose **Resize⇒Unconstrained**, the default, to resize a frame freely, changing both dimensions at once, by grabbing near a corner of the frame. The corner you grab moves, and the opposite corner remains fixed. For example, to make a square frame into a smaller square, move the pointer diagonally from any corner of the frame toward the opposite corner.

Here's the accelerator equivalent for **Resize⇒Unconstrained**:

- Hold down the **(Control)** key.
- While holding down the MIDDLE mouse button, grab the frame border near the *corner* you want to stretch or squeeze, and move the pointer toward or away from the opposite corner.
- Release the mouse button and the **(Control)** key when the bounding box location is satisfactory.

“Resize⇒Constrained”

Choose **Resize⇒Constrained** to resize one dimension of a frame by grabbing near the *middle of a side* of the frame. The side you’re grabbing moves, while the opposite side remains fixed. You can change either the height or the width while holding the other constant.

Here’s the accelerator for **Resize⇒Constrained**:

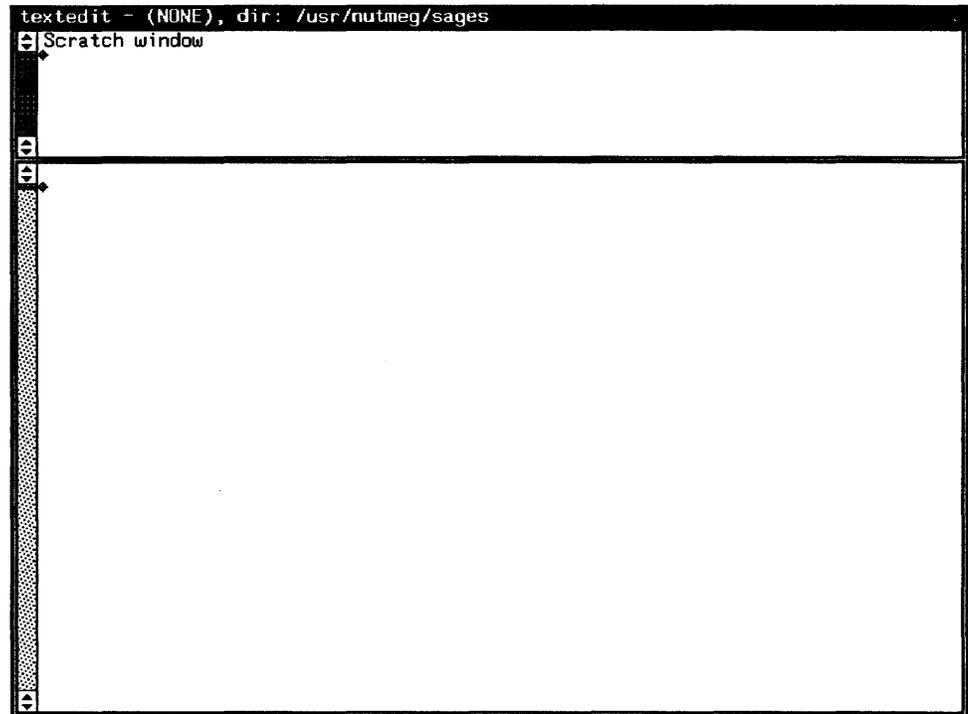
- Hold down the **Control** key.
- While holding down the MIDDLE mouse button, grab the frame border in the middle of the side you want to stretch or squeeze, and move the pointer toward or away from the opposite side.
- Release the mouse button and the **Control** key when the bounding box location is satisfactory.

A special use for the **Resize⇒Constrained** accelerator is to enlarge the Text Editor’s scratch window. This can only be done using an accelerator — it doesn’t work from the Frame menu because that resizes frames, not windows.

Resizing lets you keep more temporary information visible than you can fit in the single line of a standard scratch window. You could use the Defaults Editor to change the `Text/Scratch_window` default, but that change wouldn’t be immediate, and all subsequent Text Editor frames would reflect the change.

To resize a Text Editor scratch window:

- Move the pointer to the double line between the edit window and the scratch window. The double line becomes a solid black line and the pointer becomes a target.
- Hold down the **Control** key.
- Hold down the MIDDLE mouse button while dragging downward until the Text Editor frame looks like the following figure. Then release the **Control** key and the mouse button.

Figure 8-2 *The Scratch Window Resized*

“Resize⇒Zoom” and
“Resize⇒UnZoom”

Choose **Resize⇒Zoom** to resize a frame to the full height of the screen. When you have zoomed a frame, the menu item toggles to **UnZoom**, which remembers the former location of the frame. **Zoom** and **UnZoom** are handy when you want to list a large directory in a Command Tool, look at it briefly, and then restore the Command Tool frame to its former size.

UnZoom also restores a frame that was enlarged by **Resize⇒FullScreen**.

Here are accelerators for **Zoom** and **UnZoom**:

- To zoom a frame, move the pointer over the frame border. The pointer shape changes to a target.
- Hold down the **(Control)** key while clicking the LEFT mouse button. The frame zooms to full height.
- To restore a zoomed frame to its original size and location, follow the same procedure: Click **(Control)**-LEFT with the pointer on the frame border.

“Resize⇒FullScreen”

FullScreen enlarges a frame so that it occupies the entire screen. To restore the frame to its former size, choose **Resize⇒UnZoom** or use the **Zoom/UnZoom** accelerator.

“Front”

Think of the frames on a screen as sheets of paper on a horizontal surface, like a desk. Some of the sheets overlap. Now imagine rotating the surface ninety degrees so that it's standing up, like a screen. The overlapping sheets or frames that were on top are in front; those that were on the bottom are in back.

Front changes the overlap by exposing a frame, bringing it forward, in front of other frames. The frame from whose Frame menu you choose **Front** is the frame that is acted upon. **Front** and **Back** also work on icons.

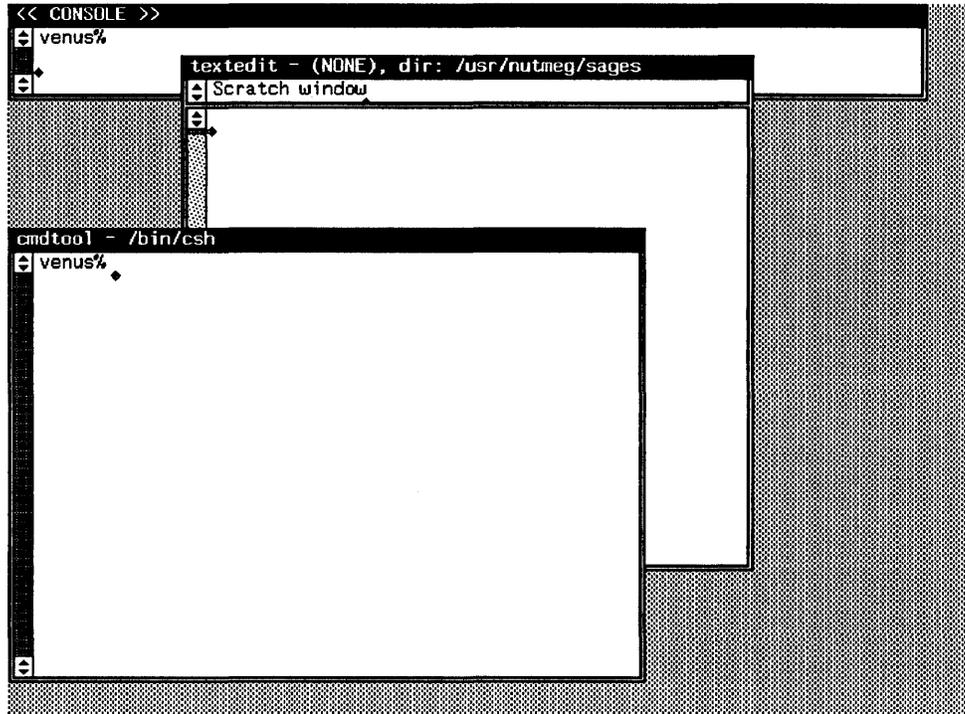
An accelerator equivalent for the **Front** menu item is:

- Move the pointer anywhere over the frame you want to bring forward.
- Press the **Front** function key.

“Back”

Back changes the overlap by hiding a frame and sending it in back of all the other frames. The frame from whose Frame menu you choose **Back** is the frame that is acted upon.

It's possible to have a frame that's in front of one frame and in back of another. In the following figure, the Text Editor frame is in front of the Console frame and in back of the Command Tool frame. If you choose **Back** from the Command Tool's Frame menu, the Text Editor frame will overlap the Command Tool frame but remain in front of the Console frame. If you choose **Back** from the Text Editor's Frame menu, on the other hand, the Text Editor frame will be overlapped by both other frames.

Figure 8-3 *Overlapping Frames***“Props”**

Choose **Props** to bring up a property sheet that lets you set options for the tool. **Props** is currently grayed — not available for use — in most SunView tools.

“Redisplay”

Redisplay redraws the frame and its contents. It's needed only when the screen image seems to have become corrupted.

You can also redraw the entire screen by choosing **Services⇒Redisplay All** from the SunView menu.

“Quit”

Quit sends away the tool that owns the frame. No icon remains. To start the tool again, choose it from the SunView menu or type its name in a shell command.

An alert message asks you to confirm that you really want to quit.

8.5. The Text Menu

You can bring up a Text menu in any text window or command window. Some windows have limits on the features you can use, and others have special features. For example, you can't use **Load File** in a scratch window, and you get an extra pull-right menu, **Cmd Modes⇒**, in a command window. The standard Text menu pull-right menus are **File⇒**, **Edit⇒**, **Display⇒**, **Find⇒**, and **Extras⇒**, plus **Cmd Modes⇒** in command windows.

8.6. The “File⇒” Pull-Right Menu

The **File⇒** pull-right menu contains commands for saving and retrieving files, setting the directory, starting a new document, plus some fast ways to end a work session.

“Save Current File”

When you're ready to save your work, bring up the Text menu and choose **File⇒Save Current File**. This saves the document in the window under the filename that appears in the frame header. If the frame header says **(NONE)**, there's no current file — type the new filename, select it, and choose **File⇒Store as New File** instead. The new filename can be in any window.

Backup Files

Each time the Text Editor does a save or store, it first copies the old version of the file to a file named `filename%`. The percent sign (%) means this file was created as a backup file. The backup file is saved in the current directory.

The next time you save or store, the Text Editor replaces `filename%` before it saves the new `filename`. Backups are automatic. You simply continue editing.

Backup files don't go away until you delete them.

Remember to Save Frequently

When you have been working in the Text Editor for a long time, the internal data structures may become fragmented so that you notice a degradation in performance. Save the document you're working on, and performance will return to normal.

Frequent saving is probably a good idea in any case — if the system crashes, the edits you made since the last Store or Save will be lost.

If you'd rather have SunView do the saving, you can use the Defaults Editor to request *checkpointing*. Change the `Checkpoint_frequency` default line in the **Text** category from its default of 0 to a positive integer *n*. This causes a checkpoint file to be written every *n* edits. The name of the checkpoint file is the name of the file you're editing with the suffix `%%` added. See Chapter 10, “The Defaults Editor”, for instructions on how to change a default.

“Store as New File”

This menu item lets you create a new file. If the filename you select is the name of an existing file, an alert window gives you a warning message.

To store a document in the window as a new file:

- Select the filename in any window. You may have to type it, in which case the Text Editor's scratch window is a handy place to do that.

- **Choose File⇒Store as New File.** The document is stored to the selected filename in the current directory (shown in the frame header).

To store a document in a different directory, select the absolute or relative pathname of the file. For example, `/home/novels/great_american` is an absolute pathname; `../comedies_and_tragedies` is a relative pathname.

“Load File”

To load a file in the Text Editor:

- Select the filename in any window. If it's not in the current directory, choose **File⇒Set Directory** first or specify the full pathname.
- **Choose File⇒Load File.**

The frame header shows the new filename.

Here's an accelerated way to load a file:

- Move the pointer into the text window where you want the file loaded.
- Type the filename by itself on the first line in the text window.
- Press the **[Esc]** key. The file is loaded into the text window.

See the topic “Set Directory” in this chapter for a way to use **Load File** to set the directory.

“Include File”

Include File lets you put a file in a text window at the insertion point. The filename in the frame header doesn't change. You can include many files in a single document and then save the document as a new file.

To include a file:

- Set the insertion point where you want the file to go.
- Select the filename in any window.
- **Choose File⇒Include File.** The file appears at the insertion point and the window scrolls to the end of the inserted file.

Here's an accelerated way to include a file in the middle of a document:

- Move the pointer into the text window where you want the file included.
- Type the filename by itself on the line where you want the file inserted.
- Press **[Shift-Esc]**.

The file is inserted into the text window, replacing its name.

“Set Directory”

The current directory for the Text Editor appears on the right side of the frame header. The default is your working directory.

To change the current directory:

- In any window, select the name of the new directory. Type it first if you need to.

- Move the pointer into the edit window and bring up the Text menu. Make sure the pointer is in the edit window, not the scratch window.
- Choose **File⇒Set Directory**.

The new directory name appears in the frame header.

Set Directory follows the same rules as the SunOS `cd` (change directory) command. For example, if you're in your home directory, `/cinammon/nutmeg/sage`, and you want to change the directory to `/cinammon/nutmeg/sage/parsley`, select the word `parsley` and choose **Set Directory**. You can change back to your home directory by selecting `..` or `~` and choosing **Set Directory** again.

There are two accelerators for **File⇒Set Directory**: `[Esc]` and **File⇒Load File**:

- To set the directory using the `[Esc]` key, type the directory name by itself on the first line in the window, then press `[Esc]`.
- Select the directory name in any window, then move the pointer into the edit window whose directory you want to change and choose **Load File**. The directory is changed and the new directory appears in the frame header.

To set the directory using **Load File**, you must first change the Defaults Editor setting for `Text/Load_file_of_directory` from `Is_error` to `Is_set_directory`, then bring up a new Text Editor frame.

“Empty Document”

Sometimes you're editing an existing file and you want to start a brand-new document. The way to do that is to save the file you're working on and then ask for an *empty document* — a text window with no text in it and no filename associated with it.

To empty the text window:

- Choose **File⇒Empty Document**. The filename is changed to `(NONE)`.

If you haven't saved your most recent edits in that window, you'll receive an alert message asking if you really mean to discard them.

Getting an empty document is just like bringing up a new Text Editor. You can create a new document from scratch. When you save it for the first time, you'll have to choose **Store as New File** because there isn't a current file yet.

“Finishing Up⇒”

The **Finishing Up⇒** pull-right menu lets you specify a combination of Save or Store together with another action.

The menu items are:

- **Close, then Save Current File**
- **Save Current File, then Quit**
- **Save, then Empty Document**
- **Close, then Store as New File**

- **Store as New File, then Quit**

The combinations that include **Store as New File** require a selection.

8.7. The “Edit⇒” Pull-Right Menu

The **Edit⇒** pull-right menu contains commands for copying and inserting text, undoing editing, and repeating commands.

“Again”

Edit⇒Again, the default, executes all of the operations that you performed in a text window since the last time you changed the insertion point with a mouse action.

The accelerators for **Edit⇒Again** are the **Again** function key or **◆-A**.

Scrolling the display has no effect on **Again**.

With **Again**, you can repeat the same series of commands at several places in a document. For example, you can choose **Again** to repeatedly insert beginning and ending brackets around text. (**Insert Brackets⇒** in the **Extras⇒** pull-right menu inserts some frequently used bracket pairs. The technique shown below lets you insert *any* brackets.) To insert brackets around a selection using **Again**:

- Select the first item — say, `First Example`.
- Choose **Cut** to delete the item and store it on the Clipboard.
- Type the beginning bracket, say `/*` (the opening comment delimiter in C programs).
- Choose **Paste** to retrieve the deleted text from the Clipboard.
- Type the ending bracket, say `*/` (the closing comment delimiter in C programs). The result is `/*First Example*/`.
- Select the second item.
- Choose **Again** to bracket the second item.
- Select the third item and choose **Again**.

And so on. . . Note that the text you bracket doesn't have to be the same in each case. For example, you can use **Again** to bracket `Second Example`, `Third Example`, and so on.

Again Ignores Undo

One convenient use of **Again** is to correct the error of inserting text at the wrong place. Just choose **Undo⇒**, move the insertion point to the correct position, then choose **Again**.

NOTE: Edits performed in one text window have no effect on the operation of **Again** in another text window.

“Again” and “Find”

You can use **Again** in conjunction with **Find** to search through a document, finding and replacing text. **Find** is one of the operations that **Again** repeats because, although **Find** changes the insertion point, it does so without a mouse action.

Again with **Find** is useful when you want to search for special characters like Tab and Return that the **Find and Replace** frame doesn't handle.

For example, you can indent a series of lines by searching for the Return character in the first line, replacing it by pressing **(Return)** followed by **(Tab)**, then choosing **Again** for each additional line to be indented.

When **Find** doesn't find matching text, the rest of the operations that **Again** is repeating are not executed.

Here's an example combining **(Again)** with **(Find)**, keyboard control of the caret, and erase accelerators (described in Chapter 9, “All about Accelerators”). You'll be able to adapt the techniques shown here to automate previously tedious search-and-replace operations by setting up a model and repeating it almost effortlessly with **Again**.

The example edits a document on women's suffrage containing the phrases many women should vote, some women should vote, and most women should vote, changing all of these phrases to all women should vote:

- Select the word women in the first phrase and press **(Copy)** to put it on the Clipboard.

```
textedit - suffragettes, dir: /home/nutmeg/sages
In solidarity with Elizabeth Cady Stanton, we declare that many
women should vote.
Surely it is only just that some women should vote.
In fact we believe it is for the good of the nation that most
women should vote.
```

- Press the **(Find)** function key to select the next occurrence of women in the document.

```
textedit - suffragettes, dir: /home/nutmeg/sages
In solidarity with Elizabeth Cady Stanton, we declare that many
women should vote.
Surely it is only just that some women should vote.
In fact we believe it is for the good of the nation that most
women should vote.
```

- **Find** puts the caret at the end of the word, so press **Control-comma** to move the caret in front of the w.

```
textedit - suffragettes, dir: /home/nutmeg/sages
In solidarity with Elizabeth Cady Stanton, we declare that many
women should vote.
Surely it is only just that some women should vote.
In fact we believe it is for the good of the nation that most
women should vote.
```

- Press **Control-W** to erase the previous word — some, many, or most. The space is erased also.

```
textedit - suffragettes (edited), dir: /home/nutmeg/sages
In solidarity with Elizabeth Cady Stanton, we declare that many
women should vote.
Surely it is only just that women should vote.
In fact we believe it is for the good of the nation that most
women should vote.
```

- Type all followed by a space to replace the word you erased.

```
textedit - suffragettes (edited), dir: /home/nutmeg/sages
In solidarity with Elizabeth Cady Stanton, we declare that many
women should vote.
Surely it is only just that all women should vote.
In fact we believe it is for the good of the nation that most
women should vote.
```

- Press **Shift-Control-comma** to move past women in the phrase you just changed, so that **Find** won't find the same word again.

```
textedit - suffragettes (edited), dir: /home/nutmeg/sages
In solidarity with Elizabeth Cady Stanton, we declare that many
women should vote.
Surely it is only just that all women should vote.
In fact we believe it is for the good of the nation that most
women should vote.
```

- Now press the **⌘-A** function key to repeat the entire series of operations. Keep pressing **⌘-A** until the end of the document is reached.

```

textedit - suffragettes (edited), dir: /home/nutmeg/sages
In solidarity with Elizabeth Cady Stanton, we declare that all
women should vote.
Surely it is only just that all women should vote.
In fact we believe it is for the good of the nation that all
women should vote.

```

“Undo⇒”

The **Undo⇒** pull-right menu lets you undo the last edit you made, or all edits since the last **Save** or **Store as New File**.

“Undo Last Edit”

Undo Last Edit, the default, undoes the effect of the most recent editing operation: **Cut**, **Paste**, or typing.

The accelerators for **Undo Last Edit** are the **⌘-U** function key and **⌘-U**.

The range of **Undo Last Edit** goes back to the last time you clicked **LEFT** to change the insertion point. If the most recent edit was the only one since the last **Save** or **Store as New File**, the *(edited)* signal is removed from the frame header.

You can choose **Undo Last Edit** repeatedly to undo a series of edits.

“Undo All Edits”

The range of **Undo All Edits** goes back to the last time you saved or stored the document. You can think of **Undo All Edits** as the equivalent of **Empty Document** plus **Load File** to reload the last saved version.

Undo All Edits can rescue you, for example, when you push the **Replace All** button in the **Find and Replace** frame, then change your mind. Hint: You need to have chosen **Save** or **Store as New File** just before the replacement, which is always a good idea.

“Copy”, “Paste”, and “Cut”

Copy, **Paste**, and **Cut** all work with the Clipboard, a buffer shared by all frames.

- **Copy** places the selection on the Clipboard. Its accelerators are the **⌘-C** function key and **⌘-C**.
- **Paste** places the contents of the Clipboard at the insertion point. Its accelerators are the **⌘-V** function key and **⌘-V**.
- **Cut** deletes the selection and places it on the Clipboard. Its accelerators are the **⌘-X** function key and **⌘-X**. **Paste** undoes **Cut** as long as you don’t move the caret.

“Show Clipboard⇒”

Show Clipboard⇒ is a pull-right menu whose menu item consists of the current contents of the Clipboard.

If the selection on the Clipboard is long, only the first several characters are shown.

For convenience, **Show Clipboard⇒** also appears in the **Find⇒** pull-right menu.

“Copy then Paste”

Copy then Paste gives you two operations in menu item. When the active caret is not with the selection — for example, when you make a selection and then move the keyboard focus to another window — **Copy then Paste** copies the selection to the caret.

In a command window, you can also use this handy combination to copy a previous command to the command line in order to edit it or execute it again. That's because the caret stays on the command line when you select text in the transcript (unless you've selected **Enable Editing**).

An accelerator for **Copy then Paste** is **⌘-P**. It works like **Stuff** in the Tty menu. Some people think **⌘-P** is the most useful accelerator there is.

8.8. The “Display⇒” Pull-Right Menu

The **Display⇒** pull-right menu provides commands for changing the text display. You can scroll to the text that contains the caret; move the caret to the top or bottom of the document and scroll there; split the window into scrollable sections; and change the way text wraps at the right window edge.

“Select Line at Number”

Select Line at Number, the default, scrolls the document so that the line whose number you specify shows in the window. This menu item requires a selection containing the line number you want.

A quick way to correct errors in a compilation is to use two windows. One contains the source program, the other contains the error listing. Starting from the *last* error shown, so that you won't change earlier line numbers when you make corrections:

- In the error-listing window, select the number of a line containing an error.
- In the source-program window, choose **Select Line at Number**. The line scrolls into view.
- Correct the error in the source listing.
- In the error-listing window, select the next number, and so on.

“What Line Number?”

What Line Number? is a request for the line number of the line that contains the selection or the beginning of a multi-line selection. An alert window appears, telling you the line number. Push the **Continue** button after you've read the alert message.

Select Line at Number and **What Line Number?** refer to the physical line number — where a line is the amount of text between Line Feed or Return

characters. Therefore the line number changes only if you add or delete a Line Feed or Return, not if you merely change the line wrap or the window width.

“Split View”

You can split a text window or command window into an (almost) infinite number of sections and scroll each one independently. Each section is known as a *view*. Before you split it, the view consists of the entire window.

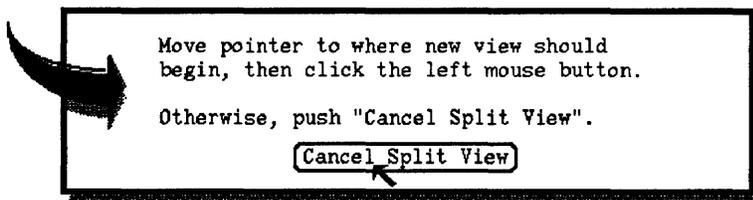
Many Views, One Caret

A window contains only one caret, representing the insertion point, no matter how many views you have.

To split a view:

- Move the pointer into the view.
- Bring up the Text menu and choose **Display⇒Split View**. An alert window appears:

Figure 8-4 *Split View Alert Window*

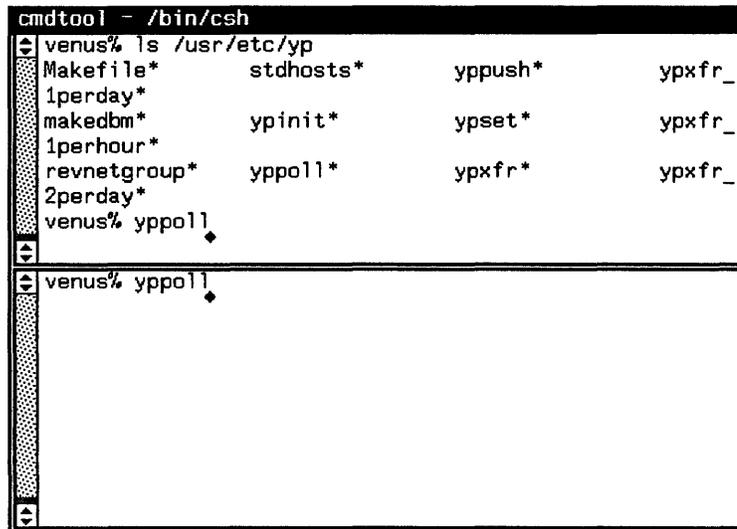


- Click **LEFT** anywhere in the view. A horizontal line appears, splitting the view at the point where you clicked.

Why Split a View?

Split views come in really handy. Here are examples from two different tools. As you work with text windows, you'll probably find additional uses.

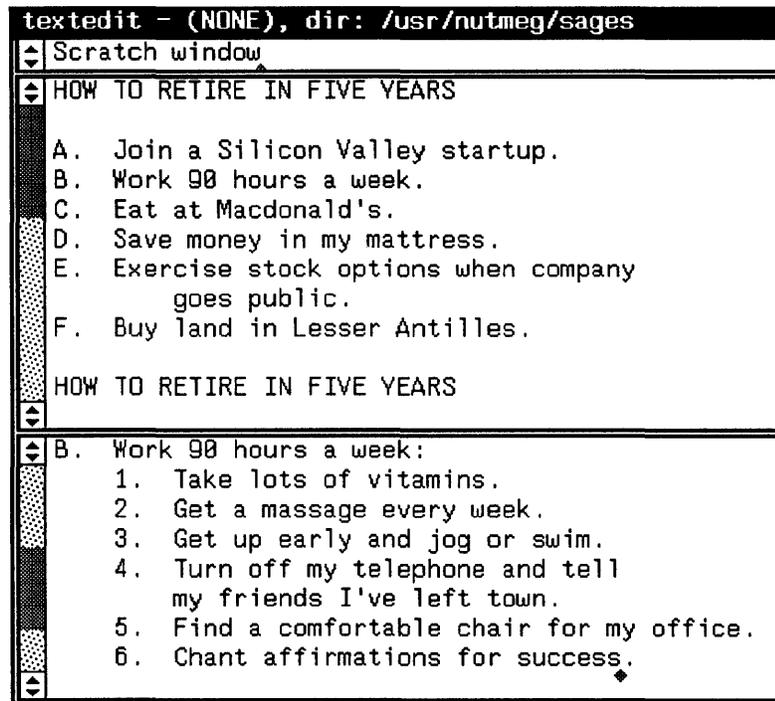
- In a Command Tool, you can list the contents of a directory, then split the view so that you can continue to see the directory listing in the top view as you type commands in the bottom view.

Figure 8-5 *Split View in Command Tool*

```
cmdtool - /bin/csh
venus% ls /usr/etc/yp
Makefile*      stdhosts*      yppush*        ypxfr_
1perday*
makedbm*       ypinit*        ypset*         ypxfr_
1perhour*
revnetgroup*   yppoll*        ypxfr*         ypxfr_
2perday*
venus% yppoll
venus% yppoll
```

- In an edit window in the Text Editor, you can load an outline of a document, duplicate it in the window, then split the view so you can continue to see the outline in the top view while writing the sections in the bottom view.

Figure 8-6 *Split View in Text Editor*



“Destroy View”

To unsplit a view:

- Move the pointer into either half of the split view.
- Bring up the Text menu and choose **Display⇒Destroy View**.

“Show Caret at Top”

Show Caret at Top moves the line containing the insertion point to the top of the window. This is a good way to retrieve a “lost caret” when you’ve been scrolling through history in a Command Tool.

“Move Caret to Start”

Move Caret to Start moves the caret in front of the first character in the document and displays it at the top of the window. Sometimes it’s helpful to do this before beginning a find and replace operation.

NOTE: **Move Caret to Start** and **Move Caret to End** do *not* affect the current selection, even though it may no longer be visible in the window. You can inadvertently delete or change a pending-delete selection by assuming that you’re working at the caret.

Accelerators for **Move Caret to Start** are **[Home]** (R7) and **[Shift-Control-Return]**.

“Move Caret to End”

Move Caret to End moves the caret to the end of the last line in the document and displays it at the bottom of the window. This is a good way to get to the end of a text window when you want to type or include more text.

Accelerators for **Move Caret to End** are **[End]** (R13) and **[Control-Return]**.

“Change Line Wrap⇒”

The **Change Line Wrap** pull-right menu lets you change the way physical lines are displayed in a window.

Wrapping means that the portion of a line that extends past the right edge of the window flows to the next line in the window, starting at the left.

“Wrap at Character”

Wrap at Character, the default, causes characters that do not fit on a line in the window to wrap around to the following window line. A single physical line may be wrapped onto several window lines, depending on the width of the window.

“Wrap at Word”

Wrap at Word operates like **Wrap at Character** with words rather than characters as the basic unit. In most text windows, **Wrap at Word** is a lot easier on the eyes — but remember that the actual length of physical lines may be obscured, so you may get strange results when you print the text. In Command Tool, you may prefer **Wrap at Character** when reading long command lines.

“Clip Lines”

When you don't want wrapping of long lines, choose **Change Line Wrap⇒Clip Lines**. This menu item causes characters that do not fit on the line to “disappear” on the right side of the window — they're still there, but you can't see them.

To wrap the lines in the document again, choose **Wrap at Character** or **Wrap at Word**.

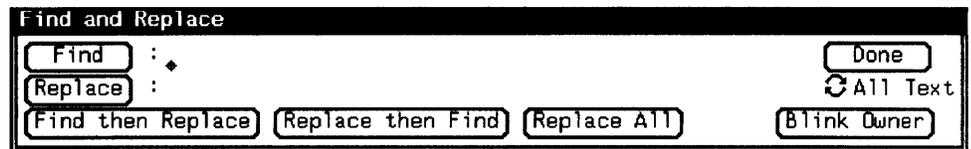
You can permanently divide long lines into shorter ones by piping the text through the `fmt` command. To do this, select the text to be formatted, make it pending-delete, and choose **Format** from the **Extras⇒** pull-right menu.

8.9. The “Find⇒” Pull-Right Menu

The **Find⇒** pull-right menu offers many ways to find and replace text. This menu includes two menu items that bring up pop-up frames for specialized Find operations.

The Find and Replace Frame

Choose **Find and Replace**, the default, to bring up the **Find and Replace** pop-up frame, shown in the next figure. An accelerator to bring up the frame is **[Control-Find]**. In the **Find:** and **Replace:** panel text items, type or paste the text you want to find and the text to replace it, then push panel buttons to do the work.

Figure 8-7 *The Find and Replace Frame*

When searched-for text is found, the text is selected and the window is scrolled to display it.

- The **Find:** text item contains the text you want to search for. As with other panel text items, you can select or delete the entire text item, or backspace from the end of it, but you can't do more complex editing.

If there's a current selection when you bring up the **Find and Replace** frame, the selection is inserted in the **Find:** text item automatically.

When you push the **Find** button, a search begins at the caret and the next occurrence of the **Find:** text item is selected.

If there's no match, the owner frame blinks and beeps.

The **Find** button searches forward by default, but if instead of clicking **LEFT** you hold down **RIGHT**, you'll see a menu that lets you set the direction to **Backward** instead. You can also search backward by pressing **[Shift]-Find**. The **[Shift]** key often reverses the direction of an action in SunView.

- The **Replace:** text item contains the text you want to use as a replacement for the found selection. Push the **Replace** button to replace the selection.
- Push the **Find then Replace** button to find and immediately replace the next occurrence. If you goofed, you can choose **Edit⇒Undo** to correct the mistake.
- Push the **Replace then Find** button to replace the found selection and immediately find the next occurrence.
- Push the **Replace All** button to change every occurrence in the document. It's a good idea to save or store just before you do this, in case the results are different from what you expected.

All Text Cycle Item

The cycle item is set to **All Text**. This means that no matter where the caret is when you begin searching, the search doesn't stop at the end of the document but wraps around to the top and continues back to the caret. If you want the search to stop at the end of the document, click **LEFT** on the cycle item to set it to **To End**.

Pushing the **Blink Owner** button blinks the owner's frame to show you where it is, in case you have more than one open text window or command window.

The **Done** button closes the **Find and Replace** frame. You can leave it open, though, and use it to search for different text. When you close the owner frame, the pop-up frame closes too.

- “Find Selection⇒”** Choose **Find Selection⇒** to search from the caret and select the next occurrence of the current selection. The default direction is forward, but you can choose **Find Selection⇒Backward**.
- “Find Clipboard⇒”** Choose **Find Clipboard⇒** to search from the caret and select the next occurrence of the Clipboard contents. The default direction is forward, but you can choose **Find Clipboard⇒Backward**.
- Accelerator for Find⇒** The **Find** function key acts as an accelerator. Here's how it works.
- If there's a selection, **Find** is the equivalent of **Find Selection⇒Forward** and **Shift-Find** is the equivalent of **Find Selection⇒Backward**.
 - If there's no selection, **Find** is the equivalent of **Find Clipboard⇒Forward** and **Shift-Find** is the equivalent of **Find Clipboard⇒Backward**.
 - If you make a secondary selection by holding down the **Find** key while selecting text, the secondary selection is searched for when you release the function key. The search is in a forward direction, but you can reverse it by holding down **Shift-Find** while making the secondary selection. This is a handy way to search without changing the Clipboard or the current primary selection.
- “Show Clipboard⇒”** Highlight **Show Clipboard⇒** and pull right to see the current contents of the Clipboard.
- For convenience, this menu item appears in both the **File⇒** and **Edit⇒** pull-right menus. A more complete description is found under the topic “The **Edit⇒** Pull-Right Menu” earlier in this chapter.
- The Find Marked Text Frame** Choose **Find Marked Text** to search for text bounded by certain delimiters. **Find Marked Text** brings up the **Find Marked Text** pop-up frame, shown in the figure below.

Figure 8-8 *The Find Marked Text Frame*

There's a *choice button* to the left of each delimiter pair in the **Find Marked Text** frame. The pairs are (), " ", ' ', " ", { }, [], |> <|, and /* */. Click LEFT on the choice button to indicate which pair to search for. The choice button highlights. Then push **Forward** or **Backward** to search in the requested direction. The next occurrence of text within the requested delimiters is highlighted.

The **Expand** button finds the next outer pair of the specified delimiters, searching in both directions from the caret.

The **Blink Owner** and **Done** buttons work just as they do in the **Find and Replace** frame.

“Match Delimiter”

Select a delimiter, then choose **Match Delimiter** to find the matching end of the delimiter pair. The delimiters and the text between them are selected. The delimiters recognized by **Match Delimiter** are the same as those displayed in the **Find Marked Text** frame.

If the beginning and ending delimiters differ, like open bracket and close bracket ([and]), **Match Delimiter** searches forward if the selection is a beginning delimiter and backward if the selection is an ending delimiter. If there's no difference, **Match Delimiter** searches forward.

“Replace |>field<|⇒”

Choose **Replace |>field<|⇒** to find a field and select it as pending-delete. The search begins at the caret. A *field* is any text between |> and <| delimiters.

Expand, the default, searches in both directions, expecting the caret to be within a delimiter pair.

Next searches forward from the caret. An accelerator for **Replace |>field<|⇒Next** is **Control-Tab**.

Previous searches backward from the caret. An accelerator for **Replace |>field<|⇒Previous** is **Shift-Control-Tab**.

8.10. The “Extras⇒” Pull-Right Menu

The **Extras⇒** pull-right menu contains some useful formatting commands. The menu gives you a way to use text filters without having to map them to keys.

When you make a selection and choose an **Extras⇒** menu item, the selection becomes the standard input (`stdin`), and the resulting standard output (`stdout`) is inserted at the caret. If the selection was pending-delete, the standard output replaces the selection. For more information on the standard input and the standard output, see *SunOS User's Guide: Doing More*.

“Format”

Format uses the `fmt` command to reformat long lines of text.

Make a pending-delete selection, then choose **Format** to format the lines in the selection with a line length that can be printed and viewed more easily.

NOTE: To make a selection pending-delete, hold down the **Control** key while making the selection.

The `fmt` command divides text into physical lines of not more than 72 characters; it does not split words between lines. See the `fmt` command in the *SunOS Reference Manual* for more information.

- “Pretty-print C”** **Pretty-print C** uses the `indent` command to format C program listings. It formats the selection according to standard specifications. If you have a `.indent.pro` file, the selection is formatted according to the specifications in the file. For more information on formatting source programs written in C, refer to the `indent` command in the *SunOS Reference Manual*.
- “Capitalize⇒”** The **Capitalize⇒** pull-right menu gives you three choices for changing the capitalization of a selection.
- NOTE: Make the selection pending-delete before choosing from the **Capitalize⇒** pull-right menu in order to replace the selection with the newly capitalized version.
- “abcd->ABCD”** Choose **Capitalize⇒abcd->ABCD** to change all the letters in the pending-delete selection to upper case.
- “ABCD->abcd”** Choose **Capitalize⇒ABCD->abcd** to change all the letters in the pending-delete selection to lower case.
- “abcd->Abcd”** Choose **Capitalize⇒abcd->Abcd** to change the first letter of every word in the pending-delete selection to upper case. Upper-case letters within a word are not changed.
- “Shift Lines⇒”** The **Shift Lines⇒** pull-right menu lets you insert or remove a Tab character at the beginning of each physical line in a pending-delete selection.
- Shift Lines⇒Right**, the default, inserts a Tab character at the beginning of the line.
- Shift Lines⇒Left** removes a Tab character from the beginning of the line.
- “Insert Brackets⇒”** The **Insert Brackets⇒** pull-right menu lets you insert brackets around a pending-delete selection. The bracket choices are (), [], { }, and " ".
- “Remove Brackets⇒”** The **Remove Brackets⇒** pull-right menu lets you remove brackets from a pending-delete selection. The bracket choices are the same as for **Insert Brackets⇒**.
- The brackets to be removed must begin and end the selection; otherwise no action is taken.
- 8.11. Customizing the “Extras⇒” Menu** You can customize the **Extras⇒** pull-right menu to include your own favorite commands by making a new **Extras⇒** menu file and placing its name in the `Extras_menu_filename` text item in the **Text** category of the Defaults Editor. It's customary to place the file in your home directory as `~/text_extras_menu`.

The standard **Extras**⇒ menu file is `/usr/lib/.text_extras_menu` and is shown in the next figure. It will give you some ideas on how to implement your own extras. Once you've created a custom file and told the Defaults Editor about it, you can make changes to the file and see them immediately implemented in the **Extras**⇒ pull-right menu.

Figure 8-9 **Extras**⇒ Menu File in `/usr/lib/.text_extras_menu`

```

textedit - /usr/lib/.text_extras_menu, dir: /home/nutmeg/sages
Scratch window
#      @(#)text_extras_menu 1.5 87/11/24 SMI
#
#      Copyright (c) 1987 by Sun Microsystems, Inc.
#
#      Text "Extras" menu
#
"Format"          fmt

"Capitalize"      MENU
  "abcd -> ABCD"  capitalize -u
  "ABCD -> abcd"  capitalize -l
  "abcd -> Abcd"  capitalize -c
"Capitalize"      END

"Shift Lines"    MENU
  "Right" shift_lines -t 1
  "Left"  shift_lines -t -1
"Shift Lines"    END

"Pretty-print C" indent -st

"Insert Brackets" MENU
  " ( )" insert_brackets ( )
  " [ ]" insert_brackets \[ \]
  " { }" insert_brackets { }
  "` `'" insert_brackets ` " \ "`
"Insert Brackets" END

"Remove Brackets" MENU
  " ( )" remove_brackets ( )
  " [ ]" remove_brackets \[ \]
  " { }" remove_brackets { }
  "` `'" remove_brackets ` " \ "`
"Remove Brackets" END

```

The name of each **Extras**⇒ menu item is surrounded by double quotation marks (""). Each pull-right menu begins with a MENU statement and ends with an END statement.

To include a backslash (\) character in the `.text_extras_menu` file — as in the `troff` inline font specifications `\fB` and `\fP` — you must use *four* backslash characters; for example:

```
insert_brackets \\\fB \\\fP
```

8.12. The “Cmd Modes⇒” Pull-Right Menu

The Text menu in command windows contains a special pull-right menu, **Cmd Modes⇒**, in addition to those described above.

“Enable Editing”

The text in a command window is usually read-only except for the command line. To allow *all* text in a command window to be edited, choose **Enable Editing**.

When you enable editing, you can use the command window as a scratch pad and cross off items you're done with by deleting them. For instance, use the `fgrep` command to find and display all occurrences of a certain text string in a set of files. Then delete each display line in the command window as you update the files in a Text Editor window.

The menu item toggles to **Disable Editing** (the normal condition in a command window).

“Disable Scrolling”

Scrolling is the default in a command window. Most tools that need to disable scrolling, such as `vi`, do it themselves and restore scrolling automatically when they're done. *Therefore, you normally will never need to disable scrolling.*

However, if you have software that requires you to disable scrolling and you want to run it in a command window, choose **Disable Scrolling** before you type the command that starts the software. If you disable scrolling by choosing this menu item, you must reenable it the same way.

Disable Scrolling toggles to **Enable Scrolling**, so you can reset the scrolling facility when you're done.

You may decide instead to run non-scrolling software in a tty window instead of a command window — for instance, in Shell Tool rather than Command Tool.

8.13. The Tty Menu

Tty windows such as Shell Tool have their own menu, the Tty menu. Its menu items are **Stuff**, **Enable Page Mode**, and **Copy then Paste**.

When you disable scrolling in a command window, you get a Tty menu instead of a Text menu. The command window's Tty menu is just like the one in a tty window except that it has an additional menu item, **Enable Scrolling**, to take the window back to normal command window operations.

“Stuff”

Stuff is provided for compatibility with previous releases. Instead use **Copy then Paste** or its accelerator -P to insert the selection at the block cursor. Or use secondary Paste, a quick-editing technique described in Chapter 6, “All about Selections”.

“Enable Page Mode”

Enable Page Mode shows you only a *page* (windowful) of text at a time, like the `more` command.



When you choose **Enable Page Mode**, the pointer changes to a stop sign when the window fills up. Instead of continuing to flow the lines through the window, the display stops so that you can look at it as long as you like. You can also change the cursor to the stop sign by typing `Ctrl-S` and change it back by typing `Ctrl-Q`.

Enable Page Mode toggles to **Disable Page Mode**, so you can switch back and forth at your convenience.

When you've finished looking at the current *page* (or windowful), press any key to see the next page. In addition, **Enable Page Mode** toggles to **Continue** whenever there's more information to be displayed. You can choose **Continue** from the menu to display the next page of information.

“Copy then Paste”

Copy then Paste inserts the selection at the block cursor. It works just like the menu item of the same name in the Text menu's **Edit⇒** pull-right menu. You'll find a description of **Edit⇒Copy then Paste** in the topic **Edit⇒Copy then Paste** earlier in this chapter.

All About Accelerators

This chapter describes:

- *Function key accelerators* — menu actions you perform by pressing a function key on the left side of the keyboard.
- *Mouse button accelerators* — Frame menu actions you perform by clicking or holding down a mouse button.
- *Keyboard accelerators* — keyboard combinations you press to erase text, move the caret, and perform Text menu actions without using the mouse.
- *Alert button accelerators* — ways to push the button in an alert window from the keyboard, without using the mouse.

Accelerators are alternative ways of doing things in SunView. Their name stems from the fact that they're often easier and faster. For example,

- Pressing the **Undo** function key
is the accelerator equivalent of
 - Bringing up the Text menu.
 - Pulling right on the **Edit⇒** menu.
 - Pulling right on the **Undo⇒** menu.
 - Highlighting **Undo Last Edit** and releasing the mouse button.

See the **Again** menu item in Chapter 8, “All about Menus” for an example of how to repeat a complex series of actions using function key accelerators, erase accelerators, and caret accelerators.

9.1. Function Key Accelerators

The function key accelerators, which are described in detail in Chapter 2, let you perform menu operations very quickly just by pressing a function key. The menu operations themselves are described in Chapter 8, “All about Menus”.

Function keys aren't the only way to perform menu operations in SunView. See Table 9-5, “Menu Accelerators”, for additional possibilities.

The Control Key

The **(Control)** key is used in conjunction with function keys or keyboard accelerators. **(Control-Find)**, an accelerator for **Find and Replace**, means *hold down the (Control) key while pressing the (Find) function key*.

The Shift Key

The **(Shift)** key is used in conjunction with function keys or keyboard accelerators to *toggle* (switch or reverse) their operation. **(Shift-Control-Return)**, an accelerator for **Display⇒Move Caret to Start**, means *hold down the (Shift) key and the (Control) key while pressing the (Return) key*.

The **(Shift)** key is often used to reverse the meaning of an accelerator. For example, **(Shift-Find)** reverses the direction of the **(Find)** function key from **Forward** to **Backward**, and **(Shift-Front)** reverses the action of the **(Front)** function key from **Front** to **Back**.

The **(Shift)** key toggles many other keyboard accelerators as well. For instance, **(Shift-Control-B)** reverses the direction of the **(Control-B)** caret action (**Go_Char_Backward**) to forward.

Table 9-1 *Function Key Accelerators*

Type4 Keyboard	Menu Action	Menu	Other Keyboards
(Stop)	Stop	—	(L1)
(Props)	Props	Frame	(L3)
(Front)	Front, Back	Frame	(L5)
(Open)	Open, Close	Frame	(L7)
(Find)	Find⇒Find Selection⇒Forward, Find⇒Find Clipboard⇒Forward	Text	(L9)
(Again)	Edit⇒Again	Text	(L2)
(Undo)	Edit⇒Undo⇒Undo Last Edit	Text	(L4)
(Copy)	Edit⇒Copy	Text	(L6)
(Paste)	Edit⇒Paste	Text	(L8)
(Cut)	Edit⇒Cut	Text	(L10)
(Shift-Front)	Back	Frame	(Shift-L5)
(Shift-Find)	Find⇒Find Selection⇒Backward, Find⇒Find Clipboard⇒Backward	Text	(Shift-L9)
(Control-Find)	Find⇒Find and Replace	Text	(Control-L9)

- **(Open)** toggles to **Close** if the pointer is over a frame rather than an icon.
- **(Front)** brings a partly exposed frame to the front. It toggles to **Back** if the frame is completely exposed.
- **(Shift-Front)** sends a completely or partly exposed frame to the back. It does *not* toggle to **Front**.
- If there is a selection, **(Find)** is an accelerator for **Find⇒Find Selection⇒Forward**. Without a selection, it's an accelerator for **Find⇒Find Clipboard⇒Forward**. **(Shift-Find)** works the same way, but in the **Backward** direction.

- **⌘-F** and **⇧-⌘-F**, shown in Table 9-5, “Menu Accelerators”, are keyboard accelerators that operate just like the function key accelerators **Find** and **⇧-Find**.
- **⌘-Find** brings up the **Find and Replace** frame.

9.2. Mouse Button Accelerators

Mouse button accelerators let you perform Frame menu actions using the mouse. These accelerators are described in the following table.

Table 9-2 *Mouse Button Accelerators*

<i>Frame Menu Equivalent</i>	<i>Mouse Action</i>
<i>Open a frame:</i>	Point to the icon and click LEFT.
<i>Move a frame unconstrained:</i>	Point to a side of the frame border near the corner, hold down MIDDLE and drag the frame, release when the bounding box is the desired size.
<i>Move a frame constrained:</i>	Point to a side of the frame border near the middle, hold down MIDDLE and drag the frame, release at the desired position.
<i>Move an icon (always unconstrained):</i>	Point anywhere in the icon, hold down MIDDLE and drag the icon to the new location.
<i>Resize a frame unconstrained:</i>	Point to a side of the frame border near the corner, hold down ⌘ -MIDDLE and resize the frame, release at the desired position.
<i>Resize a frame constrained:</i>	Point to a side of the frame border near the middle, press ⌘ -MIDDLE and resize the frame, release at the desired position.
<i>Zoom a frame to full height:</i>	Click ⌘ -LEFT on the frame border.
<i>UnZoom a zoomed frame:</i>	Click ⌘ -LEFT on the frame border of a frame that has been zoomed.
<i>Bring frame to front:</i>	Click LEFT on the frame border.

9.3. Keyboard Accelerators

There are also accelerators in SunView that are *actions*, or commands, that you type from the keyboard. In Tables 9-3 through 9-6 you'll find the following action types:

- *Erase actions* remove text without putting it on the Clipboard.
- *Caret actions* give you keyboard control of the caret.
- *Menu actions* provide menu equivalents. (Not all menu items have accelerators.)
- *Other actions*.

A command may have more than one keyboard equivalent. The best way to see how these commands work is to try them.

The Meta (◆) Keys

The *Meta* keys are the diamond (◆) keys on either side of the space bar.⁹ Meta-key commands are usually menu item equivalents. You can use whichever Meta key is more convenient. Meta keys are used in conjunction with one or more other keys to provide keyboard accelerators; for example, **◆-C**, an accelerator for **Edit⇒Copy**, means *hold down the* **◆** *key while typing the letter* **C**.

9.4. Erase Accelerators

Erase accelerators remove text from a document. The text is erased but is *not* placed on the Clipboard.

Table 9-3 *Erase Accelerators*

<i>Erase Action</i>	<i>Keyboard Equivalent</i>
Erase_Char_Backward	Delete
Erase_Char_Forward	Shift-Delete
Erase_Word_Backward	Control-W
Erase_Word_Forward	Shift-Control-W
Erase_Line_Backward	Control-U
Erase_Line_End	Shift-Control-U

9.5. Caret Accelerators

Caret accelerators let you change the position of the caret in text without having to move the mouse. You can keep your hands on the keyboard while skipping over text to reposition the caret.

⁹ On some Sun keyboards, the Meta keys are labeled *Left* and *Right*.

Table 9-4 *Caret Accelerators*

<i>Caret Action</i>	<i>Keyboard Equivalent</i>
Go_Char_Backward	R10 (left arrow) Control-B
Go_Char_Forward	Shift-Control-F R12 (right arrow) Control-F Shift-Control-B
Go_Word_Backward	Control-comma (,) Shift-Control-slash (/) Shift-Control-period (.)
Go_Word_End	Control-period (.)
Go_Word_Forward	Control-slash (/) Shift-Control-comma (,)
Go_Line_Backward	Control-A Shift-Control-E
Go_Line_End	Control-E Shift-Control-A
Go_Line_Forward	R11
Go_Column_Backward	Control-semicolon (;) R8 (up arrow) Control-P Shift-Control-N
Go_Column_Forward	R14 (down arrow) Control-N Shift-Control-P
Go_Document_Start	R7 (Home) Shift-Control-Return
Go_Document_End	R13 (End) Control-Return

NOTE *Caret actions do not affect the selection if there is one.*

The *indented* caret action commands use the **Shift** key as a toggle. For example, **Shift-Control-F** (Go_Char_Backward) reverses the action of **Control-F** (Go_Char_Forward). By using the **Shift** key, you can reduce the number of commands you need to remember.

The following caret commands are described relative to the current position of the caret. If the caret is in the middle of a word, it moves to the beginning or end of that word.

- Go_Word_Backward moves the caret to the left until it's at the beginning of a word.
- Go_Word_End moves the caret to the right until it's at the end of a word.
- Go_Word_Forward moves the caret to the right until it's at the beginning of a word.

- `Go_Line_End` moves the caret to the end of the current line. If the caret is already at the end of the current line, it moves to the end of the next line down.
- `Go_Line_Forward` moves the caret to the beginning of the next line down.
- `Go_Column_Backward` and `Go_Column_Forward` move the caret up or down in the same column as long as the next line is at least as wide as the current line. If the next line is narrower, the caret moves to the end of the line; it returns to the original column when the next line is wide enough.

The Home and End Keys

`Go_Document_Start` is the `[Home]` key (R7) on the Type4 keyboard, and `Go_Document_End` is the `[End]` key (R13).

- `[Home]` is a keyboard accelerator for the menu item `Display⇒Move Caret to Start`.
- `[End]` is a keyboard accelerator for the menu item `Display⇒Move Caret to End`.

`[Shift-Control-Return]` is another accelerator for `Display⇒Move Caret to Start`, and `[Control-Return]` is another accelerator for `Display⇒Move Caret to End`.

Arrow Keys

The arrow keys perform the following caret actions:

<i>Caret Action</i>	<i>Arrow Key Equivalent</i>
<code>Go_Char_Forward</code>	<code>[R12]</code> (right arrow)
<code>Go_Char_Backward</code>	<code>[R10]</code> (left arrow)
<code>Go_Column_Forward</code>	<code>[R14]</code> (down arrow)
<code>Go_Column_Backward</code>	<code>[R8]</code> (up arrow)

Caret Action Mnemonics

Caret accelerators that use `[Control]` key combinations have names that are easy to learn.

F, B, N, and P stand for *forward*, *backward*, *next line*, and *previous line*. For emacs users, these keys are the equivalent of arrow keys.¹⁰ When pressed together with the `[Control]` key, they do the same thing in SunView.

A and E are the emacs commands for beginning-of-line (think of *alpha*) and *end-of-line*. When pressed together with the `[Control]` key, they do the same thing in SunView.

¹⁰ emacs is an editor that can be used in tty and command windows.

9.6. Menu Accelerators

The following table contains menu accelerators for the Text menu. Additional menu accelerators for the Text and Frame menus are shown in Table 9-1, “Function Key Accelerators”.

Table 9-5 *Menu Accelerators*

<i>Menu Action</i>	<i>Keyboard Equivalent</i>
File⇒Store as New File	⌘-S
File⇒Load File	⌘-L
	Esc
File⇒Include File	⌘-I
	Shift-Esc
File⇒Set Directory	Esc
File⇒Empty Document	⌘-E
Edit⇒Again	⌘-A
Edit⇒Undo	⌘-U
Edit⇒Copy	⌘-C
Edit⇒Paste	⌘-V
Edit⇒Cut	⌘-X
Edit⇒Copy then Paste	⌘-P
Display⇒Move Caret to Start	Home (R7)
	Shift-Control-Return
Display⇒Move Caret to End	End (R13)
	Control-Return
Find⇒Find and Replace	Control-Find
Find⇒Match Delimiter	⌘-D
Find⇒Replace >field< ⇒Next	Control-Tab
Find⇒Replace >field< ⇒Previous	Shift-Control-Tab
Find⇒Find Selection⇒Forward or	⌘-F
Find⇒Find Clipboard⇒Forward	
Find⇒Find Selection⇒Backward or	Shift-⌘-F
Find⇒Find Clipboard⇒Backward	

Except as noted below, a menu accelerator operates just like the menu item with the same name.

- **Esc** is an accelerator for **Load File** and **Set Directory**. Type the pathname by itself on the first line in the document. Then press **Esc**. If the pathname is the name of a file, the line will be selected automatically and replaced by the contents of the file. If the pathname is the name of a directory, the directory will be changed and the new name will appear in the frame header.
- **Shift-Esc** is an accelerator for **Include File**. Type the filename by itself on the line in the document where you want the file inserted. (The caret must be on the same line.) Then press **Shift-Esc**. The filename will be selected automatically and replaced by the contents of the file.
- **Esc** and **Shift-Esc** and related Meta keys **⌘-L** and **⌘-I** apply only to text windows.

- When there's a selection, **⌘-F** is an accelerator for **Find⇒Find Selection⇒Forward**. Without a selection, it's an accelerator for **Find⇒Find Clipboard⇒Forward**. **Shift-⌘-F** works the same way in the backward direction.

9.7. Other Keyboard Accelerators

Here are some additional keyboard accelerators.

Table 9-6 *Other Keyboard Accelerators*

<i>Other Action</i>	<i>Keyboard Equivalent</i>
Caps_Lock	F1
Quote	⌘-Q

- The **F1** key works like a function key. It operates on the text you type. Normal typing produces capital letters only when you press the **Shift** key. The **F1** key lets you produce all capital letters or toggle back to normal typing, as the **Caps Lock** key does. However, **F1** is a SunView software feature that operates on one window at a time, while **Caps Lock** is a keyboard hardware feature that affects all windows. Press the **F1** key with the keyboard focus in the window whose typing you want to affect. When the window toggles to all capital letters, the word [CAPS] appears in the frame header. Pressing **F1** again toggles the window back to normal typing.
- **⌘-Q** (Quote) lets you “quote” a keyboard accelerator in text instead of executing it. (The character that represents the keyboard accelerator may not be a printable character.) For example, type **⌘-Q ⌘-S** to enter the character string that represents **⌘-S** (Meta-S) as text.

9.8. SunOS 3.x Keyboard Accelerators

For reference, the following table shows the keyboard commands available for SunView in SunOS 3.0, 3.2, and 3.4:

Table 9-7 *SunOS 3.x Keyboard Accelerators*

<i>Erase Action</i>	<i>3.x Keyboard Equivalent</i>
Erase_Char_Backward	Delete
Erase_Char_Forward	Shift-Delete
Erase_Word_Backward	Control-W
Erase_Word_Forward	Shift-Control-W
Erase_Line_Backward	Control-U
Erase_Line_End	Shift-Control-U

<i>Caret Action</i>	<i>3.x Keyboard Equivalent</i>
Go_Document_End	Control-Return

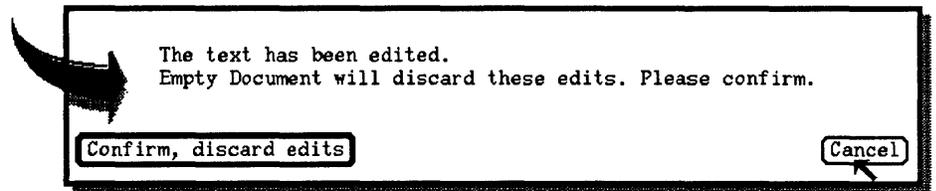
<i>Menu Action</i>	<i>3.x Keyboard Equivalent</i>
Get_Filename_and_Load	Esc
Find_Backward	Shift-Control-F
Find_Forward	Control-F
Paste	Control-G
Cut	Control-D
Copy then Paste	Control-P
Caps_Lock	F1

Additional 3.x accelerators are listed in Table 9-1, “Function Key Accelerators”.

9.9. Alert Button Accelerators

You can respond to alert buttons from the keyboard.

Figure 9-1 *Alert Buttons*



- To reply to the alert message by responding **Confirm** or another obvious *yes*, press the **Return** key. The button that responds to **Return** — the *yes* choice — has a double outline, as shown in Figure 9-1.
- To reply to the alert message by responding **Cancel** or another obvious *no*, press the **Stop** function key.

The Defaults Editor

The Defaults Editor is the key to using SunView comfortably and efficiently. A *default* is a standard way of doing something. For example, pull-right menus are the standard in SunView. The Defaults Editor lets you change the SunView defaults to suit your preferences.

It's a good idea to experiment with the default settings and find out which ones you like best. Later in this chapter you'll find a list of defaults that are likely to be of interest to you.

10.1. Starting the Defaults Editor

The preferred way to start the Defaults Editor is to bring up the SunView menu and choose **Defaults Editor** from the **Editors**⇒ pull-right menu.

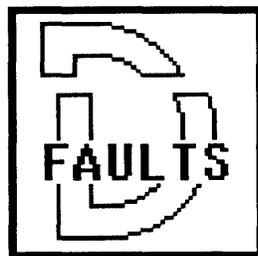
You can also start the Defaults Editor in the background from a Command Tool or Shell Tool by typing:

```
venus% defaultsedit &
```

10.2. Quitting the Defaults Editor

When you're done changing defaults, choose **Quit** or **Close** from the Defaults Editor's Frame menu. If you choose **Close**, you'll see an icon that looks like this:

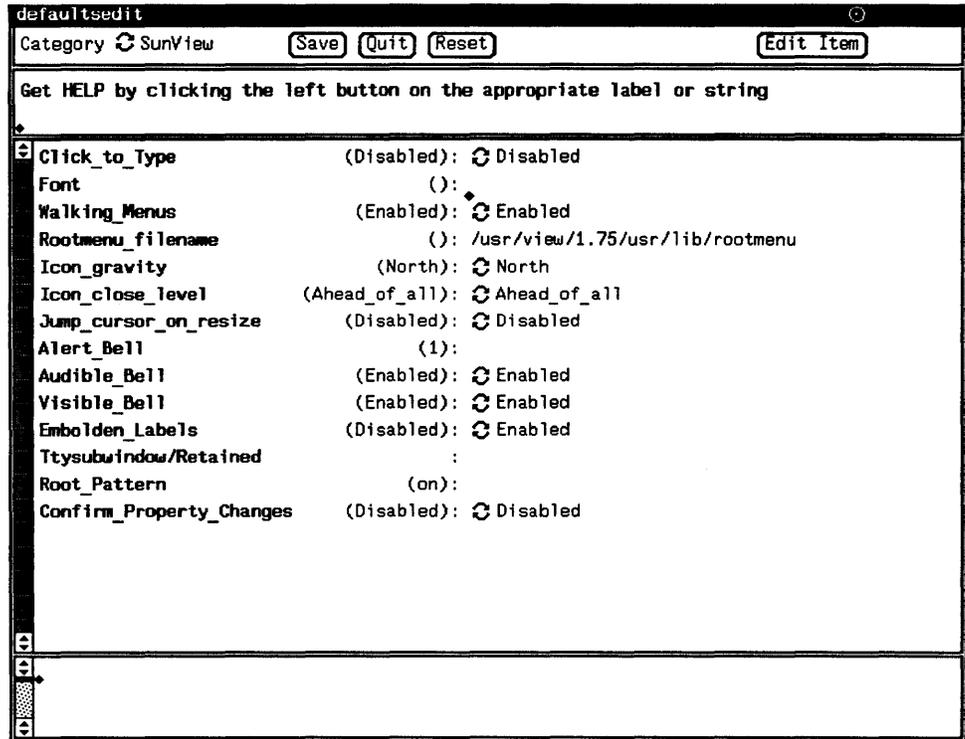
Figure 10-1 *The Defaults Editor Icon*



10.3. The Defaults Editor Windows

The Defaults Editor frame contains three panels and a scratch window, as shown in the following figure.

Figure 10-2 *The Defaults Editor Frame*



From top to bottom you see the Control panel, the Message panel, the Defaults panel, and the scratch window.

The Control Panel

The Control panel lets you manage viewing and changing defaults.

The Category Cycle Item

When you start the Defaults Editor, the SunView category defaults are displayed. There are ten other categories of defaults. They're listed in the **Category** menu, which you bring up by moving the pointer over the **Category** cycle item and holding down the RIGHT mouse button. You can choose any category from the menu to display its defaults in the Defaults panel. You can also click LEFT on the **Category** cycle item to cycle through the categories.

Each category has standard defaults that are taken from files in the `/usr/lib/defaults` directory. The filenames consist of the category name plus a `.d` suffix. Individual defaults in each category are then overlaid by the changes you've made, which are stored in the `.defaults` file in your home directory. The first time you start SunView, you haven't changed any defaults yet so you don't have a `.defaults` file. The file is created the first time you push the **Save** button in the Default Editor's control panel.

If you want to return to all the standard SunView defaults, just rename or delete the `~/.defaults` file.

The Save Button

You can make as many changes to a category as you like, and then change your mind. Changes don't become permanent until you push the **Save** button.

The Quit Button

Pushing the **Quit** button is the equivalent of choosing **Quit** from the Defaults Editor's frame menu.

The Reset Button

If you've made changes to a category and you want the defaults to return to the way they were the last time you pushed the **Save** button, push the **Reset** button.

The Edit Item Button

The **Edit Item** button is for experts who need to modify default lines in the Defaults panel. That's probably all you need to know about it. If you're one of the rare few, you'll find a description in the topic "Editing Default Lines" later in this chapter.

The Message Panel

The **Message** panel is where the Defaults Editor displays text such as the help messages that tell you how a particular default works.

The Defaults Panel

The main element in the Defaults panel is the *default line*. From left to right, the default line contains the default name, the standard SunView setting (in parentheses), and a *default value* — a text item or cycle item where you specify the changed default.

When you click **LEFT** on the default name or the default value, a help message appears in the message panel. The help message for the default item explains the purpose of the default. For cycle items, the help message for the default item explains the effect of a particular setting.

The Defaults panel varies from category to category. For instance, in the **Compatibility** category it also contains explanatory text, and in the **Scrollbar** category it also contains a panel button.

The Scratch Window

The scratch window in the Defaults Editor frame is similar to the one in the Text Editor frame. It's a text window used for temporary editing.

The scratch window is provided so that you can easily edit long text items that are not fully visible in the Defaults panel. Just copy the text item into the scratch window, edit it there, and paste it back where it came from.

10.4. Cycling Through Categories

The defaults that the Defaults Editor works with are divided into *categories* — related defaults are grouped together.

It's a good idea to familiarize yourself with the various categories by going down the list of default names in each category and viewing the help messages associated with them.

To cycle to a category, click the LEFT mouse button on the cycle symbol until the category you want is displayed. Click or hold down **(Shift)-LEFT** to cycle backward.

You can also cycle directly by holding down the RIGHT mouse button on the cycle symbol to open the **Category** menu, then choosing the category you want displayed.

The categories are described below in alphabetical order:

Compatibility Category

The **Compatibility** category lets you change SunView defaults to correspond to previous releases of SunView. For example, you can request the SunView 3.0 Text menu instead of the Text menu described in this manual.

Defaults Category

The **Defaults** category contains defaults that affect the use of the Defaults Editor itself. It's used mainly by software developers.

Help Category

The **Help** category provides the name of the directory that contains the help text files and the location of the `more help` program.

Indent Category

The **Indent** category sets defaults for the `indent` command, described in the *SunOS Reference Manual*. This category affects the action of the **Pretty-print C** menu item in the **Extras**⇒ pull-right menu.

Input Category

The **Input** category is concerned with defaults for the mouse and the keyboard.

The normal way to change input defaults is through the Defaults Editor. However, if you have changed input defaults in the SunOS operating system by using the `stty` command, you can tell the Defaults Editor to read in those changes by pushing the panel button, **Click to start setup program: defaults_from_input**.

Mail Category

The **Mail** category sets defaults for Mail and Mail Tool. The defaults are described in the *SunOS User's Guide: Getting Started* manual.

Menu Category

The **Menu** category lets you change the appearance of menus, determine how menu defaults are set, and specify what happens to the pointer after you choose from a menu.

- Scrollbar Category** The **Scrollbar** category lets you change the appearance of the scrollbar. For example, the default is to display the scrollbar all the time, but you can have the scrollbar displayed only when you're scrolling.
- The `scrolldefaults` program demonstrates the effect of the defaults in this category. To start the program, push its button in the **Defaults** panel.
- SunView Category** The **SunView** category lets you change defaults that are universal throughout SunView such as the font that is used in frame headers and elsewhere. For example, this is where you change the defaults for icon gravity and keyboard focus.
- Text Category** The **Text** category lets you change characteristics of text windows and command windows.
- Tty Category** The **Tty** category lets you change certain characteristics of tty windows and command windows.
- 10.5. Changing Defaults** You can change defaults in one category or several categories.
- Default values are set either by text items or cycle items.
- To change a default value that's a text item, type or edit as you would any panel text item. If the text is too long to be edited in the Defaults panel, copy it into the scratch window, edit it there, and then paste it back into the text item. This technique may be required for long pathnames.
 - To change a default value that's a cycle item, click **LEFT** on the cycle item or choose from its menu. Each new setting causes a new help message to appear in the message window.
- Saving Changes** To save the current category, push the **Save** button. You should save the changes before cycling to the next category you want to change. If you forget, an alert message will remind you.
- If you *don't* want to save changes — if you've changed your mind or made an error — push the **Reset** button.
- When Changes Take Effect** Changes to defaults take effect at different times, depending on the category you're changing. The following table shows when changes take effect for all the categories.

Table 10-1 When Changed Defaults Take Effect

<i>Category</i>	<i>When Changed</i>
Compatibility	When a tool is started.
Defaults	Next time SunView is started.
Help	When text is entered in file in directory.
Indent	Next time you run indent.
Input	Immediately.
Mail	When you start Mail Tool or choose Source .mailrc.
Menu	SunView menu: Next time SunView is started. Other menus: When a tool is started.
Scrollbar	When a tool is started.
SunView	Next time SunView is started.
Text	When a tool is started.
Tty	When a tool is started.

10.6. Some Interesting Defaults

Here are some defaults you should consider including in your experiments with the Defaults Editor. They're popular with many SunView users. The name of the category precedes the default name in the following list:

Input/Left_Handed

If you've decided to put your function keys on the right side of the keyboard and use your mouse on the left side, you need to tell SunView by setting Input/Left_Handed to Yes. The default is No.

Menu/Center_string_items

By default, names of menu items are left-justified. Some users find that centered menu items are easier to read. To center menu item names, set Center_string_items to True.

Menu/Default_Selection and Initial_Selection

The default setting for these menu items is Default. Change the setting to Last_Selection if you want the default for pull-right menus to be the menu item you chose last time you brought up the menu. Each pull-right menu has a separate default it "remembers."
Caution: You have to remember too, because blind choices can get you into trouble. For instance, your last choice in the Edit⇒ pull-right menu may have been Undo All Edits.

Menu/Font, Text/Font, and SunView/Font

If you don't like the fonts you see on the screen, you can change them by specifying any of the fonts in the /usr/lib/fonts/fixedwidthfonts directory, or by designing your own fonts with the Font Editor.

Menu/Stay_up

If you'd rather not keep holding down the mouse button while you decide on a menu item, you'll like stay-up menus. When you click **RIGHT**, the stay-up menu appears, and it stays there until you click **RIGHT** again. You can preview it at your leisure. To change to stay-up menus, set `Stay_up` to `True`.

Scrollbar/Vertical_bar_placement

By default, the scrollbar is on the left side of the window. If you notice that the pointer often strays into the scrollbar when you're trying to select the beginning of a line, see if setting `Vertical_bar_placement` to `East` works better for you. That puts the scrollbar on the right side of the window. Another solution is to increase the width of the left margin — the space between the frame border and the leftmost text in the window — by changing the value of `Text/Left_margin` from the default width of four pixels.

SunView/Click_to_Type

The default keyboard focus is `Pointer-in-Window`. To change to `Click-to-Type`, described in Chapter 4, set `Click_to_Type` to `Enabled`.

SunView/Embolden_Labels

You can make the text in frame headers stand out more by changing `SunView/Embolden_Labels` to `Enabled`. The default is `Disabled`.

SunView/Icon_gravity

The default icon gravity, `North`, puts icons at the top of the screen when frames are closed. To keep icons on the left side of the screen, cycle `Icon_gravity` to `West`, and so on.

Text/Adjust_is_pending_delete

If you want the primary selection to become pending-delete whenever you adjust it, set `Adjust_is_pending_delete` to `True`. `Adjust_is_pending_delete` has the same effect as holding down the **(Control)** key when adjusting a selection. This setting is useful for editing, since ordinarily you adjust a primary selection prior to replacing or deleting it. You can make a pending-delete selection into a normal selection by adjusting it while holding down the **(Control)** key. The default setting for `Adjust_is_pending_delete` is `False`.

Text/Auto_indent

If you want each new line of text you type to have the same white space as the previous line, set `Auto_indent` to `True`. Lines you insert using the **(Copy)** or **(Paste)** function keys are not affected. This setting is often preferred by programmers. The default setting for `Auto_indent` is `False`.

Text/Checkpoint_frequency

You can ask to have your Text Editor document checkpointed every *n* edits. The checkpoint filename ends with the suffix `%%`. If you're editing the `lesson1.limerick` file and the `Checkpoint_frequency` default value is 20, a checkpoint file named `lesson1.limerick%%` is written every twenty edits. The default is 0 (no checkpointing).

Text/Edit_back_char

`Edit_back_char` is the delete character. The default is the value produced by pressing the `Delete` key. To make the `Back Space` key the default, change the `Edit_back_char` default value to `\^H`.

Text/Insert_makes_caret_visible

You might prefer to have text windows set up so that when you insert text, the caret becomes visible even if it wasn't visible before. To do that, cycle `Text/Insert_makes_caret_visible` to `Always`. The default is `If_auto_scroll`.

Text/Scratch_window

You can specify the number of text lines in the Text Editor's scratch window. Extra lines are useful if you do a lot of work in the scratch window. For example, when `Scratch_window` is set to 5, a five-line scratch window is created whenever you start a Text Editor. The default value is a one-line scratch window.

Tty/Text_wraparound_size

The Console writes its history into the `/tmp` directory. The default buffer size is 0, which gives you an infinite buffer. You can avoid filling up `/tmp` by changing `Tty/Text_wraparound_size` to a reasonable number of characters — say, 50,000 — after which the space will be reused starting over at the beginning.

10.7. Editing Default Lines

Normally, the default line is read-only except for the text item or cycle item where you set the default value. However, the Defaults Editor gives you the capability of editing default lines in the various categories.

NOTE: You shouldn't attempt using the **Edit Item** button unless you're a "power user."

The **Edit Item** button has a menu containing the following menu items:

- **Copy Item** duplicates the selected default line. (You select a default line by clicking `LEFT` anywhere in the line.) You can then edit the default name and the default value of the new item. Only items with text or numeric values can be copied. **Copy Item** is useful when you want to change several instances of an item type — for example, to insert a new mail alias into the defaults database.
- **Delete Item** removes the selected default line from your private defaults database. You can't permanently delete a default that's defined in the

system database, but you can make it behave like an undefined item by giving it the special value `\255Undefined\255`. The first and last bytes of this string have the ASCII code 255.

- **Edit Label** lets you edit the default name of the selected default line. When you choose **Edit Label**, the default name in the selected line changes from bold to normal font. You can then select the default name and edit it as you would any panel text item.

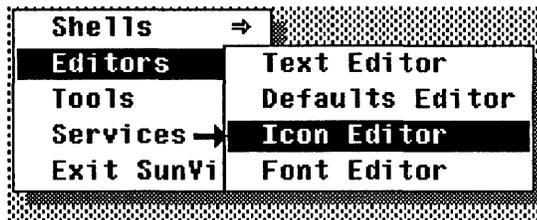
All about Tools

This chapter contains brief descriptions of the tools in the standard SunView menu. A *tool* is a shell, editor, or other application that runs in SunView. The tools are discussed in the same order as they appear in the SunView menu. For more information about a tool covered in this chapter, look up its command name in the “Tool Names Cross-Reference”, Table 11-1, then find the command in the *SunOS Reference Manual*.

11.1. Starting a Tool

To start a tool, choose its menu item from the SunView menu, as shown in the following figure.

Figure 11-1 *Starting a Tool*



You can also start a tool from a SunOS command line, including a command line in Command Tool, Shell Tool, or any other SunView command window.

- To start a tool from a command line, type:

```
venus% command [&]
```

command is the name by which the tool is known to the SunOS operating system. The command names for all of the tools in the SunView menu are cross-referenced in Table 11-1.

& is an optional parameter that causes the tool to run “in the background” — not the SunView background display, but the SunOS job control background described in the *SunOS User’s Guide: Getting Started* and the *SunOS User’s Guide: Doing More* manuals.

The following example shows you how to start tools from a command line so that the Defaults Editor runs in the background and the Text Editor runs in the foreground:

```
venus% defaultsed &
venus% [1] 6062
venus% textedit
```

The first line in the example starts the Defaults Editor in the job control background. The second line contains SunOS job control feedback. The third line starts the Text Editor in the foreground.

11.2. Tools Cross-Reference

The following table lists the tools in the SunView menu and indicates the SunOS command name associated with each one. You can look the command names up in the *SunOS Reference Manual* for further information.

Table 11-1 *Tool Names Cross-Reference*

<i>SunView Menu</i>	<i>Command in SunOS</i>
Command Tool	cmdtool
Shell Tool	shelltool
Graphics Tool	gfxtool
Text Editor	textedit
Defaults Editor	defaultsed
Icon Editor	iconedit
Font Editor	fontedit
Mail Tool	mailtool
Dbx (debug) Tool	dbxtool
Performance Meter	perfmer
Clock	clock
Lock Screen	lockscreen

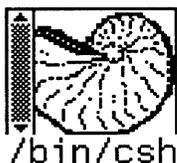
11.3. Shells

The Shells⇒ pull-right menu contains the names of SunView shells. A *shell* is a window that accepts SunOS commands, such as Command Tool and Shell Tool. Most SunView users work with several shells open at the same time.

Command Tool

Command Tool is a command interpreter that displays a standard SunView text-based command window with scrolling. For more information on command windows and the **Cmd Modes**⇒ pull-right menu, see the topic “Command Windows” in Chapter 5 and the topic “The Cmd Modes⇒ Pull-Right Menu” in Chapter 8.

Command Tool is an enhancement designed to replace Shell Tool. You can do anything in Command Tool that you can do in Shell Tool. In addition, Command Tool gives you the advantages of scrolling and the availability of the full Text menu rather than the limited Tty menu.



The Command Tool icon is shown to the left. The Command Tool Frame is shown in the following figure.

Figure 11-2 *Command Tool Frame*

```
cmdtool - /bin/csh
venus% ls *.doc
patrick_henry.doc      sched.doc
venus% head -3 patrick_henry.doc
I don't know why they always misquote me about liberty.
What they say I said ain't what I said at all.
Here are some of the remarks I've made recently on the subject.
venus% fgrep -n "Give me liberty" *.doc
patrick_henry.doc:4:Give me liberty or I'll give you what-for.
patrick_henry.doc:5:Give me liberty or I'll give you a piece of my mind.
patrick_henry.doc:7:Give me liberty or give me a salami on rye.
patrick_henry.doc:8:Give me liberty or give me a raise.
patrick_henry.doc:9:Give me liberty or give me a free lunch.
patrick_henry.doc:10:Give me liberty or give me a break.
patrick_henry.doc:11:What I want for my birthday is: Give me liberty.
patrick_henry.doc:12:You know what I always say--Give me liberty!
patrick_henry.doc:13:Give me liberty or I'll go on strike.
patrick_henry.doc:14:"Give me liberty" seems to lose its meaning after a while.
patrick_henry.doc:15:Maybe the boy who cried wolf really cried, "Give me liberty
!"
venus% cp patrick_henry.doc liberty.doc
venus% ls *.doc
liberty.doc            patrick_henry.doc      sched.doc
venus% get_selection | cat
If liberty is all it's cracked up to be, gimme some.
Give me more
and more
and more
and more
and more
and more
venus%

```

Shell Tool

Shell Tool is a command interpreter that displays a tty window. For more information on tty windows and the Tty menu, see the topic "Tty Windows" in Chapter 5 and the topic "The Tty Menu" in Chapter 8.

Command Tool has all of Shell Tool's features plus some that Shell Tool lacks. Even if you've grown accustomed to Shell Tool, you may want to try Command Tool for the same tasks and see if you prefer it.



The Shell Tool icon is shown to the left. The Shell Tool Frame is shown in the following figure.

Figure 11-3 *Shell Tool Frame*

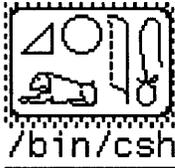
```
shelltool - /bin/csh
venus% ls *.doc
liberty.doc          patrick_henry.doc    sched.doc
venus% head -3 patrick_henry.doc
I don't know why they always misquote me about liberty.
What they say I said ain't what I said at all.
Here are some of the remarks I've made recently on the subject.
venus% fgrep -n "Give me liberty" *.doc
liberty.doc:4:Give me liberty or I'll give you what-for.
liberty.doc:5:Give me liberty or I'll give you a piece of my mind.
liberty.doc:7:Give me liberty or give me a salami on rye.
liberty.doc:8:Give me liberty or give me a raise.
liberty.doc:9:Give me liberty or give me a free lunch.
liberty.doc:10:Give me liberty or give me a break.
liberty.doc:11:What I want for my birthday is: Give me liberty.
liberty.doc:12:You know what I always say--Give me liberty!
liberty.doc:13:Give me liberty or I'll go on strike.
liberty.doc:14:"Give me liberty" seems to lose its meaning after a while.
liberty.doc:15:Maybe the boy who cried wolf really cried, "Give me liberty!"
patrick_henry.doc:4:Give me liberty or I'll give you what-for.
patrick_henry.doc:5:Give me liberty or I'll give you a piece of my mind.
patrick_henry.doc:7:Give me liberty or give me a salami on rye.
patrick_henry.doc:8:Give me liberty or give me a raise.
patrick_henry.doc:9:Give me liberty or give me a free lunch.
patrick_henry.doc:10:Give me liberty or give me a break.
patrick_henry.doc:11:What I want for my birthday is: Give me liberty.
patrick_henry.doc:12:You know what I always say--Give me liberty!
patrick_henry.doc:13:Give me liberty or I'll go on strike.
patrick_henry.doc:14:"Give me liberty" seems to lose its meaning after a while.
patrick_henry.doc:15:Maybe the boy who cried wolf really cried, "Give me liberty!"
venus% rm liberty.doc
rm: remove liberty.doc? y
venus% █
```

Graphics Tool

Graphics Tool displays two windows for running graphics programs.

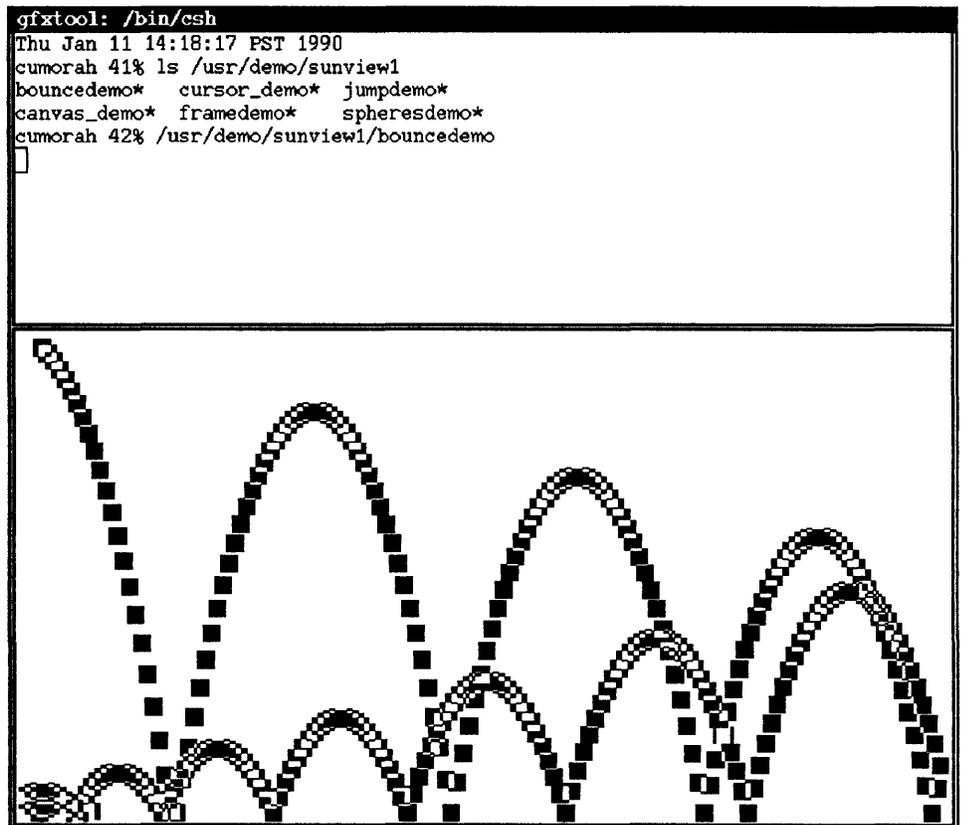
The upper window is a tty window (just like Shell Tool) where you can type commands that run graphics programs and other SunOS commands.

The lower window, which is gray when Graphics Tool starts, displays the graphics output of the command running in the tty window.



The Graphics Tool icon is shown to the left. The Graphics Tool frame is shown in the following figure.

Figure 11-4 *Graphics Tool Frame*



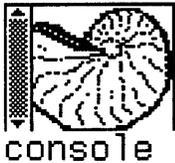
Console

The Console frame displays error messages and other messages from SunView and SunOS. If you do not have a Console window, any error or system messages will appear on the background of your screen.

A Command Tool is the recommended system console, but you can start a Shell Tool as your Console frame by modifying your `rootmenu`. See Chapter 8 "All about Menus" for information about modifying your `rootmenu`.

If you have more than one window identified as `Console` in the header, only the last one created will receive the messages. The others are just be normal `tty` windows.

The system messages that appear in the Console window are written into the `/tmp` directory. If your system generates a great many messages, they may use up all the available space in `/tmp`. You can avoid this potential problem by using the Defaults Editor to set `Tty/Text_wraparound_size` to a reasonable number; say, 50,000 characters. Then when the space you've allocated has been used up, the Console frame will reuse it by wrapping around to the beginning.



The Console icon is shown to the left. The Console frame is shown in the following figure.

Figure 11-5 *Console Frame*



11.4. Editors

The **Editors**⇒ pull-right menu contains the names of tools for creating and revising documents, icons, and fonts, plus a tool that lets you change SunView defaults.

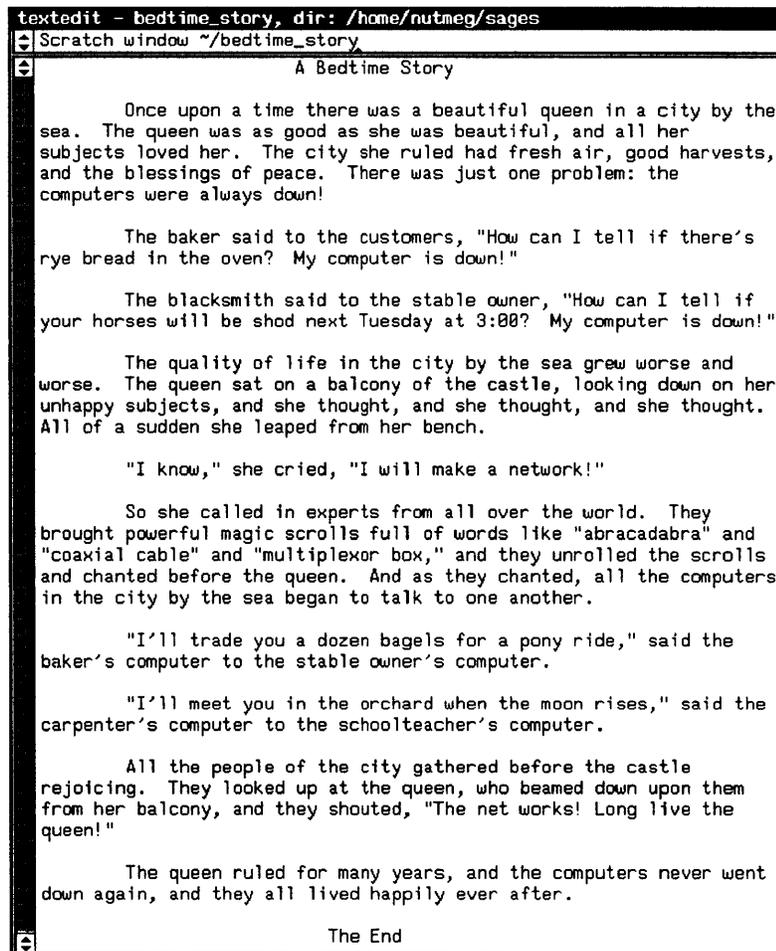
Text Editor

The Text Editor lets you create and revise text files. This manual describes the Text Editor in detail. The *Lessons* section provides an introduction for first-time users of window-based editors.



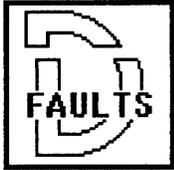
The Text Editor icon is shown to the left. The Text Editor frame is shown in the following figure.

Figure 11-6 *Text Editor Frame*



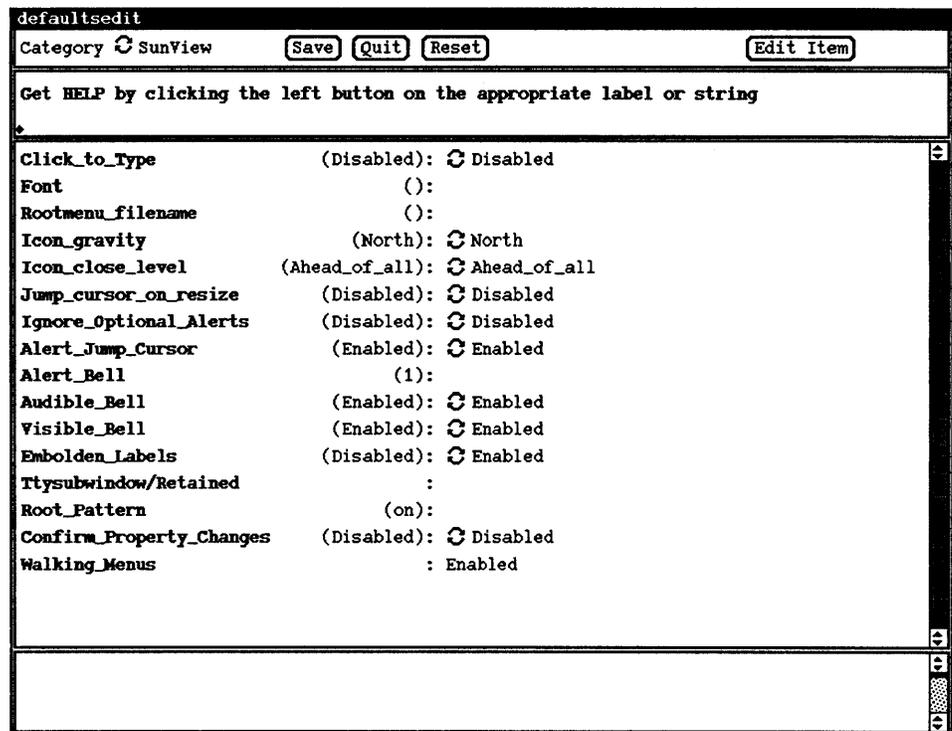
Defaults Editor

The Defaults Editor is a tool for customizing SunView by changing the standard defaults. This editor has a whole chapter devoted to it. You'll find a description of its windows and a discussion of some important defaults in Chapter 10.



The Defaults Editor icon is shown to the left. The Defaults Editor frame is shown in the following figure.

Figure 11-7 *Defaults Editor Frame*



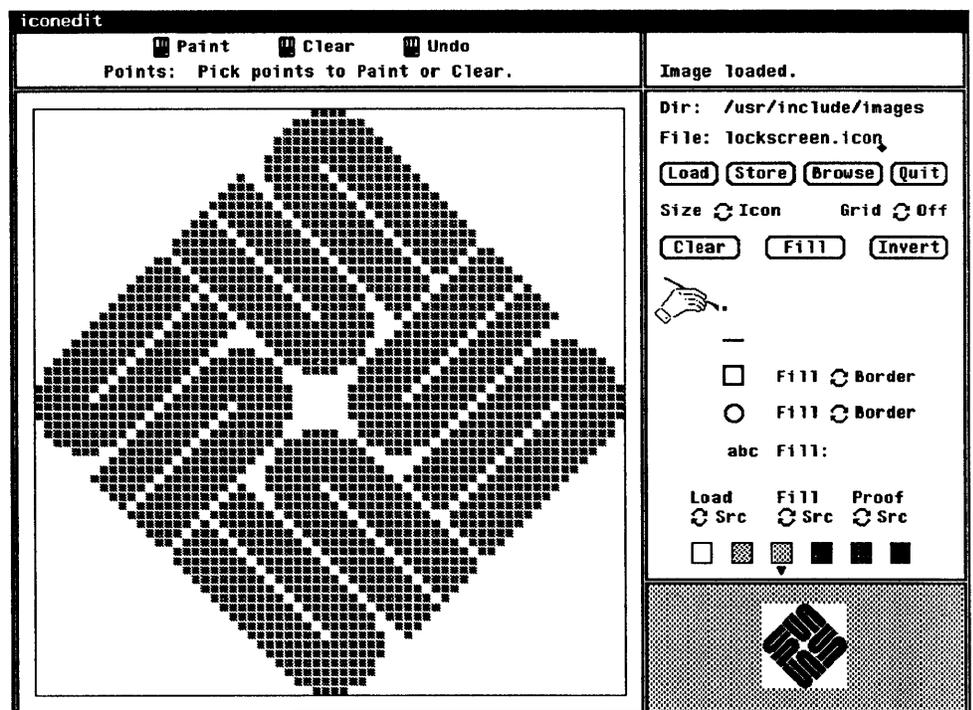
Icon Editor



The Icon Editor is a tool for creating and revising icon and cursor images. You can create a new icon or cursor and store it in a file. You can also load a file that contains an icon or cursor image, edit it, and save it.

The Icon Editor icon is shown to the left. The Icon Editor frame is shown in the following figure.

Figure 11-8 *Icon Editor Frame*



To load an image:

- In the **File:** text item in the control panel, type or copy the pathname of the file that contains the image you want. (You can find standard SunView icons, pointers, and cursors in the `/usr/include/images` directory.)
- Push the **Load** button by clicking **LEFT** on it. The image appears in the drawing canvas, the large window on the left side of the Icon Editor frame.

To store an image:

- In the **File:** panel text item, type or copy the pathname of the file where you want to save the image. If you're editing an existing file, the pathname is already there.
- Push the **Store** button.

To create or change an image:

- Set the control panel on the right side of the frame to draw points, lines, rectangles, circles, or text. Then click **LEFT** to draw a point or paste the fill text into the drawing canvas, or hold down **LEFT** and drag to draw a line, rectangle, or circle.
- Click **MIDDLE** or hold down and drag **MIDDLE** to erase.

Every change you make in the drawing canvas is reflected in the proof window in the lower right corner of the Icon Editor frame. The proof window shows you a life-size version of the image as it would appear on the SunView background.

To save a new image:

- Type its filename in the **File:** text item and push the **Store** button. If necessary, type a new directory pathname.

For more information on the Icon Editor, see the `iconedit` manual page online or in the *SunOS Reference Manual*.

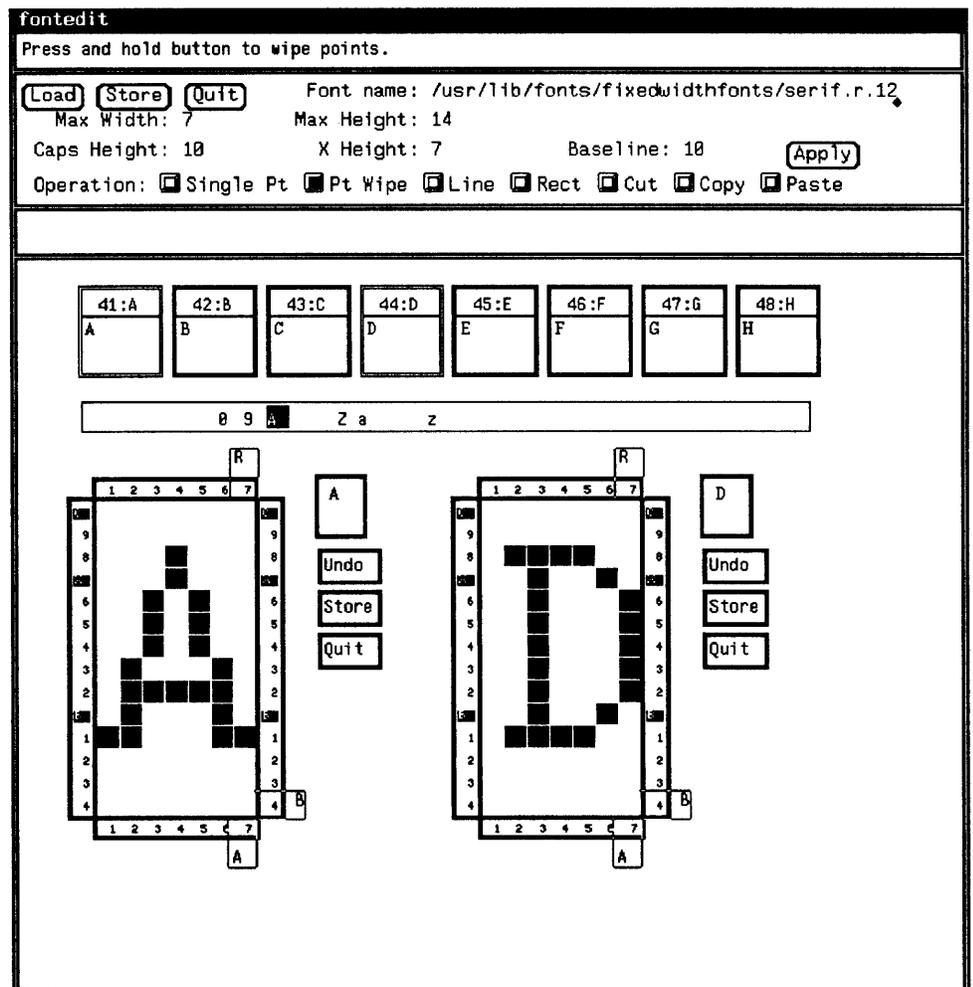
Font Editor



The Font Editor is a tool for creating and revising font images.

The Font Editor icon is shown to the left. The Font Editor frame is shown in the following figure.

Figure 11-9 *Font Editor Frame*



To load an existing font:

- Type its pathname in the **Font name:** text item in the control panel. You will find the standard SunView fonts in the `/usr/lib/fonts/fixedwidthfonts` directory.
- Push the **Load** button.

To load an individual character:

- Click **LEFT** in the slider to indicate the range of characters you want displayed. The slider is the long horizontal box that shows you where to click to get characters in the range of 0–9, A–Z, and a–z. Eight characters at a time are displayed in edit buttons above the slider.
- Click **LEFT** in an edit button to display it in enlarged format in an edit pad below the slider.
- Set the **Operation** choice item in the control panel to indicate what operation you want to perform.

To save an individual character:

- Push the rectangular **Store** button next to the enlarged image of the character.

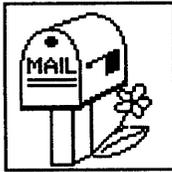
To save an entire font:

- Change the **Font name:** text item if necessary, and push the lozenge-shaped **Store** button in the control panel.

For further instructions on creating and editing fonts, see the `fontedit` command online or in the *SunOS Reference Manual*.

11.5. Other Tools

Mail Tool

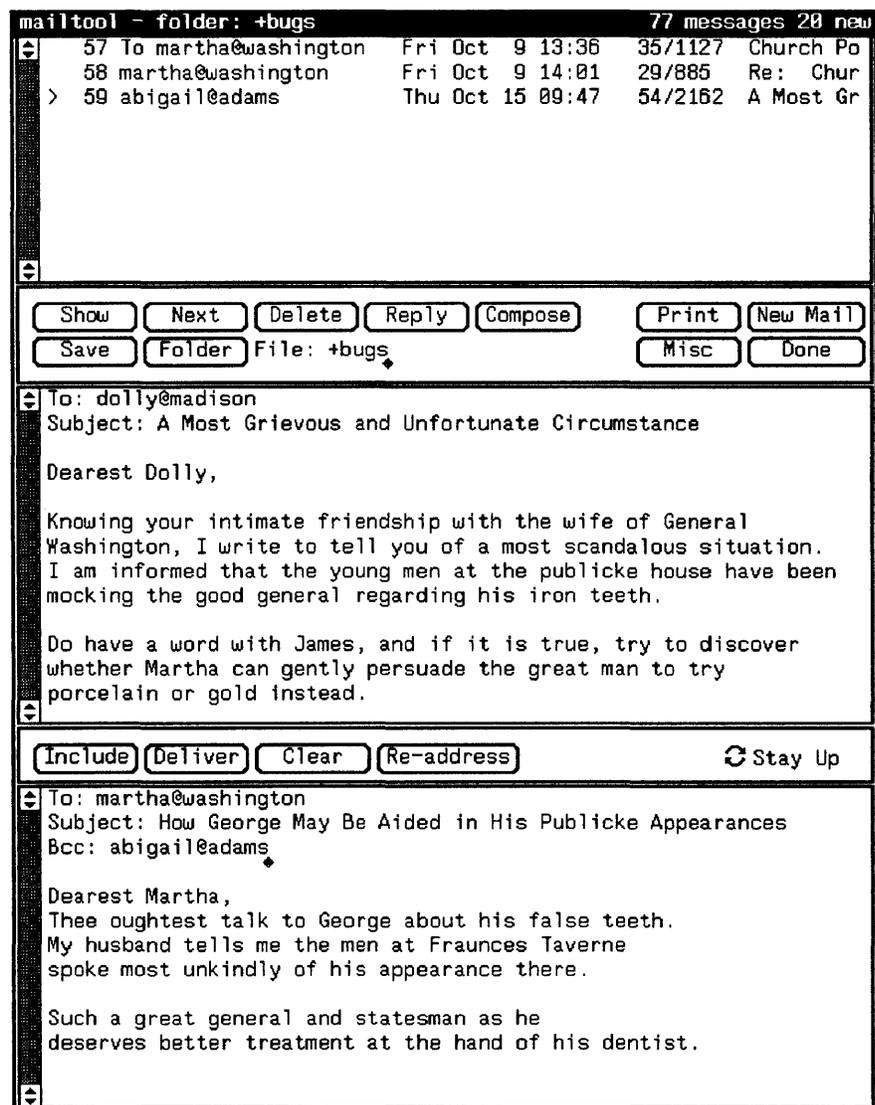


The Tools⇒ pull-right menu contains the names of some important applications.

Mail Tool provides windows for sending and receiving mail messages over the network. The *Mail and Messages: Beginner's Guide* manual contains a complete description of this tool. You'll enjoy reading the sample mail.

The Mail Tool icon is shown to the left. The Mail Tool frame is shown in the following figure.

Figure 11-10 *Mail Tool Frame*



Dbx (debug) Tool

Dbx Tool is a source-level debugger for C, Pascal, and FORTRAN 77 programs. See the `dbxtool` command in the *SunOS Reference Manual* for information on how to use the debugger.



The Dbx Tool icon is shown to the left. The Dbx Tool frame is shown in the following figure.

Figure 11-11 *Dbx Tool Frame*

```

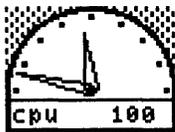
dbxtool
Awaiting Execution
File Displayed: ./example.c                               Lines: 13-32
*/
    struct few few2 = { 3, 4, NULL, "world" };
    struct few few1 = { 1, 2, &few2, "hello" };
/*
 * write a main program to use the structures
 */
main()
{
/*
 * declare the variable *fewp
 * to p[oint to a few-type structure
 */
    struct few *fewp;
/*
 * print out a message
 */
    for (fewp = &few1; fewp != NULL; fewp = fewp -> next) {
        printf("%s ", fewp -> message);
    }
}

print print * next step stop at cont stop in clear where
up down run

Reading symbolic information...
Read 155 symbols
(dbxtool) run
Running: example
hello world
execution completed, exit code is 0
program exited with 0
(dbxtool) stop at "example.c":29
(2) stop at "example.c":29
(dbxtool) print fewp
"fewp" is not active
(dbxtool)

```

Performance Meter

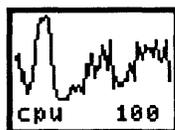


Performance Meter is a tool for observing system performance.

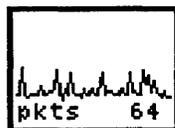
The Performance Meter icon, shown on the left, is a speedometer-like gauge whose needles move as system conditions change. The short needle tracks average performance, the long one tracks current performance. The moving display is updated about once a second.

The performance option being measured is shown in the lower left corner of the icon or frame. Its maximum value is shown in the lower right corner. For example, since CPU performance is measured as a percentage, its maximum value is 100.

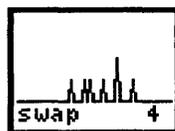
The Performance Meter frame, which is the same size as its icon, is a graph that changes as system conditions change. The **Tools⇒Performance Meter** pull-right menu contains the following options, shown in frame (open) format:



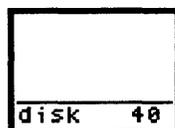
Percent CPU Used — processor performance.



Ethernet Packets — network performance.



Swapped Jobs — memory performance.



Disk Transfers — I/O performance.

There are several additional options for Performance Meter. For example, it can monitor performance on a remote machine such as your fileserver. To change the display, choose another option from its menu:

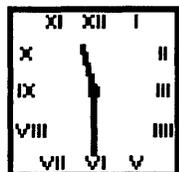
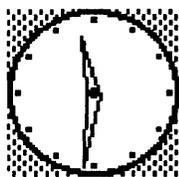
- Move the pointer over the Performance Meter icon or frame and hold down the RIGHT mouse button to bring up its menu. Highlight the option you want, then release the mouse button.

For more information, see the `perfmeter` command in the *SunOS Reference Manual*.

Clock

Clock is a tool that displays a clock face icon showing the current time of day — the hands move.

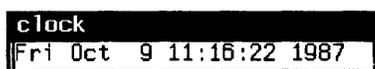
You can choose either a round or a rectangular clock icon, as shown to the left. The SunView menu displays the icons without the hands.



The Clock frame displays the day, full date, and year as well as the time, as shown in the following figure.

To update the information displayed by the Clock, use the `date` command described in the *SunOS Reference Manual*. To add additional Clock options, such as a second hand and a day-and-date display, see the `clock` command in the *SunOS Reference Manual*. to find out what options to specify, then customize your SunView menu, as described in Chapter 8, to include the optional Clock features,

Figure 11-12 *Clock Frame*



SunView Terminology

This chapter contains a summary of terms used throughout this manual. Related terms are grouped together under the following headings:

- General SunView terms.
- Mouse and pointer terms.
- Text terms.
- Scrolling terms.
- Menu terms.
- Frame and window terms.

12.1. General SunView Terms

<i>inverse video</i>	Method of displaying text or image in white on black background.
<i>highlight</i>	Display in inverse video. Highlighting indicates selected text or menu item that can be chosen.
<i>alert window</i>	A panel that is displayed in response to an action you request. It contains informative messages, warnings, and choices represented by panel buttons.

12.2. Mouse and Pointer Terms

<i>LEFT</i>	The LEFT mouse button, used to select text, push panel buttons, and make choices.
<i>MIDDLE</i>	The MIDDLE mouse button, used to adjust or extend selections and move frames.
<i>RIGHT</i>	The RIGHT mouse button, used to bring up menus and choose menu items.
<i>pointer</i>	The distinctive shape, usually an arrow pointing “north-west,” whose position on the screen is moved by the mouse. The pointer shape is a target when it’s on the frame border, a right arrow when a menu is up, and an hourglass when you must wait for an operation to complete. The pointer takes other shapes when it’s in the scrollbar and in certain applications.

<i>point</i>	Position the pointer by moving the mouse.
<i>click</i>	Press and release a mouse button.
<i>hold down</i>	Press a mouse button and continue to hold it down while performing another operation — for instance, while previewing a menu or choosing a menu item. When this manual tells you to hold down a mouse button, keep holding it down until the manual tells you to release it.
<i>select</i>	Highlight text by pointing and clicking LEFT. Selected text is marked for subsequent operations.
<i>selection</i>	Highlighted text that is recognized by SunView for menu and function key operations.
<i>secondary selection</i>	An underlined temporary selection that you make while pressing a function key.
<i>pending-delete selection</i>	A grayed selection that can be deleted by typing or pasting.
<i>extend the selection</i>	Adjust the current selection to include more (or less) text by pointing and clicking MIDDLE.
<i>keyboard focus</i>	Refers to the window where typing or pasting appears (normally the window that contains the pointer).
<i>Click-to-Type keyboard focus</i>	Lets you click in a window to set the keyboard focus. It remains there, no matter where you move the pointer, until you click in another window.
<i>Pointer-in-Window keyboard focus</i>	The normal keyboard focus in SunView — changes when you move the pointer into another window.
<i>Mouse-to-Type keyboard focus</i>	Another name for Pointer-in-Window keyboard focus.
<i>insertion point</i>	The place where the next character you insert will go. To set the insertion point, position the pointer and click LEFT.
<i>caret</i>	The insertion point indicator. In a window that has the keyboard focus, the caret is a blinking black triangle. In a window without the keyboard focus, the caret is an unblinking gray diamond.
<i>Clipboard</i>	Stores text you copy or cut.
<i>document</i>	A file that is displayed in a text window.
<i>erase text</i>	Remove text from a document without changing the Clipboard.

12.3. Text Terms

12.4. Scrolling Terms

<i>delete text</i>	Remove text from a document and place it on the Clipboard.
<i>scroll</i>	Move the document up or down in a text window so that different portions of text are made visible.
<i>scrollbar</i>	The column at the left of a window, used to scroll the document in the window.
<i>scroll boxes</i>	The boxes at either end of the scrollbar. You click in them to move the document a line or a windowful at a time.
<i>bar</i>	The light gray part of the scrollbar, bounded by the scroll boxes. You click in the bar to position the document in the window.
<i>bubble</i>	The dark gray part of the bar. Shows the relative position of the document in the window. The height of the bubble in the bar is proportional to the amount of text that's currently visible.
<i>thumb</i>	Scroll directly to a location in the document by using the MIDDLE mouse button.

12.5. Menu Terms

<i>menu</i>	A list of actions you can choose from.
<i>menu item</i>	The name of an action in a menu.
<i>bring up a menu</i>	Hold down the RIGHT mouse button.
<i>pull-right menu</i>	A submenu you bring up by pulling right on its menu item.
<i>pull-right menu item</i>	The menu item you pull right on in order to bring up a pull-right menu; for instance, you bring up the Text menu's File ⇒ pull-right menu — which contains several menu items — by pulling right on the File ⇒ pull-right menu item.
<i>highlight a menu item</i>	Indicate a menu item to be chosen.
<i>preview a menu</i>	Hold down the RIGHT mouse button to bring up a menu and highlight various menu items, including those in pull-right menus.
<i>choose a menu item</i>	With the desired menu item highlighted, release the mouse button.
<i>SunView menu</i>	The menu you bring up over the gray background. Contains menu items for starting shells, editors, and tools; obtaining various services; and quitting SunView.
<i>Frame menu</i>	The menu you bring up over a frame or icon. Contains menu items for opening and closing, moving and

resizing, changing an overlapping display, and quitting the tool.

12.6. Frame and Window Terms

Text menu

The menu you bring up in a text window or command window. Contains menu items for file operations, editing, displaying parts of a document, finding and replacing text, and optional extras.

frame

A SunView object through which a tool interacts with the user. The frame is surrounded by a *frame border* where you move the pointer to bring up the Frame menu. A frame can contain one or more windows.

icon

An image that represents a closed frame.

frame header

The black stripe at the top of the frame border that contains information displayed by the tool, such as filename and directory.

window

The part of the frame you work in. There are various types of windows.

text window

A window that lets you display and edit text.

edit window

The main text window displayed by the Text Editor. Usually contains text from a file.

scratch window

An auxiliary text window displayed by the Text Editor, Defaults Editor, and other tools. Can be used as a scratch pad for filenames and short strings of text.

command window

A text window that includes a command interpreter, such as a Command Tool window.

dialog

The interaction between the user and the shell, as in a command window or a tty window.

transcript

The record of the dialog in a command window.

panel

A type of window that can contain various items including panel buttons, text items, cycle items and other choice items, depending on the application. These items can contain hidden menus.

panel text item

An area in a panel where you can type and edit text.

panel button

A representation of an action choice, such as **Replace All** in the **Find and Replace** frame.

push a panel button

Click **LEFT** on the button to make the choice.

choice item

A panel item that lets you specify a condition, such as the delimiter choice buttons in the **Find Marked Text** frame and the **Category** cycle item in the Defaults Editor.

- slider* A type of choice item that lets you slide a marker in a bar to choose a setting.
- cycle item* A type of choice item that lets you cycle through a set of choices until you come to the setting you prefer.
- choose in a choice item* The way you choose depends on the type of choice item, but every choice item also has a menu that you can bring up to make a choice.

Command Line Arguments

This appendix lists the command line arguments available when you start a tool. For each tool you start, you can specify one or more command line arguments. Here is what each column in the following table contains.

- **Flag** is the shorthand version of the argument.
- **Long Flag** is the long version of the argument.
- **Arguments** is the parameter or parameters you specify with the flag.
- **Notes** supplies information about the flag.

Flag	Long Flag	Arguments	Notes
-Wb	-background_color	red green blue	Color of the frame
-Wf	-foreground_color	red green blue	Color of text in the frame
-Wg	-set_default_color		Establishes any values specified for <code>foreground_color</code> and <code>background_color</code> as the defaults
-Wh	-height	lines	Number of rows in the frame
-WH	-help		Provides a list of all available arguments
-Wi	-iconic		When specified, tool appears as icon at startup
-WI	-icon_image	file name	Name of file where picture that appears in the icon is stored
-Wl	-label	string	String that appears in the frame border at the top
-WL	-icon_label	string	String that appears in the icon at the bottom
-Wn	-no_label		Specifies no string appears in the frame at the top
-Wp	-position	x y	Position of the tool frame within the screen. Width and height are specified in pixels starting from the upper lefthand corner of the screen.

Flag	Long Flag	Arguments	Notes
-WP	-icon_position	x y	Position of the icon within the screen. Width and height specified in pixels starting from the upper lefthand corner of the screen.
-Ws	-size	x y	Width and height of the frame expressed in pixels starting from the upper lefthand corner of the screen.
-Wt	-font	file name	Font used in tool window.
-WT	-icon_font	file name	Font used in the icon.
-Ww	-width	number	Specifies width of the tool window

¹⁰ Standard fonts are in `/usr/lib/fonts/fixedwidthfonts`.

Index

Special Characters

- % in checkpoint filename, 127, 164
- % in backup filename, 127
- & runs tool in the background, 167
- > in Text Editor icon, 76
- ◆ keys, 150
- _ in text, 95
- | with `get_selection` command, 107

A

- aborting an operation, 98
- accelerator, 147 *thru* 155
 - alert button, 81, 155
 - caret, 51, 150
 - Close, 121
 - Copy, 133
 - Cut, 133
 - erase, 150
 - Find Selection⇒, 48
 - Find⇒, 140
 - for menu items, 31
 - Front, Back, 149
 - function key, 147
 - Include File, 128
 - introduction to, 27
 - keyboard, 150
 - Load File, 128
 - menu, 147, 150, 153
 - mouse button, 51, 149
 - Move, 36, 122, 149
 - Move Caret to End, 152
 - Move Caret to Start, 152
 - Open, 29, 121, 149
 - Paste, 133
 - Resize, 36, 122, 123, 149
 - Set Directory, 129
 - tutorial, 27 *thru* 51
 - Undo Last Edit, 133
- Adjust_is_pending_delete default, 163
- adjusting a selection, 16, 86, 96
- adjusting with MIDDLE mouse button, 4
- advanced selecting, 95
- Again, in Text menu, 130
 - ignores Undo, 130
 - used with Find, 131
- alert window, 13, 80
 - accelerators, 81, 155

- alert window, *continued*
 - arrows, 80
 - buttons, 80
 - for Resize and Move, 34
- alias, using tty escape sequence, 70
- All Text, in Find and Replace frame, 139
- All Text, in Find and Replace frame, 46
- arrow keys, 71, 152
- Auto_indent default, 163

B

- Back accelerator, 149
- Back Space key, 12, 164
- Back, in Frame menu, 30, 32, 125
- background option in `sunview` command, 55
- background, SunOS job control, 167
- background, SunView
 - customizing, 55
 - inverse video, 56
- backup files, 127
- backward, extending selection, 18
- bar, in scrollbar, 42, 109
- basic editing, 87
- basic scrolling, 109
- Blink Owner button, in Find and Replace frame, 139
- Blink Owner button, in Find and Replace frame, 44
- Blink Owner button, in Find Marked Text frame, 141
- block cursor, 81, 85
- bold font in frame header, 163
- bounding box, 32, 121
- brackets in text
 - inserting, 130, 142
 - removing, 142
- bringing up a menu, 5, 113
- bringing up SunView, 1, 55
- bubble, in scrollbar, 42, 109
- button, choice, 140
- button, mouse, 4
- button, panel
 - in alert window, 14
 - pushing, 14, 44

C

- C debugger, 180
- canceling a secondary selection, 98
- canceling a view, 137

canceling an operation, 98
 canvas, 81, 175
 Capitalize⇒, in Text menu, 142
 Caps Lock key, 154
 caret, 11, 85, 87
 blinking triangle, 87
 display and moving, 43
 insertion point, 87
 move to end of document, 138
 move to start of document, 137
 show at top of window, 137
 caret accelerators, 150
 categories in Defaults Editor, 160
 Category cycle item, in Defaults Editor, 158
 cd command, used in SunView, 1
 Center_string_items default, 162
 Change Line Wrap⇒, in Text menu, 47, 138
 changing command modes, 79
 changing defaults, 161
 changing direction with Shift key, 77
 changing keyboard focus, 66
 changing the rootmenu file, 118
 character, erasing, 88
 character, selecting, 86
 Check Printer Queue, in SunView menu, 116
 checkpoint file, 127
 Checkpoint_frequency default, 164
 choice button, 140
 choosing a menu item, 8
 Click-to-Type keyboard focus, 66
 Click_to_Type default, 163
 clicking, 15
 clicking to select text, 95
 Clip Lines, in Text menu, 50, 138
 Clipboard, 18, 88, 133
 finding contents of, 140
 showing contents of, 140
 clipping long lines, 138
 Clock, 182
 clock command, used in SunView, 168, 182
 Close, in Frame menu, 121
 closing a menu, 6
 Cmd Modes⇒ menu, in Text menu, 144
 cmdtool command, used in SunView, 168, 169
 command
 cd, 1
 clock, 168, 182
 cmdtool, 168, 169
 date, 182
 dbxtool, 168, 180
 defaultsed, 168
 emacs, 152
 fmt, 138, 141
 fontedit, 168, 177
 get_selection, 107
 gfxtool, 168, 171
 grep, 144
 head, 90
 iconedit, 81, 168, 175
 indent, 142, 160

command, *continued*
 lockscreen, 168
 lpr, 49
 mailtool, 168
 more, 145
 perfmeter, 168, 181
 rlogin, 106
 screendump, 55
 shelltool, 168, 170
 stty, 160
 sunview, 55
 swin, 67
 textedit, 168, 173
 toolplaces, 117
 command line, 78
 command modes, 79
 command names cross-reference, 168
 Command Tool, 169
 Command Tool, remote login to, 117
 command window, 78
 comparison of text handling, 106
 Compatibility category, in Defaults Editor, 160
 Console buffer size default, 164
 Console frame, 3, 172
 Constrained, in Frame menu, 33
 Continue, in Tty menu, 145
 Control key, 77, 98, 148
 Control panel, in Defaults Editor, 158
 Copy then Paste, in Text menu, 134
 Copy then Paste, in Tty menu, 145
 Copy, in Text menu, 133
 Copy, secondary, 98, 103
 copying
 a selection, 90
 and pasting text, 90
 between windows, 93
 copying text, 89, 102, 103
 .cshrc file, 70
 cursor, block, 81, 85
 customizing the Extras⇒ menu, 142
 customizing the SunView background, 55
 customizing the SunView menu, 118
 Cut function key, 88
 Cut, in Text menu, 19, 133
 Cut, secondary, 98, 104
 cutting and pasting text, 91
 cutting text, 89
 cycle item, 43, 158
 cycling, in Defaults Editor, 160

D

date command, used in SunView, 182
 Dbx (debug) Tool, 180
 dbxtool command, used in SunView, 168, 180
 default directory in Text Editor, 22
 default, in Defaults Editor, 159
 Default_Selection default, 162
 Defaults category, in Defaults Editor, 160
 Defaults Editor, 157 *thru* 165, 174
 Adjust_is_pending_delete, 163

Defaults Editor, *continued*

- Auto_indent, 163
- Category cycle item, 158
- Checkpoint_frequency, 164
- Click_to_Type, 163
- Control_panel, 158
- cycling through categories, 160
- Defaults panel, 159
- Edit Item button, 159, 164
- Edit_back_char, 164
- Icon_gravity, 163
- Input/Left_Handed, 162
- interesting defaults, 162
- left-handed keyboard use, 62
- Menu/Center_string_items, 162
- Menu/Default_Selection, 162
- Menu/Font, 162
- Menu/Initial_Selection, 162
- Menu/Stay_up, 163
- Message panel, 159
- Quit button, 159
- quitting, 157
- Reset button, 159
- Save button, 159
- scratch window, 159
- Scratch_window, 164
- Scrollbar/Vertical_bar_placement, 163
- starting, 157
- SunView/Font, 162
- Text/Font, 162
- Text/Insert_makes_caret_visible, 164
- Text/Multi_click_space, 95
- Text/Multi_click_timeout, 95
- Tty/Text_wraparound_size, 164
- when changes take effect, 161
- .defaults file, 158
- defaults in menus, 114
- Defaults panel, in Defaults Editor, 159
- defaultsedit command, used in SunView, 168
- Delete key, 12, 88, 164
- deleting and pasting back, 88
- deleting line breaks, 88
- deleting text, 14, 88, 104
- delimiters
 - finding with “Find Marked Text”, 140
 - matching with “Match Delimiter”, 141
- Destroy View, in Text menu, 137
- diagram of function keys, 62
- diamond (◆) keys, 150
- direct scrolling, 42, 110
- directory
 - starting SunView from, 55
 - /usr/include/images, 120, 175
 - /usr/lib/fonts/fixedwidthfonts, 162, 177
- directory, setting, 22, 128
- Disable Page Mode, in Tty menu, 145
- Disable Scrolling, in Text menu, 144
- dismissing a menu, 6
- Display⇒ menu, in Text menu, 134 *thru* 138
- Display⇒ menu, in Text Menu
 - Change Line Wrap⇒, 47
 - Clip Lines, 50

Display⇒ menu, in Text Menu, *continued*

- Move Caret to End, 43
- Move Caret to Start, 43
- Show Caret at Top, 43
- Wrap at Character, 48
- Wrap at Word, 49
- displaying the caret, 43
- document, 9
 - emptying, 129
 - including a file, 39
 - saving, 21
- Done button, in Find and Replace frame, 139
- Done button, in Find Marked Text frame, 141
- double-clicking, 15, 95

E

- Edit Item button, 164
- Edit Item button, in Defaults Editor, 159
- edit window, 9
- Edit⇒ menu, in Text menu, 130 *thru* 134
- Edit⇒ menu, in Text Menu
 - Cut, 19
 - Paste, 19
 - Show Clipboard, 18
 - Undo Last Edit, 23
- Edit_back_char default, 164
- (edited) in frame header, 22
- editing
 - basic, 87
 - enabling in command window, 144
 - shortcuts, 100
 - undoing, 23, 94
- editing defaults, 164
- editing panel text item, 80
- editor, 173
 - Defaults Editor, 174
 - Font Editor, 177
 - Icon Editor, 175
 - Text Editor, 8, 173
- Editors⇒ menu, in SunView menu, 116, 173 *thru* 178
- emacs command, used in SunView, 152
- Embolden_Labels default, 163
- Empty Document, in Text menu, 13, 129
- Enable Editing, in Text menu, 144
- Enable Page Mode, in Tty menu, 68, 145
- Enable Scrolling, in Text menu, 144
- End key, 152
- erase accelerators, 150
- erasing text, 12, 13, 88
- Esc key, 129
- escape sequence, 69
- escape sequences in .ttyswrc, 69
- Exit SunView, in SunView menu, 51, 58, 117
- exiting SunView, emergency, 58
- exposing a frame, 30
- exposing and hiding frames, 81
- extending a selection, 16, 86
- extending with MIDDLE mouse button, 4
- Extras⇒ menu, in Text menu, 141 *thru* 142
 - customizing, 142

F

- F1 key, 154
- |>field<|, 141
- file
 - backup, 127
 - .cshrc, 70
 - .defaults, 158
 - including in text window, 128
 - .indent.pro, 142
 - loading, 38, 128
 - rootmenu, 118
 - saving, 21, 127
 - .suntools, 117
 - .sunview, 57, 82, 117
 - .textswrc, 71
 - .ttyswrc, 68
 - /usr/lib/.text_extras_menu, 142
 - /usr/lib/defaults, 158
 - /usr/lib/rootmenu, 118
- File⇒ menu, in Text menu, 12, 127 *thru* 130
 - Empty Document, 13
 - Include File, 39
 - Load File, 38
 - Set Directory, 22
 - Store as New File, 21
- filtering selections, 78
- Find and Replace** frame, 44, 138
- Find Clipboard⇒, in Text menu, 140
- Find function key, 140
- Find Marked Text, in Text menu, 140
- Find Selection⇒, in Text menu, 140
- Find text item, in **Find and Replace** frame, 139
- Find text item, in **Find and Replace** frame, 45
- Find then Replace**, in **Find and Replace** frame, 139
- Find⇒ menu, in Text menu, 138 *thru* 141
- finding text, 43
- Finishing Up⇒ menu, in Text menu, 129
- flash lines, 7
- fmt command, used in SunView, 138, 141
- font, 162
 - bold in frame header, 163
- Font default, 162
- Font Editor, 177
- fontedit command, used in SunView, 168, 177
- Format, in Text menu, 141
- FORTRAN debugger, 180
- frame, 75 *thru* 82
 - bringing to front, 125
 - closing, 6
 - command name, 9
 - (edited), 22
 - enlarging, 124
 - exposing, 30, 81
 - Find and Replace**, 43, 44
 - Find Marked Text**, 140
 - font in frame header, 163
 - frame header, 6, 9, 22, 76
 - hiding, 30, 81
 - moving, 32, 81, 149
 - moving or resizing, 33
 - opening, 6
- frame, *continued*
 - opening, accelerator, 149
 - overlapping, 29, 81
 - pop-up, 76
 - quitting, 126
 - rescuing, 30
 - resizing, 32, 81, 149
 - sending to back, 125
 - subframe, 76
 - Text Editor, 9
 - tool name, 9
 - uncovering, 30
 - zooming, 124
- Frame menu, 6, 121 *thru* 126
 - Back and Front, 30, 32
 - FullScreen, 37
 - function keys for, 32
 - pop-up frame, 43
 - Props, 8
 - Resize⇒, 33
- frame menu
 - unconstrained resizing, 33
- Frame menu
 - UnZoom, 37
 - Zoom, 37
- Front accelerator, 149
- Front, in Frame menu, 30, 32, 125
- FullScreen, in Frame menu, 37, 124
- function key, 31
 - accelerators, 147
 - Again, 130
 - binding to a filter, 68
 - Cut, 88
 - Delete, 12
 - diagram, 62
 - Find, 48, 140
 - labels, 62
 - left-handed use, 62
 - Paste, 88
 - remapping, 68
 - secondary selection with, 96
 - Stop, 98, 155

G

- get_selection command, used in SunView, 107
- gfxtool command, used in SunView, 168, 171
- glossary, 183 *thru* 187
- Graphics Tool, 171
- gravity, icon, 7
- grayed menu item, 8, 20, 114, 126
- grayed selection, 96, 98
- grep command, used in SunView, 144

H

- head command, used in SunView, 90
- header, frame, see *frame*
- Help** category, in Defaults Editor, 160
- help command, used in SunView, 160
- hiding a frame, 30
- highlighting, 5
 - grayed for pending-delete, 98
- Home key, 152

hot spot in pointer, 84

I

Icon Editor, 175
 icon files, 120
 icon gravity, 7
 Icon_gravity default, 163
 iconedit command, used in SunView, 81, 168, 175
 icons in menus, 51, 120
 images directory, 120
 Include File, in Text menu, 39, 128
 including a file, 39
 Indent category, in Defaults Editor, 160
 indent command, used in SunView, 142, 160
 .indent.pro file, 142
 Initial_Selection default, 162
 Input category, in Defaults Editor, 160
 input focus, *see keyboard focus*
 Input/Left_Handed default, 162
 Insert Brackets⇒, in Text menu, 142
 Insert_makes_caret_visible default, 164
 inserting text, 17, 87
 insertion point, 11, 17, 87
 inverse video, 5, 56

K

key

- arrow, 152
- Back Space, 12, 164
- Caps Lock, 154
- Control, 77, 98, 148
- Delete, 12, 88, 164
- diamond, 150
- End, 152
- Esc, 128, 129
- F1, 154
- Home, 152
- Line Feed, 88
- Meta (◆), 60, 61, 64, 65, 150
- R13, 152
- R7, 152
- Return, 88, 155
- Shift, 148
- Stop, 155

keyboard, 59 *thru* 73

- accelerators, 150
- function keys, 31
- Input/Left_Handed default, 162
- labels, 62
- labels for Type2, 65
- labels for Type3, 64
- labels for Type4, 63
- left-handed user, 4
- Type2, 61
- Type3, 60
- Type4, 60

keyboard accelerators, 31

keyboard focus, 11

keyboard focus, changing, 66

L

labels, 62

- Type2 keyboard, 65
- Type3 keyboard, 64
- Type4 keyboard, 63

leaving SunView, 51, 58, 117
 LEFT mouse button, 4
 left-handed function keys, 62
 left-handed user, 4
 Left_Handed default, 162
 lessons, 1 *thru* 51
 limerick, 15
 line

- getting line number, 134
- selecting by number, 134

line break, deleting, 88
 Line Feed key, 88
 line, erasing, 88
 list, reordering, 105
 Load File, in Text menu, 38, 128
 loading a file, 38
 Lock Screen, 24, 28, 58
 Lock Screen, in SunView menu, 117
 lockscreen command, used in SunView, 24, 168
 login, 28
 login, remote, 117
 lpr command, used in SunView, 49

M

Mail category, in Defaults Editor, 160
 Mail Tool, 49, 179
 mailtool command, used in SunView, 168
 mapping keys, 68
 Match Delimiter, in Text menu, 141
 mechanical mouse, 84
 menu, 113 *thru* 145

- accelerators, 147
- bringing up, 5, 113
- Center_string_items default, 162
- closing, 6
- Cmd Modes⇒, 144
- cycle item's, 46
- Default_Selection default, 162
- defaults, 114
- dismissing, 6
- Display⇒, 134
- Edit⇒, 130
- Extras⇒, 141
- File⇒, 12
- font in, 162
- Frame, 6, 43, 121 *thru* 126
- grayed item, 8, 114
- icons in, 51
- in panel, 46, 115
- Initial_Selection default, 162
- opening, 5
- popping up, 8
- previewing, 10
- previewing in panel, 46
- pull-right, 12
- stay-up, 6, 51, 115

menu, *continued*

- Stay_up default, 163
- SunView, 23, 116 *thru* 120
- Text, 10, 127 *thru* 144
- Tty, 145
- menu accelerators, 150, 153
- Menu category, in Defaults Editor, 160
- menu item
 - choosing, 8
 - grayed, 20
 - selection required, 20
- Message panel, in Defaults Editor, 159
- Meta (◆) keys, 60, 61, 64, 65, 150
- MIDDLE mouse button, 4
- mnemonics for caret accelerators, 152
- more command, used in SunView, 145
- mouse, 3, 83
 - clicking, 15
 - keyboard accelerators, 51
 - left-handed user, 4
 - mechanical, 84
 - optical, 83
 - pointer stuck, 4
- mouse button
 - clicking, 5
 - holding down, 5
 - pressing, 5
- mouse button accelerators, 149
- Mouse-to-Type keyboard focus, 66
- mousing ahead, 78
- Move Caret to End, in Text menu, 43, 138
- Move Caret to Start, in Text menu, 43, 137
- Move, in Frame menu, 121
 - accelerator, 36, 149
 - alert window, 34
 - Constrained, 122
 - Unconstrained, 122
- moving
 - a frame, 33
- moving a frame, 32, 81
- moving a selection, 91
- moving text, 19, 102, 103
- moving the caret, 43
 - from the keyboard, 150
- multi-clicking, 95, 96
- Multi_click_space default, 95
- Multi_click_timeout default, 95

N

- new document, 129
- new file, saving, 21, 127
- newline character, 88

O

- Open accelerator, 149
- Open, in Frame menu, 121
- opening
 - frame, 6
 - menu, 5, 6
- optical mouse, 83

- overlapping frames, 29, 81

P

- pad, 3
- page mode, in tty window, 145
- paging in a window, 110
- panel, 79
 - alert window, 13, 80
 - canvas, 81
 - editing text item in, 80
 - menus in, 46, 115
 - panel button, 44
 - panel items, 43
 - text item, 44
- Pascal debugger, 180
- Paste function key, 88
- Paste, in Text menu, 133
- Paste, secondary, 96, 102
- pasting text, 87, 89
- pending-delete selection, 51, 98
- percent sign, 127
- perfmeter command, used in SunView, 168, 181
- Performance Meter, 181
- pipe (|) symbol with get_selection command, 107
- piping, 138
- piping a selection to a command, 107
- pointer, 3, 84
 - stop sign, 145
 - target, 6
- pointer shapes, 84
- Pointer-in-Window keyboard focus, 66
- pop-up frame, 76
- popping up a menu, 8
- preserving the screen, 24
- Pretty-print C, in Text menu, 142
- previewing a menu, 10, 46
- primary selection, 85
- Print Selected Text, 24
- Print Selected Text, in SunView menu, 117
- printing, 24
 - checking the queue, 116
- Printing=>, in SunView menu, 116
- privacy, 24
- proportional scrolling, 42
- Props, in Frame menu, 126
- pull-right menu, 12
- pushing a button, 14
- pushing a panel button, 4, 44

Q

- quadruple-clicking, 15, 95
- question mark, 14
- quick editing, 100
- Quit button, in Defaults Editor, 159
- Quit, in Frame menu, 126
- quitting SunView, 58, 117
- quoting commands in text, 154

R

R13 key, 152
 R7 key, 152
 rearranging text, 105
 Redisplay All, in SunView menu, 116
 Redisplay, in Frame menu, 126
 refreshing the screen, 126
 reinserting deleted text, 88
 remapping keys, 68
 Remote Login⇒, in SunView menu, 117
 Remove Brackets⇒, in Text menu, 142
 repeating an operation, 130
 Replace |>field<|⇒, in Text menu, 141
 Replace All, in Find and Replace frame, 139
 Replace text item, in Find and Replace frame, 139
 Replace text item, in Find and Replace frame, 45
 Replace then Find, in Find and Replace frame, 139
 replacing text, 43, 44, 92, 102, 103
 rescuing a frame, 30
 Reset button, in Defaults Editor, 159
 resize
 unconstrained, 33
 Resize accelerator, 123, 149
 Resize, in Frame menu, 33, 122
 accelerator, 36
 alert window, 34
 Constrained, 123
 Unconstrained, 122
 resizing
 a frame, 33
 resizing a frame, 32, 81
 resizing a scratch window, 77, 123
 Return character, 47
 Return key, 88, 155
 RIGHT mouse button, 4
 rlogin command, used in SunView, 106
 rootmenu file, 118, 120
 running SunView, 1

S

Save button, in Defaults Editor, 159
 Save Current File, in Text menu, 127
 save frequently, 127
 Save Layout, in SunView menu, 57, 82, 117
 saving
 a document, 21
 automatic backup, 127
 changes in Defaults Editor, 161
 checkpoint file, 127
 screen layout, 82, 117
 shortcuts, 129
 scratch window, 9, 21, 77, 159
 changing size, 77
 resizing, 123
 scrollbar, 40
 Scratch_window default, 164
 screen
 locking, 58, 117
 refreshing, 126
 saving the layout, 57, 117
 screen fonts, 162
 screen, locking, 24
 screen, unlocking, 28
 screendump command, used in SunView, 55
 scroll box, 40, 109
 scrollbar, 40, 109
 bar, 42
 bubble, 42, 109
 in scratch window, 40
 scroll box, 40
 Vertical_bar_placement, 163
 Scrollbar category, in Defaults Editor, 161
 scrolldefaults, 161
 scrolling, 40, 109 *thru* 111
 a line, 109
 a windowful, 109
 backward, 110
 basic, 109
 by line, 41
 by windowful, 41
 continuous, 110
 direct, 110
 directly, 42
 disabling in command window, 144
 down, 110
 enabling in command window, 144
 forward, 110
 paging, 110
 proportionally, 42
 returning to previous location, 110
 thumbing, 110
 to position in document, 110
 up, 110
 windowful, 110
 secondary Copy, 98, 103
 secondary Cut, 98, 104
 secondary Paste, 96, 102
 secondary selection, 96 *thru* 105
 canceling, 98
 problem, 98
 stuck, 98
 with Find function key, 98, 140
 Select Line at Number, in Text menu, 134
 selecting
 a character, 86
 advanced, 95
 basic, 85
 by line number, 134
 extending, 86
 word, 15
 selecting text, 14, 86
 selecting with LEFT mouse button, 4
 selection, 83 *thru* 108
 adjusting, 16, 86
 caret, 87
 copying, 90
 extend by wiping, 17
 extending, 16
 extending backward, 18
 extending multi-click, 96
 filtering, 78
 finding, 140

selection, *continued*

- get_selection, 107
- highlighted, 5
- highlighting characters, 87
- moving, 91
- pending-delete, 51, 98
- piping to a command, 107
- primary, 85
- printing, 117
- problem, 98
- replacing, 92
- required for menu item, 20
- secondary, 96 *thru* 105
- stuck on secondary selections, 98

Services⇒ menu, in SunView menu, 116

Services⇒, in SunView menu, 24

Set Directory, in Text menu, 22, 128

Shell Tool, 170

Shell Tool, remote login to, 117

shells, 168

Shells⇒ menu, in SunView menu, 116, 168 *thru* 173

shelltool command, used in SunView, 168, 170

Shift key, 148

Shift Lines⇒, in Text menu, 142

Show Caret at Top, in Text menu, 43, 137

Show Clipboard, in Text menu, 18

Show Clipboard⇒, in Text menu, 134, 140

slider, 178

Split View, in Text menu, 135

standard input, 141

standard output, 141

starting a tool, 167

starting SunView, 1

stay-up menu, 6

stay-up menus, 51, 115

Stay_up default, 163

stdin, 141

stdout, 141

Stop function key, 98, 155

stop sign pointer, 145

stopping a secondary selection, 98

stopping an operation, 98

Store as New File, in Text menu, 21, 127

stty command, used in SunView, 160

Stuff, in Tty menu, 100, 134, 145

subframe, 76

.suntools file, 117

SunView

- Click_to_Type, 163
- emergency exit, 58
- exiting, 51, 58, 117
- font, 162
- Icon_gravity, 163
- starting, 1
- starting and exiting, 55 *thru* 58
- tools, 2, 23, 168

SunView category, in Defaults Editor, 161

sunview command, to start SunView, 55

.sunview file, 57, 82, 117

SunView menu, 23, 116 *thru* 120

SunView menu, *continued*

- adding icons, 120
- customizing, 118
- editors, 116
- Exit SunView, 58
- Lock Screen, 58
- printing, 24
- Save Layout, 57
- services, 116
- shells, 116
- tools, 116

swapping text, 105

swin command, used in SunView, 67

T

Tab character, 48

target pointer, 6

terminology, 183 *thru* 187

text

- copying, 89, 102, 103
- cutting, 89
- deleting, 14, 88, 89, 104
- editing, 87
- erasing, 12, 13, 88
- filtering, 78
- finding and replacing, 43
- font, 162
- handling in different windows, 106
- in panel, 80
- inserting, 17, 87
- long lines, 47
- moving, 19, 102, 103
- pasting, 87, 89
- quick editing, 100
- quoting commands, 154
- replacing, 44, 92, 102, 103
- saving, 21
- scrolling, 109
- searching, 43
- selecting, 14, 85
- swapping, 105
- wrapping, 47

Text category, in Defaults Editor, 161

Text Editor, 8, 173

- document, 9
- frame, 9

text item, 44

text item in panel, 43

Text menu, 10, 127 *thru* 144

- Cmd Modes⇒ menu, 144
- Display⇒ menu, 134 *thru* 138
- Edit⇒ menu, 130 *thru* 134
- Extras⇒ menu, 141 *thru* 142
- File⇒ menu, 127 *thru* 130
- Find⇒ menu, 138 *thru* 141

text window, 9, 10, 71, 76

- scratch window, 21
- .text_extras_menu file, 142

Tty/Text_wraparound_size, 164

textedit command, used in SunView, 168, 173

.textswrc file, 71

thumbing, 110

To End, in Find and Replace frame, 139

To End, in Find and Replace frame, 46

toggle, 8

Open and Close, 121

toggling, 148

tool

Clock, 182

Console, 3

Dbx (debug) Tool, 180

editors, 173

in SunView menu, 23

Mail Tool, 179

Performance Meter, 181

starting, 167

Tools⇒ pull-right menu, 116

tool names, 168

toolplaces command, used in SunView, 117

Tools⇒ menu, in SunView menu, 116

triple-clicking, 15, 95

Tty category, in Defaults Editor, 161

tty escape sequences, 69

Tty menu, 145

tty window, 68, 81

.ttyswrc error, 68

.ttyswrc file, 68

tutorial, 1 thru 51

Type2 keyboard, 61

Type2 keyboard labels, 65

Type3 keyboard, 60

Type3 keyboard labels, 64

Type4 keyboard, 60

Type4 keyboard labels, 63

typing all capital letters, 154

U

unconstrained

in Frame menu, 33

uncovering a frame, 30

underlining for secondary selection, 96

underscore (`_`) in text, 95

Undo All Edits, in Text menu, 133

Undo Last Edit, in Text menu, 23, 133

Undo operation, 94

Undo⇒, in Text menu, 133

undoing editing, 94

undoing scrolling, 110

undoing, warning, 23

UnZoom, in Frame menu, 37, 124

/usr/include/images directory, 120, 175

/usr/lib/.text_extras_menu file, 142

/usr/lib/defaults file, 158

/usr/lib/fonts/fixedwidthfonts directory, 162, 177

/usr/lib/rootmenu file, 118

V

Vertical_bar_placement default, 163

view, destroying, 137

view, splitting, 135

W

What Line Number?, in Text menu, 134

window, 75 thru 82

command, 78

edit, 9

emptying, 13

resizing, 123

scratch, 9, 77, 159

text, 9, 71, 76

text handling comparison, 106

tty, 68, 81

windows, copying between, 93

wiping to extend a selection, 17

word

defined, 95

selecting, 15

wrapping at, 138

word, erasing, 88

Wrap at Character, in Text menu, 48, 138

Wrap at Word, in Text menu, 49, 138

wrapping long lines, 138

wrapping text, 47

Z

Zoom, in Frame menu, 37, 124

SunView User's Guide: Quick Reference

Function Key Accelerators

Type4 Keyboard	Menu Action	Menu	Other Keyboards
Stop	Stop	—	L1
Props	"Props"	Frame	L3
Front	"Front", "Back"	Frame	L5
Open	"Open", "Close"	Frame	L7
Find	"Find⇒Find Selection⇒Forward", "Find⇒Find Clipboard⇒Forward"	Text	L9
Again	"Edit⇒Again"	Text	L2
Undo	"Edit⇒Undo⇒Undo Last Edit"	Text	L4
Copy	"Edit⇒Copy"	Text	L6
Paste	"Edit⇒Paste"	Text	L8
Cut	"Edit⇒Cut"	Text	L10
Shift-Front	"Back"	Frame	Shift-L5
Shift-Find	"Find⇒Find Selection⇒Backward", "Find⇒Find Clipboard⇒Backward"	Text	Shift-L9
Control-Find	"Find⇒Find and Replace"	Text	Control-L9

Mouse Button Accelerators

Frame Menu Equivalent	Mouse Action
<i>Open a frame:</i>	Point to the icon and click LEFT.
<i>Move a frame unconstrained:</i>	Point to a side of the frame border near the corner, hold down MIDDLE and drag the frame, release when the bounding box is the desired size.
<i>Move a frame constrained:</i>	Point to a side of the frame border near the middle, hold down MIDDLE and drag the frame, release at the desired position.
<i>Move an icon (always unconstrained):</i>	Point anywhere in the icon, hold down MIDDLE and drag the icon to the new location.
<i>Resize a frame unconstrained:</i>	Point to a side of the frame border near the corner, hold down Control -MIDDLE and resize the frame, release at the desired position.
<i>Resize a frame constrained:</i>	Point to a side of the frame border near the middle, hold down Control -MIDDLE and resize the frame, release at the desired position.
<i>Zoom a frame to full height:</i>	Click Control -LEFT on the frame border.
<i>UnZoom a zoomed frame:</i>	Click Control -LEFT on the frame border of a frame that has been zoomed.
<i>Bring frame to front:</i>	Click LEFT on the frame border.

Text Handling in Windows and Panels

<i>Description</i>	<i>Text Window</i>	<i>Command Window</i>	<i>Tty Window</i>	<i>Panel Text Item</i>
Insertion point	Caret	Caret	Block cursor	Caret
Menu	Text menu	Text menu with "Cmd Modes⇒"	Tty menu	1 or more chars
Select text	1 or more chars	1 or more chars	1 or more chars	Anywhere in line
Insert text	Anywhere	Anywhere in command line	At end of command line	At end of item
Erase text	Anywhere	Anywhere in command line	From end of command line	Anywhere in line
"Copy"	1 or more chars	1 or more chars	1 or more chars	1 or more chars
"Paste"	Anywhere	Anywhere in command line	To end of command line	Anywhere in line
"Cut"	Anywhere	Anywhere in command line	No	Anywhere in line
"Undo"	Anywhere	Anywhere in command line	No	No
Pending-delete selection	Anywhere	Anywhere in command line	No	Anywhere in line
Secondary selection — source	1 or more chars	1 or more chars	1 or more chars	1 or more chars
Secondary selection — destination	Anywhere	Anywhere in command line	End of command line	Anywhere in line

Menu Accelerators

<i>Menu Action</i>	<i>Keyboard Equivalent</i>
"File⇒Store as New File"	⌘-S
"File⇒Load File"	⌘-L
"File⇒Include File"	Esc ⌘-I
"File⇒Set Directory"	Shift-Esc
"File⇒Empty Document"	Esc ⌘-E
"Edit⇒Again"	⌘-A
"Edit⇒Undo"	⌘-U
"Edit⇒Copy"	⌘-C
"Edit⇒Paste"	⌘-V
"Edit⇒Cut"	⌘-X
"Edit⇒Copy then Paste"	⌘-P
"Display⇒Move Caret to Start"	Home (R7) Shift-Control-Return
"Display⇒Move Caret to End"	End (R13) Control-Return
"Find⇒Find and Replace"	Control-Find
"Find⇒Match Delimiter"	⌘-D
"Find⇒Replace >field< ⇒Next"	Control-Tab
"Find⇒Replace >field< ⇒Previous"	Shift-Control-Tab
"Find⇒Find Selection⇒Forward" or "Find⇒Find Clipboard⇒Forward"	⌘-F
"Find⇒Find Selection⇒Backward" or "Find⇒Find Clipboard⇒Backward"	Shift-⌘-F

Erase Accelerators

<i>Erase Action</i>	<i>Keyboard Equivalent</i>
Erase_Char_Backward	Delete
Erase_Char_Forward	Shift-Delete
Erase_Word_Backward	Control-W
Erase_Word_Forward	Shift-Control-W
Erase_Line_Backward	Control-U
Erase_Line_End	Shift-Control-U

Caret Accelerators

<i>Caret Action</i>	<i>Keyboard Equivalent</i>
Go_Char_Backward	R10 (left arrow) Control-B
Go_Char_Forward	Shift-Control-F R12 (right arrow) Control-F Shift-Control-B
Go_Word_Backward	Control-comma (,) Shift-Control-slash (/) Shift-Control-period (.)
Go_Word_End	Control-period (.)
Go_Word_Forward	Control-slash (/) Shift-Control-comma (,)
Go_Line_Backward	Control-A
Go_Line_End	Shift-Control-E Control-E
Go_Line_Forward	Shift-Control-A R11
Go_Column_Backward	Control-semicolon (;) R8 (up arrow) Control-P
Go_Column_Forward	Shift-Control-N R14 (down arrow) Control-N Shift-Control-P
Go_Document_Start	R7 (Home) Shift-Control-Return
Go_Document_End	R13 (End) Control-Return

Other Keyboard Accelerators

<i>Other Action</i>	<i>Keyboard Equivalent</i>
Caps_Lock	F1
Quote	◆-O

Quick Editing with Secondary Selections

<i>To do this...</i>	<i>Make primary selection...</i>	<i>Make secondary selection using...</i>
<p>DUPLICATE HERE: Copy secondary selection to caret.</p> <p>MOVE HERE: Move secondary selection to caret.</p> <p>DUPLICATE AND REPLACE HERE: Copy secondary selection to replace primary selection.</p> <p>MOVE AND REPLACE HERE: Move secondary selection to replace primary selection.</p>	<p>normal or none (sets caret)</p> <p>normal or none (sets caret)</p> <p>pending-delete</p> <p>pending-delete</p>	<p>Paste</p> <p>Control-Paste</p> <p>Paste</p> <p>Control-Paste</p>
<p>DUPLICATE THERE: Copy primary selection to left or right of secondary selection without moving caret.</p> <p>MOVE THERE: Move primary selection to left or right of secondary selection without moving caret.</p> <p>DUPLICATE AND REPLACE THERE: Copy primary selection to replace secondary selection without moving caret.</p> <p>MOVE AND REPLACE THERE: Move primary selection to replace secondary selection without moving caret.</p>	<p>normal</p> <p>pending-delete</p> <p>normal</p> <p>pending-delete</p>	<p>Copy</p> <p>Copy</p> <p>Control-Copy</p> <p>Control-Copy</p>
<p>DELETE THERE: Delete secondary selection without affecting primary selection or caret.</p>	<p>normal, none, or pending-delete</p>	<p>Cut</p>
<p>SWAP: Exchange locations of primary selection and secondary selection.</p>	<p>pending-delete</p>	<p>Control-Copy , then regular Paste .</p>

Notes

Notes

Notes

Notes

Systems for Open Computing™

Corporate Headquarters
Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043
415 960-1300
TLX 37-29639

**For U.S. Sales Office
locations call:**
800 821-4643
In CA: 800 821-4642

European Headquarters
Sun Microsystems Europe, Inc.
Bagshot Manor, Green Lane
Bagshot, Surrey GU19 5NL
England
0276 51440
TLX 859017

Australia: (02) 413 2666
Canada: 416 477-6745
France: (1) 40 94 80 00

Germany: (089) 95094-0
Hong Kong: 852 5-8651688
Italy: (39) 6056337
Japan: (03) 221-7021
Korea: 2-7802255
New Zealand: (04) 499 2344
Nordic Countries: +46 (0)8 7647810
PRC: 1-8315568
Singapore: 224 3388
Spain: (1) 2532003
Switzerland: (1) 8289555
The Netherlands: 033 501234

Taiwan: 2-7213257
UK: 0276 62111

**Europe, Middle East, and Africa,
call European Headquarters:**
0276 51440

**Elsewhere in the world,
call Corporate Headquarters:**
415 960-1300
Intercontinental Sales



Part Number: 800-3831-10
Revision A, of March 27, 1990