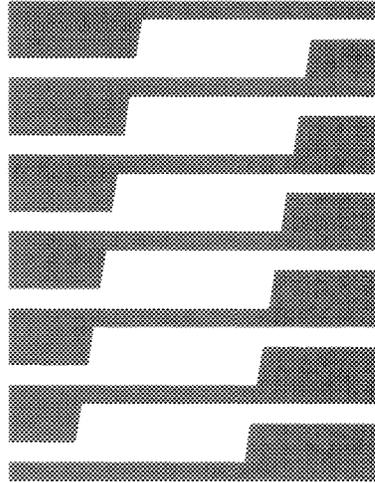# Interleaf

Technical Publishing Software

Reference Manual

Volume 1

Sun / Release 3.0

This manual was prepared using Interleaf Technical Publishing Software.

About the Reference Manuals

There are two *Reference Manuals*. Volume 1 is a comprehensive guide to the document preparation and desktop management aspects of the publishing software. Volume 2 describes the graphic capabilities of the software as well as printing procedures. Each volume contains a combined index to the two *Reference Manuals*.

In most cases, you need to read only the chapters that apply to a particular task. But we urge you to read the *Introduction* section in Volume 1 before you proceed to other chapters. It describes essential aspects of the Interleaf publishing software and gives information not repeated in later chapters.

Volume 1

The *Introduction* describes the basic user interface. *Fundamentals* presents and defines the objects — menus, icons, windows — with which you will work. *Basic Procedures* explains methods used throughout the software; they are presented here as the basis for more advanced procedures. *The Keyboard* lists the special commands available through function keys and key sequences.

Text Processing addresses the features that allow you to enter and format text. You will also find information about spell checking, hyphenation, search and replace, autonumbering and autoreferencing in this section.

The third section, *Page Makeup*, provides you with the means for determining and implementing an overall structure and design for your documents. Frames, which allow you to reserve space on a page for graphics or specially formatted text, are discussed in this section, as are the procedures for using the main document text editor in frames. The last chapter of this section, *Document Templates*, describes methods for creating templates from your standard page designs and for making the templates available to other users of the publishing software.

Managing Documents presents tools and in-depth discussions concerning document management issues. You will learn to access automatically created backup and checkpoint documents, to organize your documents on the desktop, to manage large documents, and to create indexes and tables of contents.

Upgrade Alerts contains brief descriptions of the new features of the Interleaf publishing software in Release 3.0. If you have used the Interleaf publishing software before, read this section first.

Volume 2

The six chapters of the *Diagramming* section explain the process of creating and modifying vector graphics, from simple objects like boxes to the most complicated technical illustrations.

The *Diagramming Extras* section contains three chapters. In *The Graphics Cabinet*, you will learn about the library of illustrations provided with the publishing software. All of them can be tailored to your requirements. The *Images* chapter describes the manipulations you can perform on raster graphics and instructions for editing images you can create with the screen capture feature of the publishing software. *Equations* describes the optional equation editor.

The *Charts* chapters offer detailed instructions for creating charts and information about modifying them to present your data to suit different purposes.

The *Printing* section describes the procedures for printing Interleaf documents from within an open document, from the desktop, or from the operating system.

In the last section of Volume 2, *Appendixes*, you will find illustrations of all the popup and pulldown menus, a discussion of the virtual terminal, and techniques for combining the features of the diagramming system to produce professional illustrations.

Graphic Aids in the Reference Manuals

Instead of numbering instruction steps, we use graphics that show you what you are going to be doing as you execute each instruction. For example, the usual instructions might look like this:

1. Move the mouse cursor.
2. Click the left mouse button.
3. Hold down the middle mouse button.
4. Release the button.
5. Type no.

In our instructions the graphics reinforce the meaning of the words and the type fonts provide emphasis.

   ⧉▲ Move the mouse cursor.

   ▮⬚⬚ Click the **left** mouse button.

   ⬚▮⬚ Hold down the **middle** mouse button.

   ⬚⬚⬚ Release the button.

   ⬚ Type **no**.

The following list tells you what the symbols used in instructions mean.

▮☐☐ Click the **left** mouse button.

☐▮☐ Click the **middle** mouse button.

☐☐▮ Click the **right** mouse button.

▮☐☐ Hold down the **left** mouse button.

☐▮☐ Hold down the **middle** mouse button.

☐☐▮ Hold down the **right** mouse button.

☐☐☐ Release whichever mouse button you are holding down.

▨ Represents the mouse.

≡ Indicates that you should move either the mouse — as in ≡▨—or the mouse cursor—as in ≡▲.

▣ Indicates that you need to use the keyboard, rather than the mouse, for this action.

↙ Means that you are being asked to do something that involves a series of actions you have already learned.

Sometimes we provide alternative steps. If the alternative is a single step, this is what you will see:

▣ At an operating system prompt, type **exit**.

or ▣ Hold down the **CTRL** key and type **d**.

If the alternative procedure consists of several steps, you will see the following:

↙ **Cut** the paragraph and **Paste** the *from* icon on your desktop.

or

↙ **Copy** the paragraph and **Paste** the *from* icon on your desktop.

↙ **Open** the *from* icon to make sure you really want to cut this paragraph.

↙ Then **Cut** the original paragraph.

Most of the symbols used throughout the publishing software are very easy to interpret because they represent objects you can see on your screen. For example, on your desktop you can see this icon ▯ which represents a *document*. Icons are used to make your choices simple and intuitive.

Related Publications

In the *Documentation* drawer of the *System* cabinet, you will find the *ReleaseNotes* folder. This folder contains *DocumentationSet*, a complete list of manuals that pertain to this product. If you don't find what you need in the *Reference Manuals*, refer to this list to find the most likely source for information.

Table of Contents

Introduction

Text Processing

Page Makeup

Managing Documents

Indexes and Tables of Contents 17

Upgrade Alerts

New and Modified Features 18

Index

Figure List

Methods

Chapter 1

Fundamentals

Fundamentals is an overview of the user interface for the Interleaf publishing software. Even if you have used other publishing software, skim this chapter to see how our software is different. You will find it essential to your understanding of the rest of the manual.

Chapter 2, *Basic Procedures*, provides a summary of the procedures you need to use the publishing software. It is not an exhaustive guide to all of the possible variations, but it will help you quickly locate the means for completing most elementary tasks. You will find Chapter 2 useful both as an introduction to these basic techniques and as a reference when you need to refresh your memory later. It is the only place in the *Reference Manual* that many of these procedures are covered in detail.

The Desktop

The **desktop** is the overall document management tool in the Interleaf publishing software. The organization and performance of the Interleaf desktop can be described using three building blocks — **icons, windows,** and **menus** — and one overall action called **selection**. On the desktop, you can create and cut documents, copy them, print them, and organize them in directories by project, due date, name, or any other criterion.

Figure 1-1 shows a desktop with an open window, icons in the window and elsewhere on the desktop, and a menu.
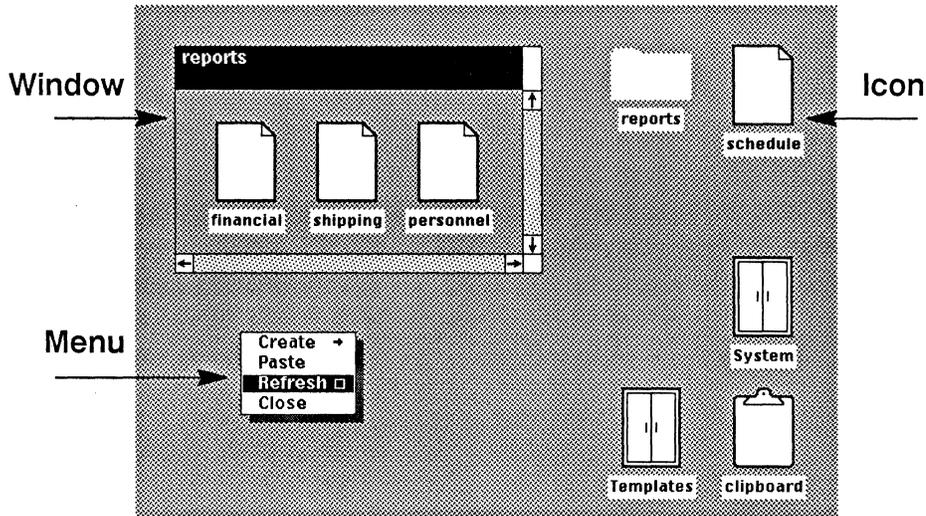
Figure 1-1. Interleaf desktop

Documents, directories, and devices, are represented on the desktop and in directory windows by **icons**. These icons can be **selected** and then acted upon by commands listed on **menus**. The menu you get depends upon the window you are in, whether or not you have selected an icon, and also where your cursor is in a window. The menu you see will always contain the commands that are appropriate for your current location and selection state, although you may have to fulfill additional conditions to use some of these commands.

Windows provide access to the contents of documents and directories. The desktop itself can be considered the master window of the publishing software. On the desktop, you can open up to 16 windows at once. You can edit any document whose window is open, cutting (or copying) and pasting text and graphics from one window to another. You can also cut, copy, and paste icons between directory windows.

Structured Document Editing

One of the most powerful features of the publishing software is its **structured document editing** capability. With this feature, you can streamline document design, alter layout easily, and ensure consistent style throughout a document. Figure 1-2 shows the elements that can make up a document. Arrows point from containers to objects they contain.

```
                          Document
                             │
                             ▼
                         Components
                        ╱          ╲
                      ╱              ╲
                    ▼                 ▼
        Text                          Frames
                    ╱      ╱  │  ╲      ╲
        Microdocuments     │  │  │  ╲      Groups
              ╱            ╱   │  │   ╲      ╲
            ▼            ╱     │  │    ╲      ▼
          Text         ╱      │  │     ╲   Freehand
          Strings     ▼       │  │      ▼   Drawings
                  Equations   │  │   Primitive
                  (optional)  ▼  │   Shapes
                            Images
```
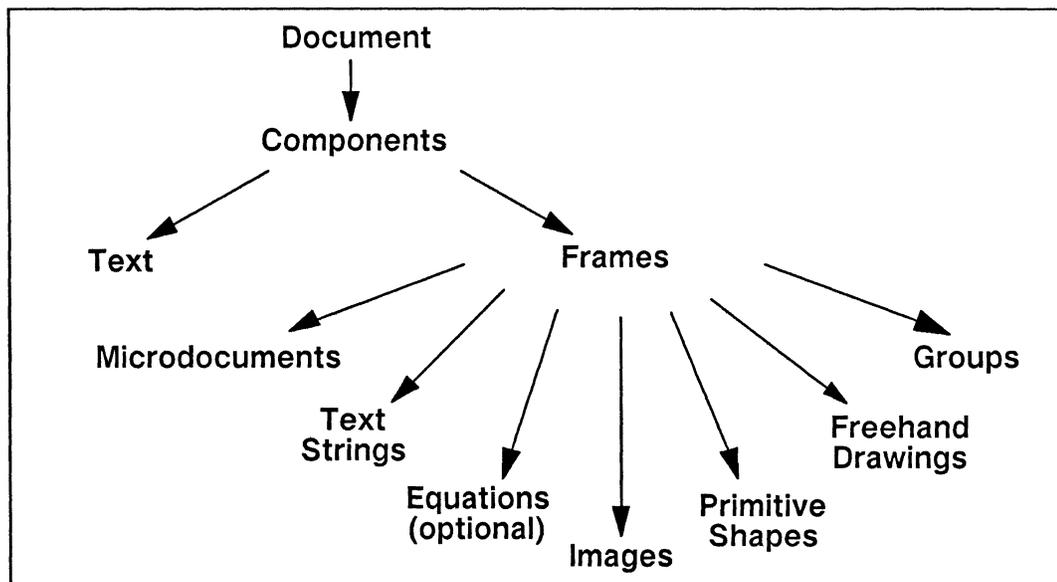
Figure 1-2. Structure of a document

Document Page Properties

Each document in the publishing software has its own **page properties** — height, width, margins, number of columns, and other properties — which you can customize to meet your requirements. These properties determine the overall boundaries within which other elements of a document are organized.

Components

Documents are made up of **components** in which you can put text, graphics, or both. Each component has associated **properties**; these include line length, margins, tabs, typefaces, type sizes, and relationships to other components and the page. From the default component definitions provided with the publishing software, you can create components with new names and properties. Components let you produce a **structured document** that you can use as a template for other documents.

If you have used word processing software, you may be accustomed to using carriage returns to create new paragraphs, to make lists, or to move from the body

of a letter to the closing. In the publishing software you would begin a new component instead.

Frames

Frames are the containers within which you create or paste graphics. Like components, frames have names and share properties such as height, width, and placement on the page. You can make all frames with a particular name share the same contents if you want. Frames are created inside components.

Microdocuments

You can create documents within frames to produce layouts that differ from the main document layout. These documents within frames are called **microdocuments**. They have their own page properties and components, and to edit them, you can use all of the editing tools available in regular documents. However, since they are diagramming objects, you also can *move* them, *size* them, and alter them with other diagramming commands. Microdocuments cannot contain frames.

Templates

When you combine custom page properties with custom-designed components, you can create **templates** for document types that you will create repeatedly. Although you can start with an empty document and a single component, the publishing software includes a set of document templates for you to copy and modify to meet your own specifications. You can access these templates through the same menu you use to create new icons. You can also copy them from the *Templates* directory we supply.

By adding frames to a template, you can make sure that illustrations in a document adhere to your size and placement standards. You can even use shared-content frames to specify standard rules, borders, or other graphics in your design.

The Book

To provide the ability to create large structured documents without losing the manageability of smaller documents, we provide a directory called the **book**. This directory allows you to have separate documents share the same page number stream as well as any other autonumber streams you create. The book also provides you with commands to create **indexes** and **tables of contents** for a series of documents.

Tokens

You can create **tokens** *in text* that produce automatically updated number streams, cross references to any token in that stream, and index entries that can

be automatically assembled in an index document. You can create tokens *in frames* to display the correct **page number** or **document name**.

The System Cabinet

In this release of the publishing software, we supply two objects on the desktop: the *clipboard*, which manages the cutting, copying, and pasting of objects, and a **System cabinet**. The *System* cabinet contains the following objects:

- a **Documentation drawer**, which contains on-line documentation and training aids for use in conjunction with the manuals we provide

- a **Graphics cabinet**, in which you will find clip art that you can copy and paste into your own documents

- a **Samples drawer**, in which you will find sample documents that show some of the text and graphics capabilities of the software

- a **Create cabinet**, which contains the icons that determine what names appear on the menu used to create icons

The default *Create* cabinet contains an empty copy of each of the following objects: document, folder, drawer, cabinet, book, and terminal. It also contains a **Templates cabinet**, which itself contains documents that already have layouts and in some cases, text and frames set up for you. Because these template documents are in the *Create* cabinet, you can produce memos, letters, and other standard items without having to structure them from scratch.

Your system administrator may have already customized the *System* cabinet. Although you cannot modify the *System* cabinet itself, you can copy things from it and place them on your desktop. You can also make your own desktop *Create* cabinet to override the *Create* cabinet in the *System* cabinet.

On-line Documentation and Training

In the *System* cabinet on your desktop, we include a drawer icon labeled *Documentation*. This drawer contains a folder labeled *Training*, which contains documents and other icons you can use in conjunction with the *Training Manual,* and a folder labeled *Reference*, which contains additional documentation added at the time of the software release. You should pay special attention to the *ReleaseNotes* folder, since it may contain vital information about changes or additions to the software.

Safety Measures

Throughout the software you will find built-in safety measures that allow you to recover from mistakes and accidents. Here are some examples.

- Text and graphics cut from *within* documents are not immediately disposed of but instead are placed in a directory called the *clipboard*. You can open the clipboard to recover anything you inadvertently cut.

- To prevent you from littering your system's memory storage unit with deletions, items newly cut or copied from *within* a document overwrite anything on the clipboard that came from within a document.

- When you cut *icons*, the icons are placed on the clipboard. Icons cannot be overwritten on the clipboard.

- Each of your documents is a **file** — that is, a collection of related information stored under one name. When you edit a document, the original, called the **document file**, is not modified until you decide to save your changes. At that point, the original becomes a **backup file**, and a new original is created. The software also periodically saves your changes in a separate **checkpoint file**. The **Revert** command allows you to revert to any existing version of a file.

- If you make changes in a document and then decide you are not sure you like the edited version, you can **Rename** the document. The renamed document will contain your changes and, as long as you have not saved the changes, the original document is closed in its unedited form.

- The **Undo** command will undo the last change you made to a diagram.

- If you make changes on a property sheet and then decide you would rather maintain the original properties, you can either reset the sheet to undo a whole series of changes or cancel the most recent ones only.

Tools

The Mouse

The user interface to the Interleaf publishing software is based on the **mouse**. The mouse is used to cut and paste text, to create diagrams and business charts, and to manipulate documents on the desktop. The only activity for which you cannot use the mouse is actual text entry. Many of the commands for manipulating text are duplicated on the keyboard, so that when you are typing a lot of text, you need not switch between mouse and keyboard constantly.

Because the Interleaf mouse has three buttons, we were able to give each of the buttons a strictly defined use that is the same in every part of the software.

The **left mouse button** is used to select an initial item. For example, you point the mouse cursor and click the left mouse button to select a position in text, an object in diagramming, an icon on the desktop, or a choice on a stickup menu. If you have previously selected any items, the left mouse button will deselect those items and select the new one.

The **right mouse button** is used to extend your selection. After you select a position in text, you can use the right mouse button to select a range of text on either side of that position. After you select an object in diagramming or an icon on the desktop, you can use the right mouse button to select additional objects or icons. The right mouse button is also used to modify your selection. For example, you can deselect an icon or diagramming object by pointing the cursor at it and clicking the right mouse button.

The **middle mouse button** is used to choose a command. When you hold down the middle mouse button, you will see a popup menu showing you the commands currently available. With the mouse button still held down, you move up or down the menu to the command you want to execute, and when the command is in reverse video, you release the button to execute the command.

Use the left button to select, the right button to extend or modify the selection, and the middle button to choose and execute a command.

Mouse Cursor Shapes

When you move the mouse, you move a **mouse cursor** on your display screen. As you work in the different parts of the software, you will notice that the mouse cursor changes its shape. A change in cursor shape indicates that a different set of commands is available to you. The following list shows the cursor shapes and the context in which they appear:

↑ On the desktop, in any open window except a document window, and on a stickup menu, the mouse cursor is an arrow.

⋏ In the body of a document window, it is a chevron.

◆ In the component bar to the left of the document, it is a diamond.

▲ In the header and scroll bars of a window, it is a triangle. You will also see a triangle when you hold down the left mouse button on the desktop or in a diagram.

☐ On a popup or pulldown menu, it is a small square.

⟍ In an open frame (in diagramming), it is a thin arrow.

In most of its forms, the mouse cursor blinks after it has been idle for a few seconds. The blinking makes it easier for you to find the cursor when you want to use it. The mouse cursor will not blink when you are using it on a pulldown, popup, or stickup menu because you are engaged in an operation.

Carets

Carets are place markers. As with cursors, carets change shape depending upon your place in the software. The fill pattern of carets can also change.

The list below shows the various text carets. On the left is the shape of the plain caret, and on the right is the shape of the caret when merged with the mouse cursor. If the cursor is on a document page, caret and cursor are merged when you use the keyboard. On the screen the differences are more obvious than they are here because the caret is larger.

△ △ are the carets in text when the font is roman;

▲ ▲ are the carets in text when the font is bold;

◸ ◸ are the carets in text when the font is italic;

◢ ◢ are the carets in text when the font is bold-italic.

There are two additional carets in the publishing software.

∧ is the caret in a property sheet or stickup menu;
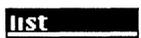
list▬ shows the component caret below the name of a component.

Icons

Figure 1-3 shows the icons available in the publishing software.

	Interleaf			External
Documents	document			file
Special Documents	dictionary			
Directories	folder	drawer	cabinet	directory
Special Directories	book	clipboard		scanner (optional)
Graphic Objects	diagram	image		plot
Printing File	Printerleaf			
Devices	terminal			
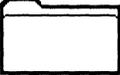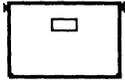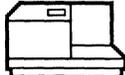
Figure 1-3. Icons available on the Interleaf desktop

A **document** is the object in which you can enter text and create diagrams. You can paste graphic icons and other documents into an open document.

The **dictionary** icon represents the *user dictionary* that you can use to make the spell checking and hyphenation features of the software recognize words not already in the main dictionary.

There are four kinds of standard **directories**: the desktop itself, folders, drawers, and cabinets. Directories can hold documents and other directories. If you want, you can reserve cabinets for *supplies* that you use in the preparation of documents. The *Graphics* and *Templates* cabinets are examples of this use of cabinets.

The **book** icon represents a special directory that assembles its contents into a unified document, allowing consistent page, figure and list numbering, as well as automatic table of contents and index creation for any number of documents placed in it.

The **clipboard** is a special directory that is used to transfer objects from one place to another. If, in *any* part of the software, you execute a **Cut** or **Copy** command, the object you cut or copy, whether it is a folder, a character, or a circle from a diagram, will be placed on the clipboard. When you **Paste**, the selected object(s) on the clipboard will be pasted. (Some objects cannot be pasted into others. For example, you cannot paste a folder into a diagram. Other objects can only be pasted into documents one at a time.) You can have only one clipboard icon on a desktop.

Graphic icons show a representation of the kind of graphics they contain. They cannot be opened: to see their contents you must paste them into a document. A **diagram** icon represents any graphic object cut from a **frame**. An **image** icon can represent a screen image or a scanned image.

The **Printerleaf** icon represents a file created from a document and converted into a form suitable for printing but not any other purpose.

The **terminal** icon represents a *virtual terminal*, which when open gives you a window from the desktop into UNIX. The UNIX window allows you to perform functions in UNIX without having to close your desktop first.

The **external** icons represent text files, directories, or plotter files brought into the publishing software from other sources. The **scanner** icon represents the directory in which images are stored after they are scanned (an optional feature requiring additional hardware). You can have only one scanner icon on a desktop.

Menus

Popup Menus

Most of the commands in the publishing software are found on **popup menus**. A popup menu is a list of commands that "pops up" when you hold down the middle mouse button. Figure 1-4 shows a popup menu.

```
Create  →
Paste
□Refresh
Close
```
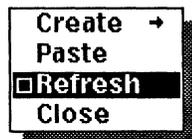
Figure 1-4. A popup menu

Popup menus contain the commands you can use in the current location under the current general conditions. If you change these conditions, you get a different menu. If you decide not to execute any of these commands, you can slide the cursor off the menu and release the mouse button.

Two Sets of Popups

In most parts of the publishing software, there are two possible popup menus: one if you have something selected, the other if you do not. For example, if you are working in the text of a document and no text is selected, when you hold down the middle mouse button you will see a popup showing commands like **Paste**. If you have a text region selected, the popup you see when you hold down the middle mouse button will show commands like **Cut** and **Copy**.

Pulldown Menus

If you hold down the middle mouse button when the mouse cursor is pointing into one of the header boxes in a document window, you will see a **pulldown menu** with commands appropriate to the label in that box (Figure 1-5).

```
          chapter1
  Class  Close  □   Bold  Italic
         Save    →
         Rename
         Revert  →
         Version
```
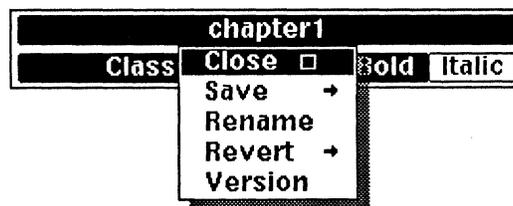
Figure 1-5. A pulldown menu

Pulldown menus are like popup menus except that you can access a particular pulldown menu in only one location.

Fundamentals

Submenus

Many of the commands on a popup or pulldown menu have an arrow after them. The arrow indicates that there is a **submenu** of options for that command. Figure 1-6 shows a submenu on one popup menu.

```
Create | Frame  →
Paste  | □Index
Fonts  | Auto # →
Center | Ref.
Misc   →
```
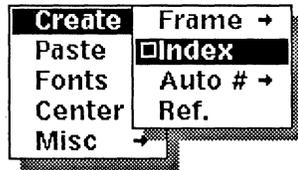
Figure 1-6. A submenu of a popup menu

You can see from the arrows that submenus may have further submenus. In general, the most commonly used commands are on the main menu. Less frequently used commands are on submenus, as are more complex commands.

Default Commands

When you hold down the middle mouse button, one command on the menu is displayed in reverse video even before you move the mouse. That command is the **default**. If you release the middle mouse button without waiting for the menu to be displayed, the default command on the popup will be executed.

In many parts of the software, the default command changes in response to the work you do. Figure 1-7 shows a default change on the Page pulldown menu.

```
The system default is              Next
    Properties                 becomes the
                                 default.

□Properties                    Properties
Next          but if you       □Next
Previous       choose          Previous
Go To          Next            Go To
First          ...             First
Last                           Last
Current                        Current
```
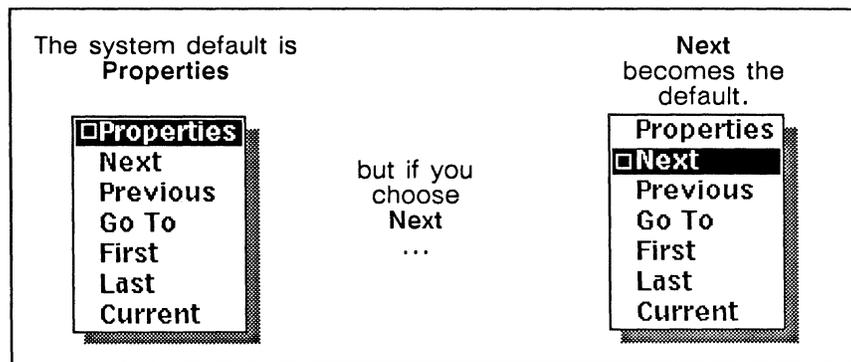
Figure 1-7. Dynamic defaults

We call this **dynamic defaulting**. The default you see when you first look at a menu is called the **system default**, and the command that becomes the default in response to your work is the **temporary default**. Most commands can become the temporary default, but it is easier for the user if certain commands do not become the default after being used. For example, if you create a new icon on the desktop, **Create** does not become the default.

The default on a popup or pulldown menu will often change back to the system default when you switch to a different part of the software. You can also reset

the system default yourself. Just hold down the middle mouse button, slide the mouse completely off the menu, so that no command is displayed in reverse video, and release the button.

So that you can take advantage of the defaulting feature, test the defaulting behavior of menus and submenus under a variety of conditions before you use them without looking at the command you are executing.

Windows

Figure 1-8 shows the parts of a directory window in the publishing software.
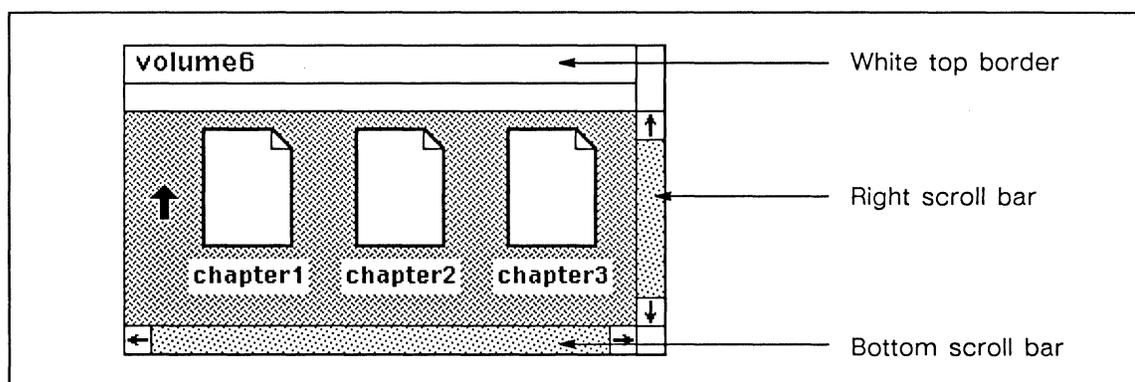
Figure 1-8. Active Directory window

In this software, a **window** provides a view of something — a page of a document, for example, or some of the icons in a folder. Windows can be resized and moved. The contents of a document or directory can be scrolled through its window. Windows can be stacked on top of each other, and you can choose one to bring to the front or move to the back of the stack. Property sheets, open documents, open directories, and open terminals are all shown in windows.

If you point the cursor at the top, right, or bottom border of a window and hold down the middle mouse button, you will see a popup menu that lists the operations you can perform on a window. In addition, the dotted **scroll bars** along the right and bottom edges of a window can be used to scroll the contents of most windows.

If you click the left mouse button while the mouse cursor is in the side scroll bar of the window, its contents are moved down, so you see whatever was above the original view. If you click the right mouse button, the contents of the window are moved up, and you see what was below the original view. In the bottom scroll bar, the left and right buttons move the contents of the window horizontally.

When you are working in a window, its top border area is white to show that it is the **active window**. All of the other windows are shown with a black top border. As you move your mouse cursor from one window into another, you will notice the top borders changing to show which is active.

Document Windows

Document windows have additional features not found in directory windows. A **component bar** on the left edge of the document displays the names of the components that contain the text and graphics visible in the window. This component bar also shows which, if any, of the components visible on the page are selected, or it may show the component caret. Figure 1-9 shows a document window.
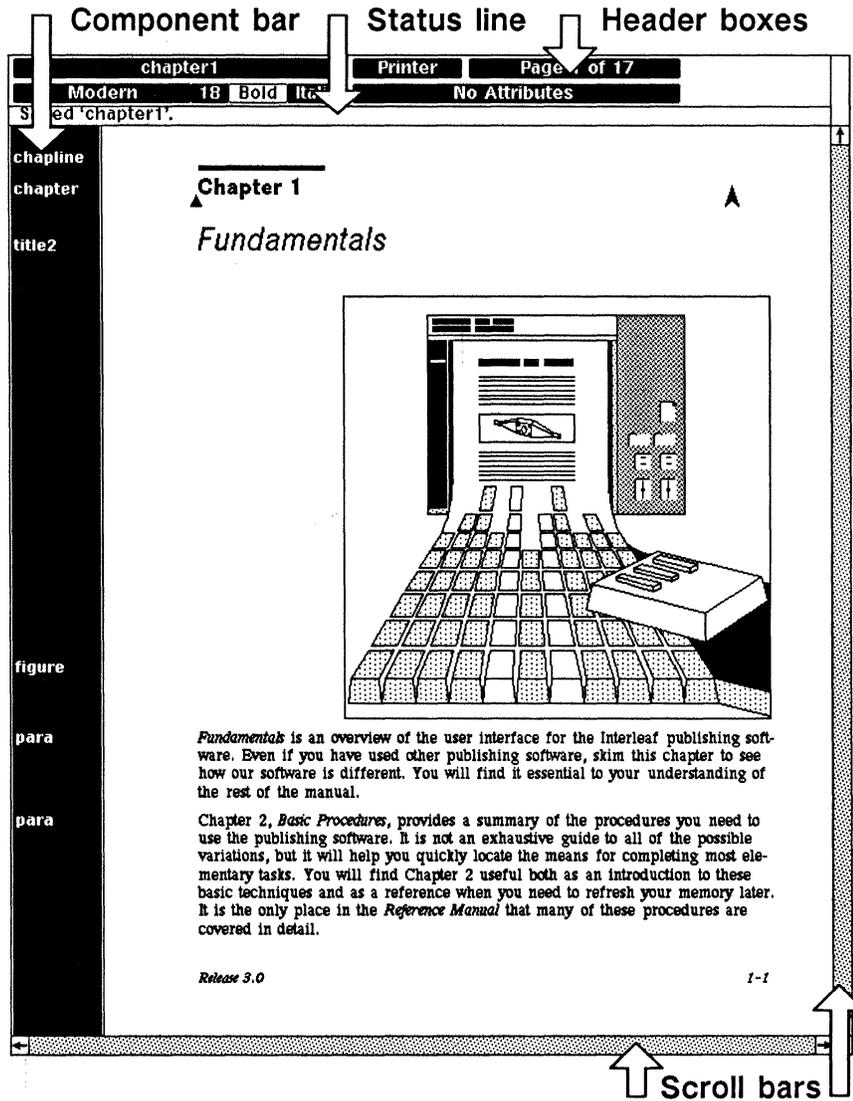
Figure 1-9. Document window

Document windows also have **header boxes** that show the name of the document, the font and other text attributes at the text caret, and the number of the page currently displayed. These boxes also provide access to pulldown menus with commands that print, scroll, save, close, change fonts, change printing and page properties, rename, and alter the version of a document. Figure 1-10 shows a detailed view of the header boxes for one document.

Name box ⊓   Printer box ⊓   Page box ⊓

| chapter1 | Printer | Page 1 of 17 |
| Modern   18 Bold Italic | No Attributes |

Font box ⊔   Attributes box ⊔
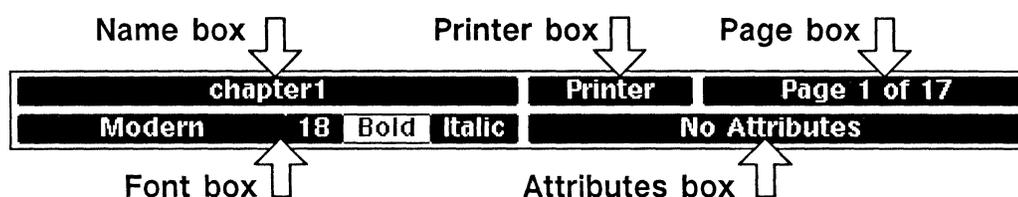
Figure 1-10. Header boxes of a document window

Frames

A **frame** is an object within a document that is reserved for diagramming activities, including the creation of microdocuments. A frame can be very small and a part of a line of text like any character, medium-sized and separated from its reference in text, or it can be so large that it is the only object on a page.

There are two major types of frames in the Interleaf publishing software.

- **Header/footer frames** occupy the top and bottom margins of every page in a document. Their properties, including the way they share their content throughout a document, are determined on the Page property sheet.

- **Text-anchored frames** are the frames you use for illustrations inside the text area of the page. You also can put one or more microdocuments in a text-anchored frame to create a special layout. By making a change on its property sheet, you can place a text-anchored frame at its anchor or at other locations on the page.

Text-Anchored Frames

Anchors ▣ are place markers that act as reference points for frames in text. They do not print, do not really take up room in text, and can be made invisible through a command available when no text is selected.

Text-anchored frames have three states: closed and unselected, closed and selected, and open. When a frame is open as in Figure 1-11, you have access to diagramming commands.
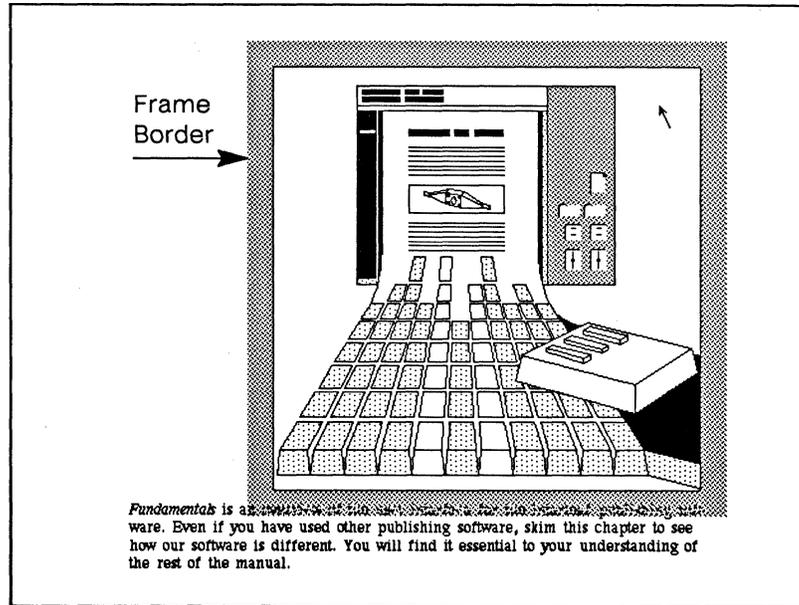
Figure 1-11. Open frame containing diagramming objects

Messages

When the Interleaf publishing software needs to communicate with you, it will
either display a message in the status line of a window or it will display a stickup
menu in the middle of the screen.

The Status Line

The **status line** is in the top border of the window, just below the header boxes
in a document window and just below the first line of a book or property sheet
window. It is used for error and status messages. For example, when the software
has finished saving a document, it displays a message on the status line
(Figure 1-12).
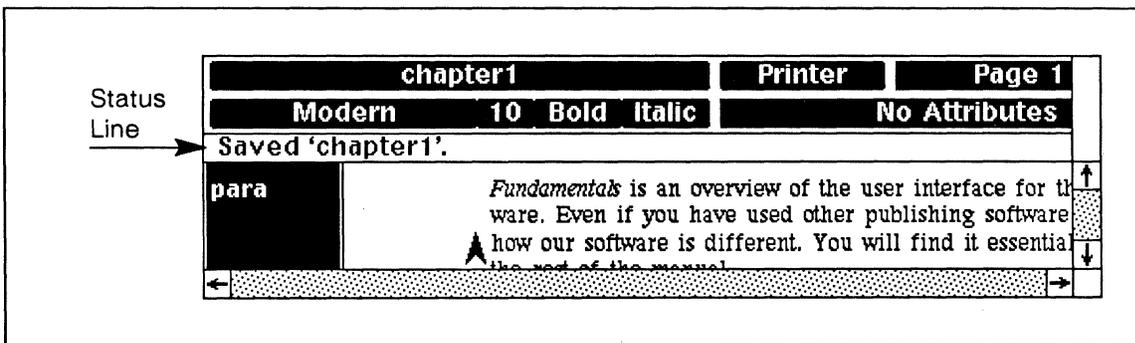
Figure 1-12. Sample message on the status line

Stickup Menus

Stickup menus are the software's way of asking for a response from you. For example, when you are moving to a specific page in your document, you type the page number in a stickup. Figure 1-13 shows one stickup, with its parts labeled. Unlike the other menus, you *must* respond to a stickup before you can do anything else, even if you just **Cancel** or **Continue**.
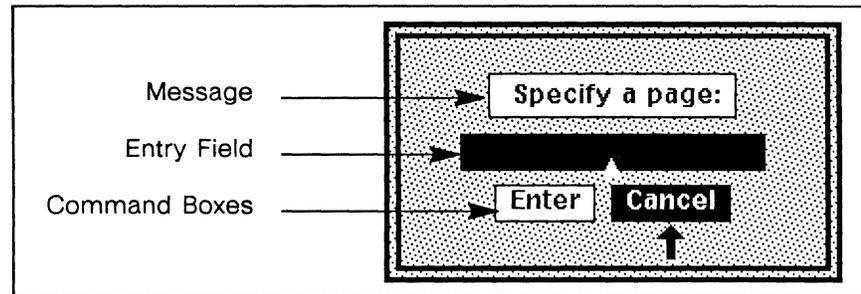
Figure 1-13. A stickup menu

In many cases, a stickup menu is a safety feature that says, in effect, *Are you sure you want to do that?*, before you take an action with serious consequences. Other times, it simply tells you that you *cannot* do what you just tried to do.

There are three possible parts to a stickup.

- At the top of the stickup there will be a message that reports information or requests a specific action.

- There may be a black rectangular field in which you can type a response. This field does not appear on all stickups.

- There is a set of command boxes. The mouse cursor points at one box, which will be shown in reverse video. To choose one of the other commands, slide the mouse left or right. The mouse cannot go farther than the last box on either end of the row. When the command you want to execute is displayed in reverse video, select it by clicking the left button or pressing **RETURN**.

The Interrupt Stickup

The Interrupt stickup menu is usually called up intentionally so that you can execute a command from it. You can call it up by holding the **CTRL** key and pressing **z** at any time during your activities on the desktop or inside any document or directory. (It is *not* available when the mouse cursor is in a virtual terminal window.) Certain software problems also trigger the appearance of the Interrupt stickup, which provides commands that can help you recover from these problems and save changes to your files.

Property Sheets

Many objects in the publishing software have **property sheets** that show the basic qualities of the object and allow you to change them. To look at the property sheet of an object, execute the **Props** or **Properties** command on a pulldown or popup menu. In most cases you will have to select the object before executing the command.

A property sheet appears in its own window. Figure 1-14 shows a Page property sheet.

Page Properties	**Normal**	Custom

Orientation	**Portrait** Landscape
Columns	1
Width	8.50 inches
Height	11 inches
Margins	
Top	1 inches
Bottom	1 inches
Left	1.40 inches
Right	1.10 inches
Starting Page #	Inherit 1
Page # Prefix	
Page # Style	**Arabic** Lowercase Roman Uppercase Roman
Headers/Footers	
Page Layout	Single Sided **Right 1st Page** Left 1st Page
Bleed	**No** Yes
Diff 1st Header	No **Yes**
Diff 1st Footer	**No** Yes

Figure 1-14. A property sheet

Multiple Property Sheets

Some property sheets are made up of several sheets. For example, the property sheet in Figure 1-14 has a *Normal* and a *Custom* sheet, accessible via the boxes you see next to the property sheet name in the window header. You can gain access to a sheet not currently shown by pointing the mouse cursor into its box at the top of the sheet and clicking the left button.

Chapter 2

Basic Procedures
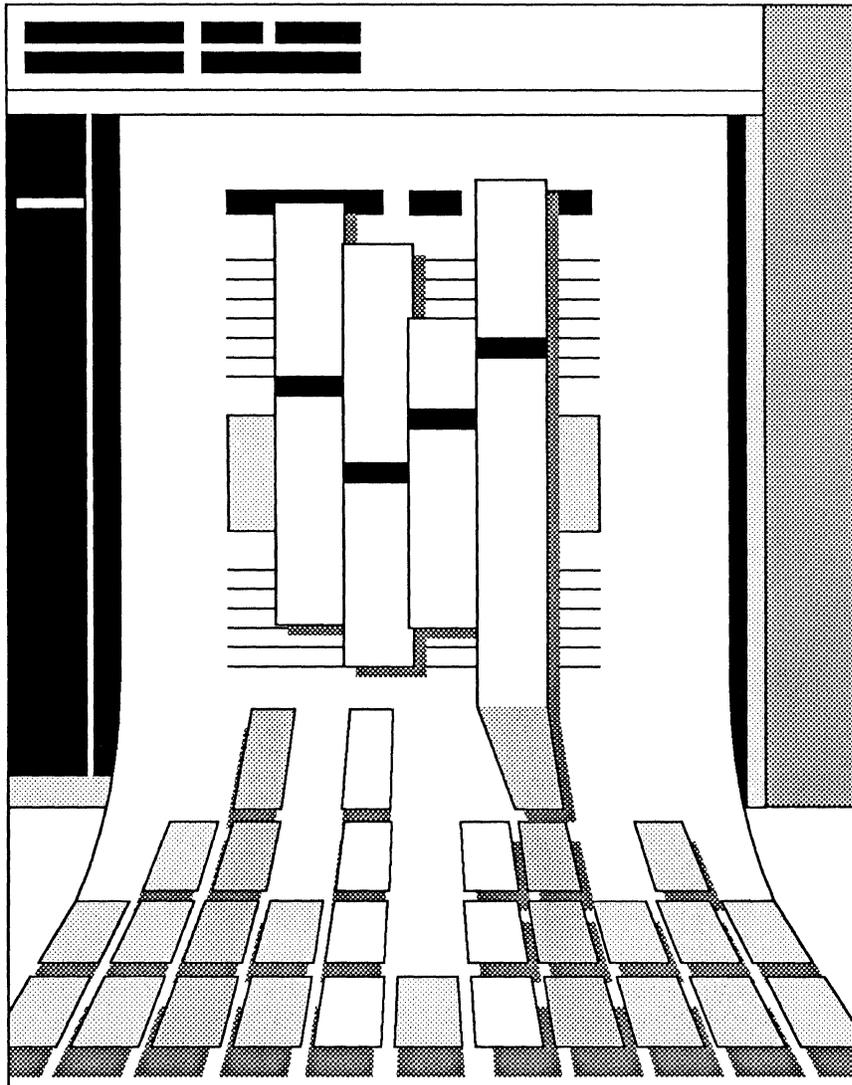
This chapter provides a summary of the basic procedures needed to use the publishing software. It is not an exhaustive guide to the many variations possible but will help you quickly locate the means for completing most tasks. The procedures described here are essential throughout the software and are used as the basis for more advanced procedures.

How to Open and Close a Desktop

The first time you log in to a workstation, the software creates and opens a desktop for you. After that, each time you log in with your name, *your* desktop opens.

To log in to a workstation and open your desktop:

▢ Hold down the CTRL key and press **c**.

▢ Type your login name followed by a RETURN.

▢ If you have a password, type your password and a RETURN.
If your desktop does not open, type ops *and a* RETURN.

To close your desktop and log out from the workstation:

▯▮ With the mouse cursor on the desktop itself and no icons in reverse video, hold down the **middle** button to see the Nothing Selected menu (Figure 2-1).

```
┌─────────────┐
│ Create   →  │
│ Paste       │
│ Refresh  □  │
│ Close       │
└─────────────┘
```
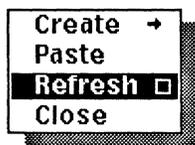
Figure 2-1. Desktop Nothing Selected popup menu

▯ Move the mouse cursor until **Close** is in reverse video.

▯▯▯ Release the button.
If you try to close your desktop while you have open documents with unsaved changes, a stickup menu appears stating that you cannot close the desktop without saving or closing those documents. Instructions for saving and closing documents are available on page 2-12.

If you execute the Close *command without exiting from operations you were performing in a terminal window, a stickup menu gives you the opportunity to cancel or confirm your decision to close the desktop.*

When you have no open documents with unsaved changes, and no active terminals on your desktop, a stickup menu appears asking you to confirm that you want to close the desktop (Figure 2-2).

```
┌─────────────────────────────────────────┐
│  ┌───────────────────────────────────┐  │
│  │ Confirm if you wish to exit the desktop. │  │
│  └───────────────────────────────────┘  │
│         ┌─────────┐ ┌────────┐          │
│         │ Confirm │ │ Cancel │          │
│         └─────────┘ └────────┘          │
│                    ↑                    │
└─────────────────────────────────────────┘
```
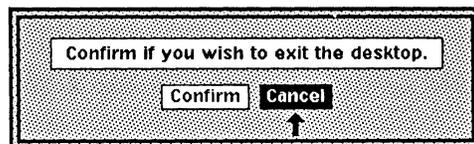
Figure 2-2. Close Desktop stickup

▯↑ Move the mouse cursor until **Confirm** is in reverse video.

▮▯▯ Click the **left** button.

How to Execute Commands on Menus

To call up a popup menu:

◻◼◻ Hold down the **middle** button.
You will see a menu similar to that in Figure 2-3a.
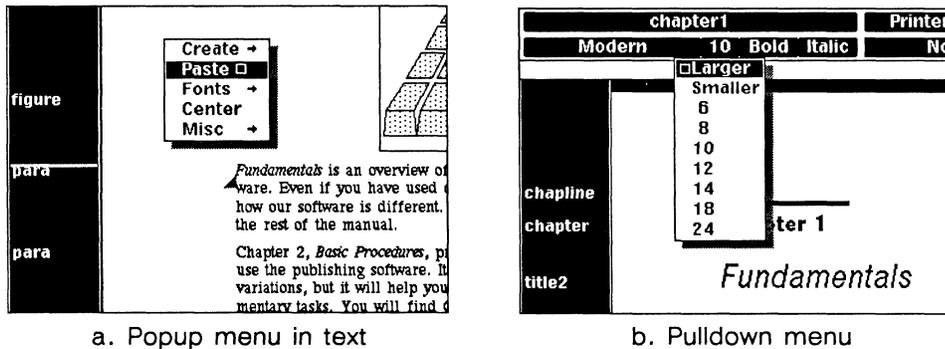
a. Popup menu in text   b. Pulldown menu

Figure 2-3. Popup and pulldown menus

To call up a pulldown menu:

▪▲ Move the mouse cursor into the document header region
and point into the box describing the menu you want to use.

◻◼◻ Hold down the **middle** button.
You will see a menu similar to that in Figure 2-3b.

To execute a command from a popup or pulldown menu:

▪◻ With the **middle** button still held down, move the mouse
cursor until the command you want is in reverse video.

◻◻◻ Release the button.
*The command that was in reverse video is executed. If you know you
want to execute the default command on a menu, you can click the
middle button without waiting for the menu to display.*

To make no selection from a menu:

▪◻ Move the mouse cursor off the menu *before* releasing the
middle button.
This also resets the system default if a previous action has changed it.

To call up a submenu:

⊡ While holding down the **middle** button, slide the mouse cursor to the command for which you want to see the submenu. Then slide the cursor in the direction of the arrow to display the submenu.

Submenu commands are executed in the same way as commands on the main menu. Figure 2-4 shows a menu with a submenu. The arrows on the right of the submenu indicate that the submenu has submenus.

Create	Frame →
Paste	▣ Index
Fonts	Auto # →
Center	Ref.
Misc →	
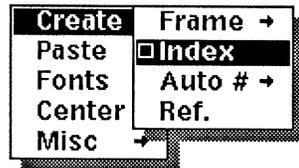
Figure 2-4. Menu with submenu

Stickup Menus

When a stickup menu appears, whether it was called up by you or appeared as a result of another action you took, you must respond to it before you can continue other activities in the publishing software.

To execute a choice on a stickup menu:

⊡ If you have information to enter, **type** it at the keyboard.

⊡↑ Move the mouse cursor until the command you want is in reverse video.

▮◻◻ Click the **left** button.

The Interrupt Stickup

What you use the Interrupt stickup for will depend upon whether its appearance is initiated by the software or initiated by you.

To call up the Interrupt stickup:

⊡ Hold down the CTRL key and type the letter z.

You will see a message and a series of choices (Figure 2-5).

User Interrupt Commands:
File \<f\> Exit \<e\> **Cancel \<C\>** Continue \<c\> Screen Image \<i\>
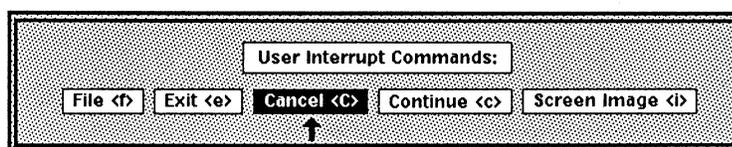
Figure 2-5. Interrupt stickup menu

You execute commands on the Interrupt stickup either by moving the mouse until the desired option is in reverse video and then clicking the left button, or by typing the letter next to your choice on the menu. If you use the keyboard, be aware of upper- and lower-case letters that produce different results.

The choices on the stickup menu vary according to the version of the software used and the options you have available. Figure 2-6 shows the standard choices.

Command	Effect
File	Allows you to create a crash file for any open document.
Exit	Removes you from the desktop, does not save any changes in open documents, kills processes in terminal windows. Always try to file before exiting if software failure triggered Interrupt stickup.
Cancel	Stops execution of certain commands and returns to the desktop. Cancels document opening, printing, search/find operations, and replace operations. Cancels multiple cut, copy, and props operations after the first icon completes the process.
Continue	Ends stickup display and continues any operations in progress.
Screen Image	Initiates process for screen image capture.

Figure 2-6. Interrupt Stickup commands

Crash Recovery

When the Interrupt stickup appears by itself, follow these steps:

- Try to **Continue**.

- If you cannot **Continue** and you have open documents, **File** those documents in crash files and then **Exit**.

- Report the problem to your organization's system administrator.

How to Select and Deselect

When an icon, a block of text, a frame, or a component is selected, it is displayed in reverse video. When a diagramming object is selected, it flashes. This section describes selection and deselection techniques for each type of object. Some popup menus have additional selection commands that offer alternatives to these methods.

For most objects in the software, you can select, deselect, or reverse (toggle) selection for a single item, a specified group of items, or all items in the context (directory window, frame, or component bar) you are in.

Icon and Diagramming Object Selection

The techniques for selecting icons in a window and diagramming objects in a frame are nearly identical, except that additional selection methods are available for diagramming objects.

Selection techniques do not extend across window or frame boundaries.

You are instructed to "point the mouse cursor at" something throughout this section. In these cases, you must have the tip of the mouse cursor pointing into the object *and* overlapping it, as shown in Figure 2-7.
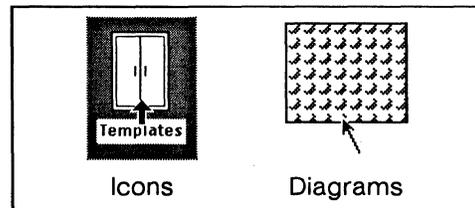
Figure 2-7. Pointing for selection

To select a single object (and deselect all others in that window or frame):

≡↑ Point the mouse cursor at the object you want to select.

▮□□ Click the **left** button.

To deselect (reverse selection of) a selected object:

≡↑ Point the mouse cursor at the *selected* object.

□□▮ Click the **right** button.

(Clicking the right button will select *an unselected object, but will not affect other objects.)*

To select any collection of objects in a window or frame:

⇗ Point the mouse cursor at the particular object you want to select.

▊⫿⫿ Click the **left** button to select it.

⫿⫿▊ To add each additional object to the selection, point at that object and then click the **right** button.

To select a group of adjacent objects in a window or frame:

↙ Imagine a box that surrounds the objects you want to select, and then position the mouse cursor at one corner of the imaginary box.

▊⫿⫿ Hold down the **left** button and *drag* the mouse until the selection box you are creating encloses all of the objects you want to select.

⫿⫿⫿ Release the button.

All objects within the box are selected (see Figure 2-8).

This drag selection...   ...produces this result
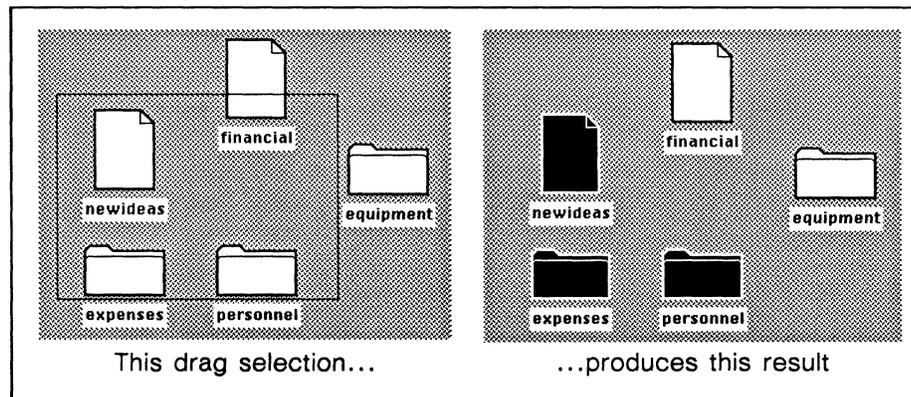
Figure 2-8. Drag selection of multiple icons

Notice that partially enclosed icons in Figure 2-8 are not selected, and that the icon name is not considered part of the icon.

To toggle the selection state of a group of adjacent objects:

↙ Imagine a box that surrounds the objects whose selection you want to reverse.

⫿⫿▊ Hold down the **right** button and drag the mouse until the box encloses all of the objects you want to reverse.

⫿⫿⫿ Release the button.

Window- or Frame-wide Selection Methods

In addition to single-object selection, several other selection techniques are effective across an entire window or frame. They affect all of the objects inside a particular active window or frame. When the mouse cursor is on the desktop itself, these techniques affect all icons on the desktop itself but no icons in windows.

To select everything in a window or frame:

≡↑   Place the mouse cursor anywhere inside the window or frame whose objects you want to select.

▮◻▮   Hold down the **left** button and click the **right** button.

◻◻◻   Release the **left** button.

To toggle the selection state of every object in a window or frame:

≡↑   Position the mouse cursor anywhere in the window or frame whose objects you want to toggle.

▮◻▮   Hold down the **right** button and click the **left** button.

◻◻◻   Release the **right** button.

Any selected objects are deselected and any unselected objects are selected.

To deselect every object in a window or frame:

↑   Make sure the mouse cursor is not pointing at an object.

▮◻◻   Click the **left** button.

or ↙   Hold down the **middle** button, move the mouse until **Deselect** is in reverse video, and then release the button.

Multiple selection allows you to do anything that individual selection would. You can *open, cut, copy, print,* open *property sheets, move,* and *deselect* as many icons as you select. You can *move, size, rotate,* and execute other diagramming commands for as many objects as you select.

Component Selection

You can select any number of components in a document. These components may be contiguous or they may be separated as in Figure 2-9.
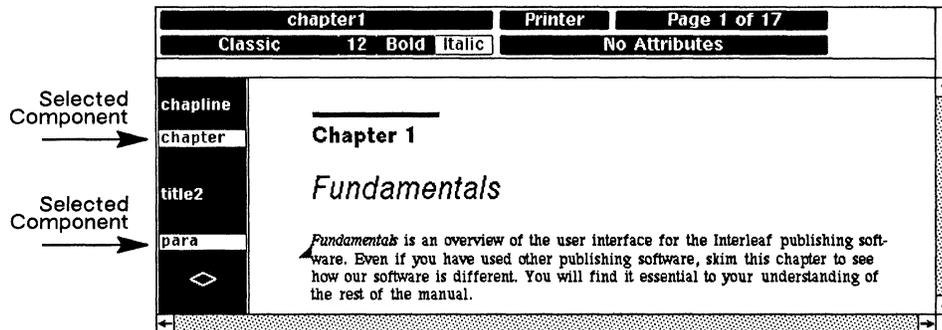
Figure 2-9. Selected components in a document

To select a single component:

⮕ Hold down the **left** button and drag the mouse cursor through the component bar, releasing the button when the name of the component you want to select is displayed in reverse video.

or ▓▯▯ Place the mouse cursor so that it overlaps the name of the component (Figure 2-10) you want to select and *click* the **left** button.

Figure 2-10. Cursor placement for component selection

To select two or more adjacent components:

🗸 Hold down the **left** button and drag the mouse cursor until the component caret is immediately above or immediately below the series of components you want to select.

🗸 Hold down the **right** button, and move the mouse cursor through the component bar across the names of the components you want selected.

▯▯▯ Release the button when the names of all components you want to select are highlighted.

After you release the button, you can move the mouse cursor in the component bar and use the right button to add additional components to your selection.

To select two or more scattered components:

    ✔ Select the name of the first component with the **left** button.

    With no buttons held down, move the cursor until it is above or below the name of another component you want to select.

    Place the mouse cursor over the name of that component, and click the **right** button. Repeat this step for each additional component you want to add to the selection.

To toggle the selection state of all components in a document:

    Hold down the **middle** button, move the mouse cursor until the **Select** command is in reverse video, and then move the cursor onto the **Select** submenu.

    Release the button when **Toggle** is in reverse video.

To deselect all components in a document:

    ✔ Hold down the **left** button and drag the mouse cursor until the component caret is visible. Then release the button.

    or With the **middle** button held down, move the mouse cursor until the **Deselect** command is in reverse video, and then release the button.

Additional selection methods, including selection of all components of a given name and selection to the beginning or end of a document, are possible using the Select submenus on the Component popup menus (Figure 2-11).

Create	All		note
Paste	**All Of**		paragraph
Select	Thru Page		□**section**
Find	To Front		subhead
Join	To End		title

Component Location menu
(sample component names shown)

Cut			
Copy			note
Change	All		□**paragraph**
Props	**All Of**		section
Select	Toggle		subhead
Deselect			title
Misc	→		

Component Selected menu
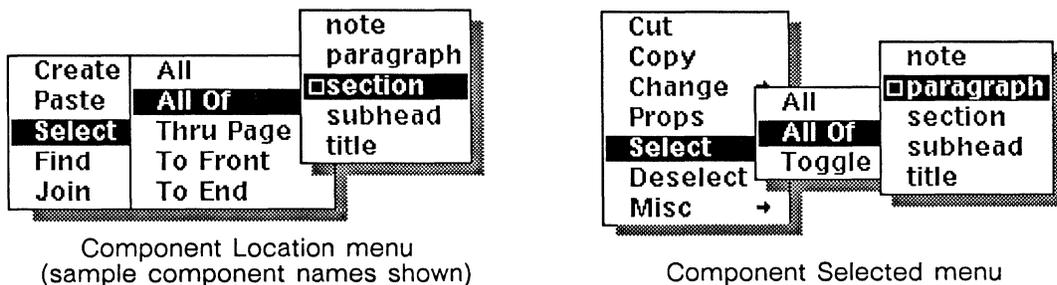
Figure 2-11. Component Select submenus

Text Selection

Text selection differs from other types of selection because all text selected at any one time in a document must be connected. However, your text selection can extend across several components, as Figure 2-12 shows.

chapter1	Printer	Page 9 of 17
Classic 12 Bold Italic		No Attributes

para	There are four kinds of standard directories: the desktop itself, folders, drawers, and cabinets. Directories can hold documents and other directories. If you want, you can reserve cabinets for *supplies* that you use in the preparation of documents. The *Graphics* and *Templates* cabinets are examples of this use of cabinets.	↑
para	The book icon represents a special directory that assembles its contents into a unified document, allowing consistent page numbering, page style, figure and list numbering, and indexing for any number of documents placed in it.	↓

Figure 2-12. Selected text (in reverse video)

To select text:

▮☐☐ Hold down the **left** button and move the text caret to where you want to begin your text selection.

☐☐☐ Release the button to position the text caret.

✔ Hold down the **right** button, move the mouse until the text you want to select is in reverse video.

If the cursor reaches a window boundary, the document will scroll.

☐☐☐ Release the button.

✔ If you want to add to the selection *in either direction,* move the mouse cursor beyond one end of the text that is in reverse video. Then hold down the **right** button again, and move the mouse until the additional text is in reverse video.

You can also select text by moving the cursor to where you want to begin the selection, clicking the left button to position the text caret there, moving the cursor to the end of the text block you want to select, and clicking the right button.

To deselect a portion of text by holding down the right button:

▲ Place the mouse cursor at the end of the selected block of text that is opposite the text caret.

☐☐▮ Hold down the **right** button and move the cursor to the character at the end of the text you want to stay selected.

☐☐☐ Release the button.

You can also deselect text by moving the cursor to where you want to end the selection of text and clicking the right button.

Documents and Directories

Documents and directories are represented by icons. When you open a document or directory, it is displayed in a window.

Opening a Document or Directory

To open a document or directory:

🖛 Select the icon for the document or directory you want to open.

▧ Hold down the **middle** button and move the mouse cursor until **Open** is in reverse video.

▧ Release the button to execute the command.

To cancel the opening of a document:

▣ Hold down the CTRL key and type **g**.

Saving and Closing a Document

To save a document and leave it open:

▴ Point the cursor into the Name box in the document header.

▧ With the **middle** button held down, move the cursor until **Save** is in reverse video (Figure 2-13).

```
┌──────────────────────────────────┐
│            chapter1               │
├──────────┬───────────┬───────────┤
│   Class  │  Close    │old  Italic│
│          ├───────────┤           │
│          │ Save □  → │           │
│          │ Rename    │           │
│          │ Revert  → │           │
│          │ Version   │           │
│          └───────────┘           │
```
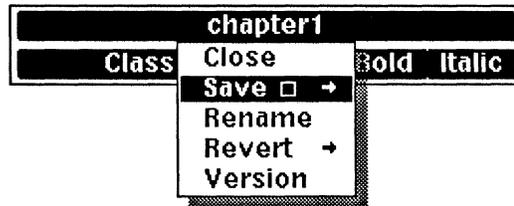
Figure 2-13. Save command on the Name pulldown

▧ Release the button.

To close a document:

▴ Point the cursor into the Name box in the document header.

▨ Because **Close** is always the default on the Name pulldown menu, just click the **middle** button.

If you have changed the document without saving it, a stickup menu appears offering the choice of saving the document, canceling the Close *command, or closing without saving the changes.*

To save and close a document:

    ✔ Follow the procedure for closing a document.

    ▊▢▢ When the stickup menu appears, move the mouse so that the black arrow points at the **Save** box and click the **left** button.

    As soon as the document has been saved, it will close automatically.

Closing a Directory

To close a directory:

    ⬆ Move the mouse cursor into the directory window.

    ✔ Deselect all icons in the window.

    ▢▊▢ Because **Close** is always the default command in a directory window when no icons are selected, just click the **middle** button.

Windows

The windows that contain documents, directories, and property sheets can be resized, moved, and stacked in a different order on the display screen. Up to 16 windows can be open on the desktop at one time. You change the placement and size of windows using the Window popup, shown in Figure 2-14.

```
Refresh
Resize ▯
Move
▪Resize ▯
Deselect
Front
Back
```

Figure 2-14. Window popup menu

The default on this menu varies. ▭Resize▯ is the default in a scroll bar. **Move** is the default in the white area of the document header. **Front** is the default on any window that is partially covered by another window.

Resizing a Window

The *maximum* size of a window is the size of the display screen itself. This does not limit the page size of a document. If you have a page larger than the screen, you can scroll in all four directions to see any portion of the page. You can also scroll from page to page in a document window.

The *minimum* size is approximately one inch square. This is the smallest window that will continue to give you access to the scroll bars, header, and the interior of the window. Without that access, you could not close, move, resize, or change the stacking order of the window.

If you resize a document or directory window, the next time you open the window, it will open at the new size.

To resize a window:

⊮ Move the mouse cursor either into a scroll bar or into the white area of a document header.

⊒☐ Choose ⌈Resize☐⌉ or ⌈Resize☐⌉ on the Window popup menu.

⊒▥ Move the mouse, and you will see an outline of the window. One corner of this outline is anchored. The opposite corner can be moved using the mouse.

▯▮▯ When the outline is the size you want the window to be, click the **middle** button to resize the window.

Moving a Window

You can move a window to anywhere on the desktop, but windows cannot be positioned so that any part of them runs beyond the screen boundaries.

To move a window:

⊮ Move the mouse cursor either into the white area of a document header or into a scroll bar.

⊒☐ Choose **Move** on the Window popup menu.

⊒▥ Move the mouse to move the outline of the window.

▯▮▯ When the outline is where you want the window to be, click the **middle** button to drop the window in its new spot.

Bringing a Window to the Front or Back

When you have more than one window open on the desktop, these windows have a default front-to-back stacking order. By default, each newly opened window is in front of all of the windows already open.

You can use the **Front** and **Back** commands to change the stacking order of any window on the desktop so that you can open several windows, stack them on top of each other, and uncover each one as you need it.

To change the front-to-back order of a window:

🖰 Move the mouse cursor either into a scroll bar or into the white area of a document header.

⠇☐ Choose **Front** to bring that window to the front of the other open windows or **Back** to move it to the back of the other windows.

The Scroll Bars

Each window has a *starting position* from which you can scroll in one direction only. You cannot scroll the contents down from the *starting position* using the right scroll bar, nor to the right from the *starting position* using the bottom scroll bar. Once you have scrolled up, you can then scroll down. Likewise, once you scroll to the left, you can then scroll to the right. In each case, you can scroll back to, but not past, the starting position.

To scroll the contents of a window up or down:

⠇▲ Move the mouse cursor into the right scroll bar.

⃝⃝▮ Click the **right** button to move the contents *up* through the window.

▮⃝⃝ Click the **left** button to move the contents *down* through the window.

In the right scroll bar, clicking the the right button moves the line at the cursor position to the top of the window. Clicking left button moves the line at the top of the window to the position of the cursor. The *lower* you are on the scroll bar, the more movement there will be at each click.

To scroll the contents of a window left or right:

⠇↑ Move the mouse cursor into the bottom scroll bar.

⃝⃝▮ Click the **right** button to move the contents to the *left* through the window.

▮⃝⃝ Click the **left** button to move the contents to the *right* through the window.

In the bottom scroll bar, clicking the right button moves whatever is at the cursor position to the left edge of the window. Clicking the left button moves whatever is at the left edge of the window to the position of the cursor. The closer you are to the *right* end of the scroll bar, the more movement there will be at each click.

Figure 2-15 shows the effect of various scroll bar actions. If you click the left button first in either scroll bar, no movement from the starting position occurs.
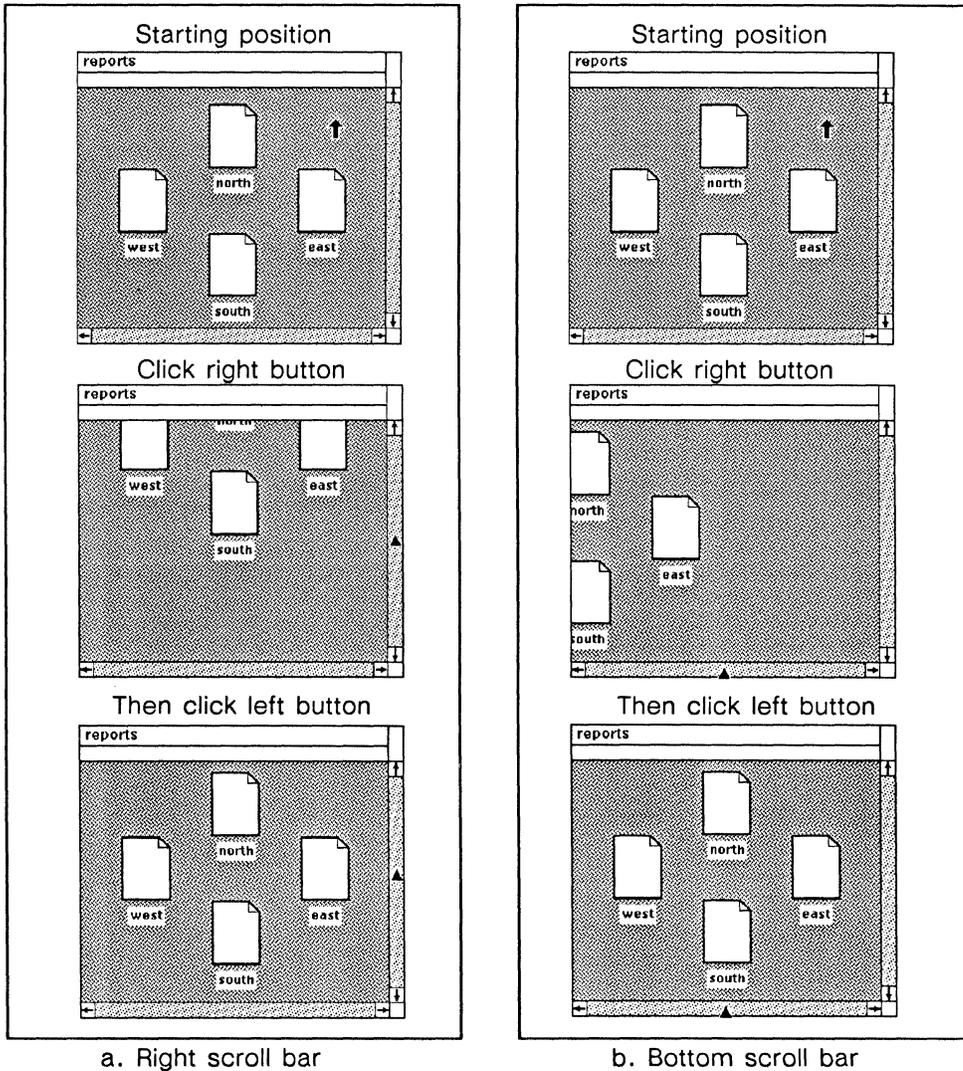
Figure 2-15. Scroll bar effects on a directory window

Remember that the triangular cursor ▲ indicates that you are in the scroll bar — any other cursor means that your clicking could produce a different result.

Window Refresh

The **Refresh** command on the the Window popup menu cause a redisplay of the area within the window. This is useful when system messages have printed over the page. A screen-wide **Refresh** command is available when the mouse cursor is on the desktop and nothing is selected.

Access *to Frames*

Text-anchored frames have three states: closed and unselected, closed and selected, and open. Figure 2-16 shows these three selection states.

There is evidence that the meaning of *widget* changed numerous times between the Dawn of Civilization and 1982 when Widget International was founded by Paul Reynolds. Shakespeare is said to have considered using *widget* in place of *bodkin* in Hamlet's "To be or not to be..." soliloquy. **a. unselected**	There is evidence that the meaning of *widget* changed numerous times between the Dawn of Civilization and 1982 when Widget International was founded by Paul Reynolds. Shakespeare is said to have considered using *widget* in place of *bodkin* in Hamlet's "To be or not to be..." soliloquy. **b. selected**	There is evidence that the meaning of *widget* changed numerous times between the Dawn of Civilization and 1982 when Widget International was founded by Paul Reynolds. Shakespeare is said to have considered using *widget* in place of *bodkin* in Hamlet's "To be or not to be..." soliloquy. **c. open**

Figure 2-16. Frame selection state

A closed, unselected frame is invisible — only its contents are visible. When you select a text-anchored frame, the frame, its contents, and its anchor are shown in reverse video (except for At Anchor frames, whose anchors are not visible). When you select a frame, its anchor is automatically selected and vice versa.

To select a text-anchored frame:

⇟ Move the mouse cursor into the frame area.

▤ Click the **left** button.

To open a frame:

✓ If the frame is a text-anchored frame, select it.

▤ With the cursor in the frame, click the **left** button.
You do not have to (and cannot) select a header/footer frame before clicking the button to open it.

To close a frame:

✓ Move the mouse cursor outside of the frame and click the **left** button.

or ✓ Execute the **Close** command.

To deselect a text-anchored frame:

✓ Execute the **Deselect** command.

The Clipboard

The *clipboard* is the temporary storage area for all objects — including documents, directories, components, text, frames, and diagramming objects — that you cut or copy.

- When you cut something, it is removed from its original location and placed on the clipboard.

- When you copy something, it remains in its original location and a copy of it is placed on the clipboard.

- When you cut or copy an icon, it remains on the clipboard until you paste or purge it. It is not overwritten by other icons you may cut or copy.

- Material cut or copied from *within* a document or terminal window appears on the clipboard with its icon name preceded by *from*. The contents of any *from* icon on the clipboard is subject to *overwriting* by the next cut or copy from within a document or terminal window.

- All material from one cut or copy operation inside a document — even if it consists of components scattered throughout a document — becomes one document on the clipboard.

- You can select and cut or copy several icons at once.

- Cut or copied text and graphics can be pasted on the desktop, within other directories, or within documents.

- Directories and devices cannot be pasted into documents.

- When you want to dispose of an object permanently, you can select and *purge* it from the clipboard.

- You can cut, copy, or paste with the clipboard closed. However, you may find the clipboard easier to understand if you open it the first few times you use it.

How to Cut and Copy

To copy an object:

- ✔ Select the object (icon, component, text region, frame, or diagramming object).

- ✔ Execute the **Copy** command. Use the default choice (**Normal**) if you are copying an icon.

  An icon representing the copy appears on the clipboard. The copy on the clipboard appears in reverse video to show that it is still selected. The original version of the object is no longer selected.

To cut an object:

✔ Select an object (icon, component, text region, frame, or diagramming object).

✔ Execute the **Cut** command.

The object disappears from its original position and an icon representing it appears on the clipboard. The icon appears in reverse video to show that it is still selected.

Cutting and Copying from Documents

You produce different icons depending upon what you cut or copy and how you execute the action.

- If you select and cut or copy a diagramming object from an *open* frame, that object appears on the clipboard as a *diagram* icon.

- If you copy text from the terminal window, it appears as an *external document* icon.

- In all other cases — even if you cut an entire frame but no other text — you see a *document* icon on the clipboard.

The 'from' Icons

Portions of documents, diagramming objects, and text copied from a virtual terminal are labeled with the *from* prefix on the clipboard. Figure 2-17 shows how different objects appear on the clipboard.
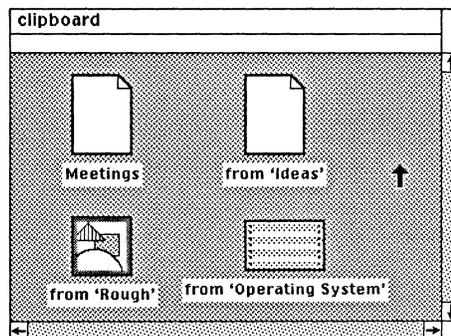
Figure 2-17. Icons on the clipboard

If you want to keep something you have cut from a document, *immediately* paste it onto the desktop, or into a directory or document. The desktop and other directories can maintain multiple *from* icons with the same name. The clipboard can safely store icons with the same name, as long as they are not *from* icons.

There normally can be only one *from 'document'* icon, one *from 'diagram'* icon, and one *from 'terminal'* icon on the clipboard at any time. Each succeeding cut or copy overwrites the contents of the *from* icon of its particular type already on the clipboard. This prevents your system from being filled with useless deletions.

You may have two *from 'document'* (or *from 'diagram'*) icons on your clipboard if you cut from inside a document, paste the *from 'document'* icon onto the desktop, cut from another document, and finally cut the first *from 'document'* icon back to the clipboard. *You should assume that the contents of both of these icons are subject to overwriting.*

Pasting from the Clipboard

Whatever is *selected* on the clipboard is what gets pasted. Immediately after you cut or copy anything, one or more icons representing the cut or copied object(s) appear on the clipboard in a selected state; *all other icons on the clipboard are deselected* after the new **Cut** or **Copy**. To paste any icon from the clipboard, simply select it, deselect all other icons, and follow the **Paste** procedure.

You can select and paste several items at once from the clipboard, except when you are pasting into a document. If you select *several* icons on the clipboard and attempt to paste them into a document, only one will be pasted — the others will remain on the clipboard in a selected state. You can repeat the **Paste** command until all of the selected icons are pasted.

To paste an object:

➤ Make sure the object or objects you want to paste, and only those objects, are selected on the clipboard.

➤ Move the mouse cursor to where you want the object(s) pasted and make sure nothing is selected there.

If you want to paste a diagramming object inside an existing frame, be sure the frame is open.

➤ Execute the **Paste** command.

If you paste an object as *text* or *components*, it is placed at the position of the caret in selected state. To deselect the pasted material, click the **left** button.

An object pasted as an *icon* or a *diagramming object* is placed in move state.

To finish pasting an icon or diagramming object:

≡▥ Move the mouse to place the object where you want it.

▊▢▢ If you plan to work with an icon or diagramming object immediately or want to select additional items, click the **left** or **right** button to drop the item and leave it selected.

or ▢▊▢ To deselect an icon or diagramming object, click the **middle** button.

Component and Page Information

In general, when you cut or copy from inside a document, you maintain the maximum amount of information available for whatever you cut or copy. For example, if you cut text from a document named *chapter1*, the document *from 'chapter1'* on the clipboard will contain not only the text itself but the properties of the component(s) the text was in and the page properties for *chapter1* as well. Because this information is preserved, you can paste text as text, as a component, or as a complete document.

All master definitions in a document are also brought to the clipboard whenever the document (or text or components from it) is cut or copied. This makes available any component, frame, or autonumber stream for which a master existed in the original document, even when there is no occurrence of the object the master defines.

When you paste documents or components as text, the software warns you that you will lose some level of organization. In these cases, a stickup menu (Figure 2-18) gives you the option of confirming or canceling the **Paste** operation.


```
Pasting the selection as text will destroy the
component information. Please confirm.

     Confirm   Cancel
              ↑
```
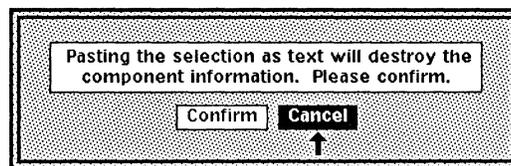
Figure 2-18. Loss of information stickup

If you cut or copy a *diagramming object* from an open frame, *no* frame, component, or page information is retained. If you cut or copy document *a* and paste it into the component bar of document *b*, the page properties of document *a* are lost.

Purging Objects

Icons that you cut or copy onto the clipboard stay there until you either paste them somewhere or purge them completely from your desktop. Objects on the clipboard use valuable storage space on your disk. Therefore, it is usually a good idea to clear the clipboard at least once a day.

When you **Purge** an object, there is no way to recover it.
Purge only those objects that you are sure you have no use
for. Be very careful with multiple icon selection so that you
do not purge by accident objects that you want to keep.

To purge objects from the clipboard:

✔ Select and **Open** the clipboard.

✔ Select the icon for the item to be purged.

✔ Call up the Clipboard Icon Selected popup menu
(Figure 2-19) and execute **Purge**.

```
┌──────────────┐
│ Purge        │
│ ▌Close  □    │
│ Deselect     │
│ Move         │
└──────────────┘
```
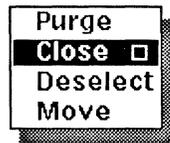
Figure 2-19. Clipboard Icon Selected popup menu

⇑ On the Purge stickup menu, you will be asked to confirm
that you really do want to purge the icon. Move the mouse
cursor to **Confirm**, and click the **left** button.

Keep in mind that you might not be able to purge or paste certain icons from the
clipboard because of their *read/write permissions*. If this happens, see your system
administrator.

How to Print Documents

You can print one or more documents or the content of one or more directories without opening these icons. You also can print open documents. Both techniques allow you to use different printers, print selected pages, and print multiple copies of a document, options discussed in the chapter, *Printing*. Desktop printing takes longer than open document printing because the software assigns it lower priority.

To print documents from the desktop:

✓ Select the icon(s) for object(s) you want to print.

✓ On the Icon Selected popup menu (Figure 2-20), slide onto the **Print** submenu and release the mouse button when the name of the printer you want is in reverse video.

This will print one copy of the object(s) selected. For other options, slide onto the Printer submenus before releasing the button.
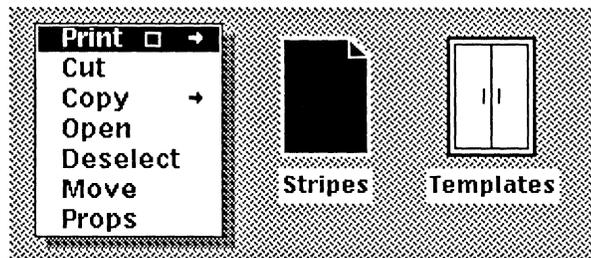
Figure 2-20. Icon Selected menu with desktop print command

To print from an open document:

✓ Point the cursor into the Printer box in the document header (Figure 2-21).

Figure 2-21. Cursor placement for Printer menu access

✓ If you want to print one copy of the whole document on your default printer, just click the **middle** button.

For other options, use the Print submenus.

Defaults

When you print from an open document to a printer other than the default, that printer becomes the temporary default until you select another printer or close the document. There is no default for desktop printing — *if you do not specify a printer, no printing takes place*. Both printing methods retain single-copy printing as the default, even if you have selected other options.

How to Change Properties

Figure 2-22 lists the property sheets available in the Interleaf publishing software and the type of properties they change.

Sheet Name	Properties Controlled
Object (Icon)	Name, Permissions, Icon Type, Pathname (links only)
Component	Name, Margins, Indents, Line Spacing, Alignment, Font, Hyphenation, Tabs, Widow and Orphan Control, New Page, Component Breaks, Multicolumn Straddle, TOC Pickup, Auto Number
Frame	Name, Size, Placement, Alignment, Shared Contents
Auto Number Stream	Stream Name, Number of Levels, Symbol Type, Prefix, Suffix, Starting Number
Auto Number Token	Tag Name, Stream Type, Level
Reference	Auto Number Tag to Reference, Page # or Auto Number pickup
Page	Orientation, No. of Columns, Size, Margins, Starting Page #, Page # Prefix, Page # Style, Header/Footer Style, Hyphenation, Gutter Width, Rev Bar Placement, Column Balancing, Vertical Justification
Printer	Header Page on/off, Double Sided on/off, Manual Feed on/off Print Rev Bars on/off, Print Strikethrough on/off, Print Underlines on/off, Final Output Device, Default Printer
Chart	Data, Style, Custom Design
Index	Entry for Index, Reference Font, Cross Reference, Range of Text for Entry, Index Document Name, Sort Strings
Equation (option)	Equation contents
Image (option)	Image Content, Framing, Contrast

Figure 2-22. Property sheets in the publishing software

To open a property sheet:

✔ Select the item whose properties you wish to see or change. For the Page or Printer properties sheet, point your mouse cursor into the appropriate header box.

✔ Execute the **Props** or **Properties** command.

Some property sheets are made up of several sheets. For example, there are four Component property sheets, all accessible through the **Props** command on the Component Selected menu. When multiple sheets are available, you will see a series of boxes next to the name of the property sheet, as in Figure 2-23. The box in reverse video indicates the sheet currently displayed.

Figure 2-23. Sheet choice boxes in property sheet header

To switch property sheets when multiple sheets are available:

↙ Point the mouse cursor into one of the unselected boxes in the property sheet header and click the left button.

Editing a Property Sheet

There are three different ways to change information on a property sheet.

- Typing information into or editing a **field**.
- Turning a **toggle box** on or off.
- Selecting a property in a **list box**.

Figure 2-24 shows a property sheet that contains each of these property types.
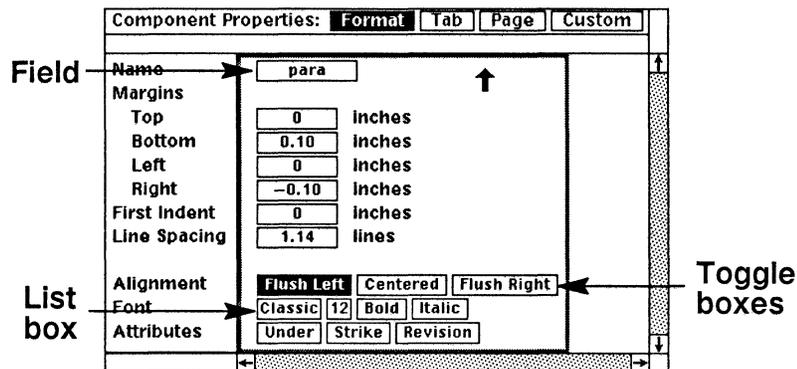
Figure 2-24. Component property sheet

To enter numbers or text in a field on a property sheet:

↑ Point the mouse cursor into the box for the field in which you want to make an entry.

▣▢▢ Click the **left** button to position the caret in the box.

▣ Type the text you want in the box.

Anything in the box when you start typing is automatically erased.

▣ Press TAB to confirm what you have typed and to move the caret to the next box in which you can type.

or ▣ Press RETURN to confirm what you have typed without moving the caret.

You can edit the contents of one of these boxes without starting from scratch. If you first use Control key sequences or the arrow keys to move the caret within a box, you can then insert or delete characters. You must confirm edited property sheet entries with TAB or RETURN.

- CTRL **f** or the right arrow key ➤ moves the caret forward one character.

- CTRL **b** or the left arrow key ◀ moves the caret backward one character.

- CTRL **d** deletes the character to the right of the caret.

- DEL deletes the character to the left of the caret.

- CTRL **k** deletes all contents in a box.

- CTRL **e** moves the caret to the end of the text in the field.

- CTRL **a** moves the caret to the beginning.

To change a property in a toggle box:

↑ Point the mouse cursor into the box you want to turn on, and click the **left** button to select it.

The active property is displayed in reverse video, and the deactivated property is displayed in normal video. For example, the Flush Left Alignment is turned on in Figure 2-24 and the other alignments are turned off.

To change a property in a list box:

↑ Point the mouse cursor into the box you want to turn on, and click the **left** button to move one way through the list or click the **right** button to move the other way.

Applying Property Changes

Once you make changes on a property sheet, you must **Apply** those changes to make them take effect. You apply changes differently depending upon whether you want to them to affect a single object or several objects of the same name. On most property sheets, you can apply the properties to a single object only.

To apply property changes to a single object:

✔ Make any desired changes to the property sheet.

▯▮▯ Click the **middle** button.

To close a property sheet after applying changes (or without changes):

◻▊◻ Click the **middle** button.

To cancel property changes before applying:

◻▊◻ Hold down the **middle** button and execute **Cancel**.

To cancel property changes after applying:

◻▊◻ Hold down the **middle** button and execute **Reset**.

◻▊◻ Click the **middle** button to **Apply** the reset properties.

Component and Frame Properties

Component property changes can be applied to one component or to all components of the same name within a document. In the same way, frame property changes can be applied to one frame or to all frames with the same name.

Master Definitions

Named components and frames are based on **master definitions** stored for the document. A new master definition is created whenever you enter a new name in the *Name* box of a Component or Frame property sheet and then apply that change. Master definitions are also updated when you execute a **Global Apply** or **Unify** command. Once you create a master definition in a document, you need not keep a copy of the component or frame it defines.

To apply property changes to all components or frames with the same name:

◻▊◻ Hold down the **middle** button and select [Global Apply →]
[Confirm] .
If you decide not to apply these changes, select **Cancel**.

To cancel global property changes after applying:

◻▊◻ Hold down the **middle** button and select **Reset**.

◻▊◻ Hold down the **middle** button and select [Global Apply →]
[Confirm] .

To apply all of the properties of one component or frame to all components or frames of that name:

◻▊◻ Before changing any properties or after you have applied your last changes, hold down the **middle** button and execute [Unify →] [Confirm] on the Close Property Sheet menu.

Both **Global Apply** and **Unify** should be used with extra care because of their ability to make changes throughout a document. If a component property change eliminates a distinction between the default text and a special font within it, that distinction cannot be restored with the **Reset** command. Make sure you have considered the effect a property change will have on all of the components it will affect.

Error Handling in Property Sheets

If you enter an inappropriate value in a property sheet field, an error message will appear in the status line of the property sheet. This may happen either immediately after you enter the value or after you **Apply** it. For example, if you enter a negative Top Margin in a Component property sheet, when you apply that change the error message *Top margin cannot be less than 0 inches.* is displayed in the status line.

- If the error message appears immediately upon entry, you need not take any additional action.
- If the error message appears after you **Apply**, you must either enter and **Apply** a new value or **Cancel** the inappropriate value.

In either case, you should examine the error message to see what was unacceptable about the value you entered.

How to Create New Objects

There are **Create** commands for icons, components, diagramming objects, frames, index tokens, autonumber streams, and autoreferences. Index tokens, autonumber streams, and autoreferences are discussed elsewhere in this manual.

Creating Icons

The Create submenus on the desktop or in a directory window allow you to create new generic icons — *document, folder,* and so on. You can redefine the properties of these icons after you create them.

Using the Templates submenu on these menus, you can create a copy of a template, including not just its properties but its content as well. In addition, your system administrator may add other items to the Create submenus.

To create a new document, directory, device, or template:

▤↑ Place the mouse cursor in the directory window in which you want the new icon created.

↙ Deselect any selected icons.

▯▮▯ Hold down the **middle** button.

▤▯ Slide onto the Create submenu.

▯▯▯ Release the button when the name of the icon you want to create is in reverse video.

▯▮▯ Click the **middle** button to deselect the icon.

or ▮▯▯ Click the **left** button to drop the icon and leave it selected.

Creating Components

To create a component:

▮▯▯ Hold down the **left** button and drag the cursor until the component caret is positioned where you want to create the new component.

▯▮▯ Hold down the **middle** button.

▤▯ Slide onto the Create submenu.

▯▯▯ Release the button when the name of the component you want to create is in reverse video.

The new component has the same properties as the master component of the same name. Once you create a component, you can change it to another component type using the Change *submenu.*

To duplicate a particular component (but not its contents):

▮◱◱ Hold down the **left** button and drag the caret through the component bar until it touches the component you want to duplicate.

◱◱◱ Release the button.

▢ Press LINE FEED.

A new component is created below the component with the caret. The new component's properties are identical to those of the component with the caret, even if these properties differ from the master.

Creating Frames

To create a frame:

▮◱◱ Hold down the **left** button and drag the caret through the text to the position where you want the frame or its anchor.

◱◱◱ Release the button.

◳▢ Hold down the **middle** button. Slide onto the Create submenu and then onto the Frame submenu.

◱◱◱ Release the button when the name of the frame you want to create is in reverse video.

The new frame inherits the properties of the master of that name.

◱▮◱ Click the **middle** button to deselect the frame.

Letting the Software Create a Frame for You

If you paste a diagramming object directly into text or into the component bar, the software attempts to create a frame the size of the object. For objects that exceed the page margins, the software creates a frame the full width of the page (ignoring margins) and adjusts the object proportionally to this frame.

Creating Diagramming Objects

To create a diagramming object:

↙ Move the mouse cursor into the frame where you want to create the object and click the left button twice to select and open the frame.

↖ Make sure that the mouse cursor is in that frame.

◳▢ Hold down the **middle** button. Slide onto the Create submenu.

▢▢▢ Release the button when the name of the diagramming object you want to create is in reverse video.

▢▢▢ Move the cursor until the object is the size you want it.

▢▢▢ Click the **middle** button to deselect the object.

Creating Text in Frames

You can include two types of text in frames.

- **Microdocuments** provide most of the features of a document inside a frame. They are made up of components and have their own page property sheets. However, you can also move and size them like any diagramming object, and use other diagramming commands to manipulate them.

- **Text Strings** are useful for short labels. They do not use most of the editing features of the publishing software. The default entries in header/footer frames are text strings.

To create a microdocument in a frame:

✔ **Open** the frame.

↖ **Point** the mouse cursor where you want the microdocument to be placed.

▢ Begin typing.

When you begin typing, a box appears with its top left corner at the cursor position; a default component called **caption** *is created.*

▢▢▢ When you are finished entering text, move the cursor so that it is pointing at an empty space in the frame and click the **left** button to close the microdocument.

You can cut or copy text from a microdocument and paste it outside a frame. Likewise, you can paste outside text into a frame.

To create a text string in a frame:

✔ **Open** the frame.

↖ **Point** the mouse cursor where you want the text string.

▢ Press **CTRL o** to create a text string, then **CTRL l** (for flush left text), **CTRL c** (for centered text), or **CTRL r** (for flush right text).

▢ Begin typing.

▢▢▢ When you are finished entering text, click the **left** button to deselect the text string.

How to Enter Text and Change Fonts

You can add new text to a document anytime the mouse cursor is within the boundaries of the document window by typing characters at the keyboard. The new text is entered to the left of the text caret.

To enter text in a document:

▮◻◻ Position the text caret where you want to begin your text entry.

◻ Begin typing.

Fonts and Attributes

The publishing software includes fonts with the following variations (not all variations are available in each family):

- **Families**: Symbols, Greek, Typewriter, Classic, Modern, Math, Symbols
- **Sizes**: 6, 8, 10, 12, 14, 18, and 24 points
- **Bold**: on and off
- **Italic**: on and off

You also can mark any character with the following attributes:

- **Underscore**
- **Strikethrough**

To change the default font of a component:

↙ **Select** the component and open its property sheet.

↙ On the Format sheet, make the changes needed to create the *new* default.

↙ **Apply** the changes.

To change the font or attributes of existing text:

↙ Select the text whose font or attributes you want to change.

◻▮◻ Hold the **middle** button and slide onto Fonts submenu until the change you want to make is in reverse video.

◻◻◻ Release the button.

To change the font or attributes at the caret
without changing existing text:

✔ Position the text caret where you want to enter text with the new font or attributes.

⫿ Hold the **middle** button and slide onto Fonts submenu until the change you want to make is in reverse video.

⫿ Release the button.

New text you enter at the point will have the new font or attributes.

You can also change the font or attributes either for selected text or at the caret position by using the Font pulldown menus in the document header.

Chapter 3

The Keyboard

Many of the actions that you can accomplish using the mouse or the mouse and a menu can also be accomplished using the keyboard or a mouse/keyboard combination. All Sun 2 and Sun 3 workstations provide **function keys** for many of these actions. They also support **keystroke combinations** that accomplish many of the same results.

> The illustrations in this chapter refer to the keyboards of new Sun 3 workstations. If you have a Sun 2 or a Sun 3 upgrade, note that the commands are the same, but you do not have keys labeled CAPS LOCK or ALT and your ⟨\⟩ key is next to the RETURN rather than next to the ⟨±⟩ key.

Most of the keyboard commands described in this chapter are text processing commands that work only within a document window. However, some keyboard commands perform other operations within the publishing software.

This chapter is divided into three sections. The first two sections, *The Function Keys* and *The Control Key Sequences*, describe the two types of command keys and how to use them. To accommodate the needs of different users, most of the basic functions can be performed using either type of key. Users who have experience with text editors may prefer to use the control key sequences.

The last section of this chapter, *Keyboard Commands,* contains a comprehensive list of keyboard commands and provides a description of each. The commands are categorized according to what they do. Each category includes several keystroke functions or special characters. Function and character names in a category are highlighted in bold type.

> If you own a typesetter and have specified it as the *Final Output Device* on the Printer property sheet, some characters shown in the font layouts at the end of this chapter will not be available to you. See the Interleaf typesetter documentation for more information.

The following list provides a quick reference to the location of commands by category:

The Function Keys

The Sun 3 keyboard (Figure 3-1) has function keys on the left keypad, on the right keypad, and on the top row of the main keypad. Many of the function keys have 3 levels of meaning, each determined by what you press *before* you press the function key. To execute the first level, press nothing before pressing the function key; to execute the second level, press the BACK SPACE key first; to execute the third level, press the ESC key first. For example:

- if you press the right arrow key (R12 on the right keypad), the text caret moves *one character* to the right;

- if you press and release the BACK SPACE key before you press the right arrow key, the caret moves *one word* to the right;

- if you press and release the ESC key before you press the right arrow key, the caret moves to the *end of the current line*.

L2			R3
cancel			back search fwd search
L4	R4	R5	R6
caps lock	insert hyphen	open line	replace
		R8 ⬆ begin of doc begin of comp prev line	
L8	R10 ⬅		R12 ➡
paste comp paste text	begin of line prev word prev char		end of line next word next char
L9 L10	R13	R14 ⬇	R15
copy comp cut comp copy text cut text	go to page prev window prev page	end of doc end of comp next line	go to page next window next page

```
            FUNCTION KEYS

       ESC:  | top line    |
BACK SPACE:  | middle line |
   NORMAL:   | bottom line |

For functions using the ESC key,
press and release the ESC key,
and then press the appropriate
function key.

For functions using the BACK
SPACE key, press and release
the BACK SPACE key, and then
press the appropriate function
key.
```

		F3 default last	F4 bold	F5 italic	F6 underline	F7 strikethrough	F8 smaller	F9 larger	back space
esc	! 1	@ 2 # 3	$ 4 % 5	^ 6 & 7	* 8 (9	) 0 _ -	+ =	| \	~ `
tab	Q W E R T Y U I O P						{ [	}]	del
ctrl	A S D F G H J K L					: ;	" '		return
shift	Z X C V B N M					< ,	> .	? /	shift	line feed
caps		hard space space							

Figure 3-1. Sun 3 keyboard

Control Key Sequences

All the CTRL key sequences involve holding down the CTRL key and typing a character. For example, to move the text caret forward one character, you hold down the CTRL key and type **f**. In Figure 3-2, the CTRL key and the keys that are used with it are shown with white backgrounds.

† These commands will be executed multiple times if you precede them with CTRL u.

Figure 3-2. The Control key sequences

Note that **CTRL h** and **CTRL j** mimic the **BACK SPACE** and **LINE FEED** keys, respectively, as those keys are used in the publishing software.

Repeating Commands Automatically

You can repeat some of the CTRL key commands automatically if you precede the CTRL sequence for the command with CTRL-u (labeled *count* in Figure 3-2). A message in the status line of the document header will tell you what the current count is. The initial default for count is *4*, which means that CTRL-u followed by CTRL-f will move the text caret four characters to the right.

You can change the value of *count* using two different methods. Using the first method, you type a number after pressing CTRL-u. For example, if you want to repeat a command 64 times, you could press the CTRL-u sequence and type the number 64 followed by the command you want repeated (e.g., CTRL-u **64** CTRL-f moves the caret forward 64 characters). In the second method, you change the value of the count by repeating the count sequence. Each time you execute the CTRL-u sequence, the count value is multiplied by 4. For example, if the message in the status line after you execute CTRL-u is *Count: 4,* to repeat your next command 64 times, you can execute CTRL-u two more times followed by the command you want repeated (i.e., CTRL-u CTRL-u CTRL-u CTRL-f translates into 4 x 4 x 4 = 64 and moves the text caret forward 64 characters).

Control X Sequences

Some commands combine the sequence CTRL-x (labeled *more* for *more commands* in Figure 3-2) with a character or an additional CTRL sequence. For example, to move the text caret to the beginning of the document, you hold down the CTRL key and type x. Then you release the CTRL key and type <.

Keyboard Commands

In this section, the keyboard commands are broken down into categories, most of which have their own command charts. In the charts that follow, function and character names are highlighted in bold type. Keystrokes separated by commas or the word *or* perform the same function.

Moving the Text Caret

You can use the keyboard in several ways to move the text caret within a document window. Many of these commands also work in property sheet fields. When referring to the arrow keys, we use their function key names. The arrow keys correspond to function keys on the right keypad as follows:

Left Arrow key	◄	R10
Right Arrow key	►	R12
Up Arrow key	▲	R8
Down Arrow key	▼	R14

Movements that do not require you to press the BACK SPACE or ESC key can be repeated by holding down the keys. For example, as long as you hold down the right arrow key (R12), the text caret will continue moving forward character by character in text. (Likewise, as long as you hold down the CTRL key and *f,* the caret moves forward in text one character at at time.)

If you hold down a function key after using it with the BACK SPACE or ESC key, the function key executes its *normal* (unshifted) command.

If one of these commands does not produce the result you expect, check to see whether any microdocuments are open. If you have a microdocument open, some of the caret movement commands will have their effect in the microdocument only, not on the main document.

Moving the Text Caret (continued)

Function	Keystroke	Result
Previous Character	R10 *or* CTRL–b	Moves the caret backward in text one character at a time.
Previous Word	BACK SPACE R10 *or* CTRL–]	Moves the caret backward in text one word at a time.
Previous Line	R8 *or* CTRL–p	Moves the caret up one line.
Previous Window	BACK SPACE R13 *or* CTRL–\	Scrolls text downward and moves the caret to the previous window.
Previous Page	R13	Scrolls text downward and moves the caret from text or a frame to the beginning of the previous page.
Next Character	R12 *or* CTRL–f	Moves the caret forward in text one character at a time.
Next Word	BACK SPACE R12 *or* CTRL–c	Moves the caret forward in text one word at a time.
Next Line	R14 *or* CTRL–n	Moves the caret down one line.
Next Window	BACK SPACE R15 *or* CTRL–v	Scrolls text upward and moves the caret to the next window.
Next Page	R15	Scrolls text upward and moves the caret from text or a frame to the beginning of the next page.
Beginning of Line	ESC R10 *or* CTRL–a	Moves the caret to the beginning of the current line.
Beginning of Component	BACK SPACE R8	Moves the caret to the beginning of the component in which it is positioned (or to the beginning of the previous component if it is at the beginning of the current one).
Beginning of Document	ESC R8 *or* CTRL–x <	Scrolls text down and moves the caret to the beginning of the document.
End of Line	ESC R12 *or* CTRL–e	Moves the caret to the end of the current line.

Moving the Text Caret (continued)

End of Component	BACK SPACE R14	Moves the component caret to the end of the component in which it is positioned (or to the end of the next component if it is at the end of the current one).
End of Document	ESC R14 *or* CTRL x >	Scrolls text upward and moves the caret to the end of the document.
Go To Page	ESC R13 *or* ESC R15	Gives you access to the Go To Page stickup menu.
Center Caret	CTRL-l	Moves the current caret position to the vertical center of the document window (unless the caret is at the beginning of the document).

Deleting

When you *delete* text, the text is completely erased.

Function	Keystroke	Result
Previous Character	DEL	Deletes backward in text, character by character.
Previous Word	BACK SPACE DEL *or* CTRL-w	Deletes backward from the caret to the previous space.
Next Character	CTRL-d	Deletes forward in text, character by character.
Next Word	CTRL-t	Deletes forward from the caret to the next space.
End of Line	CTRL-k	Deletes from the caret position to the end of the line.
Text From Frame	CTRL-d	Deletes the contents of a selected text anchor, but leaves the anchor.

Cutting

When you *cut* text, the text becomes the contents of a *from 'document name'* icon on the clipboard and is still available to you for pasting.

Function	Keystroke	Result
Text	L10, DEL, CTRL-d, *or* CTRL-k	Cuts *selected text* and puts it on the clipboard.
Components	BACK SPACE L10	Cuts *selected components* and puts them on the clipboard.

If you select text and then execute one of the *delete* command keys, the selected text is cut and is available for pasting.

Copying

Function	Keystroke	Result
Text	L9	Puts a copy of *selected text* on the clipboard.
Components	BACK SPACE L9	Puts a copy of *selected components* on the clipboard.

Pasting

Function	Keystroke	Result
Text	L8	Pastes what is *selected* on the clipboard at the position of the text caret.
Components	BACK SPACE L8	Pastes what is *selected* on the clipboard at the position of the component caret.

Searching and Replacing

Function	Keystroke	Result
Search Forward	R3 *or* CTRL–s	Searches forward in text for the string you specify.
Search Backward	BACK SPACE R3 *or* CTRL–r	Searches backward in text for the string you specify.
Replace	R6 *or* CTRL–q	Replaces the search string and continues searching.
Global Replace	CTRL–x CTRL–q *or* CTRL–x q	Replaces the search string globally.

Changing Fonts and Attributes

You can use these keys to change fonts and text attributes while you are typing or you can use the mouse to select text and then use a key to change the font or attributes of the selected text. Except for **Last**, each command key sequence operates the same way as its corresponding pulldown or popup menu command.

The **Last** key does not mimic the **Last** popup command. If you select text and press the **Last** key, the text is not changed to the last font. Instead, the last font shows in the header bar and *new* text entered at the caret will have that font. Until you have used one font and changed to another in an editing session, **Last** defaults to the font shown in the document header when you opened the document.

The effects of the **Bold** and **Italic** keys vary depending upon whether you have optional fonts that include bold-italic characters.

Function	Keystroke	Result
Last	F3, CTRL–x CTRL–l *or* CTRL–x l	Changes the font at the text caret to the last font you used.
Default	BACK SPACE F3	Changes the font at the text caret to the default for the component.
Bold	F4, CTRL–x CTRL–b *or* CTRL–x b	This key toggles bold on and off at the text caret. In selected text it turns bold off where it is on and on where it is off.

Changing Fonts and Attributes (continued)

Function	Keystroke	Result
Italic	F5, CTRL-x CTRL-i *or* CTRL-x i	This key toggles italic on and off at the text caret. In selected text it turns italic off where it is on and on where it is off.
Underline	F6	This toggle key turns underline on and off at the caret. It underlines selected text that is not underlined. It removes underlining in selected text that is underlined.
Strikethrough	F7	This toggle key turns strikethrough on and off at the caret. It strikes through selected text that does not have strikethrough on. It removes strikethrough in selected text that has strikethrough on.
Smaller	F8 *or* CTRL-x [	Changes the font at the text caret to the next smaller size in the same family. It changes selected text the same way. If you select different sized fonts, this command reduces each one to its next smaller size.
Larger	F9 *or* CTRL-x]	Changes the font at the text caret to the next larger size of the same family. It changes selected text the same way. If you select different sized fonts, this command increases each one to its next larger size.

Checking Spelling

Function	Keystroke	Result
Search for Next Spelling Error	CTRL-x CTRL-e *or* CTRL-x e	Searches for the next misspelled word.
Add Word to Dictionary	CTRL-x CTRL-a *or* CTRL-x a	Adds one or more selected words to the desktop dictionary.

Creating and Aligning Text in a Frame

When you enter text within a frame, by default you create a *microdocument*. If you instead want to create a text string (a simpler text object without the structure of a document), you must use one of three control sequences before you enter text. The sequence you choose determines the alignment of the text string.

Function	Keystroke	Result
Text String	CTRL-o	Creates a text string.
Flush Left	CTRL-l CTRL-o *or* CTRL-o CTRL-l	Creates a left-aligned text string. Aligns the left side of a selected text string with the text anchor.
Center	CTRL-c CTRL-o *or* CTRL-o CTRL-c	Creates a centered text string. Aligns the center of a selected text string with the text anchor.
Flush Right	CTRL-r CTRL-o *or* CTRL-o CTRL-r	Creates a right-aligned text string. Aligns the right side of a selected text string with the text anchor.

Additional Text Processing Commands

Caps Lock (CAPS LOCK or L4) toggles the case of characters. If you press the key when the keyboard is in lower case mode, the characters you type will be upper case until you press the key again. The key labeled CAPS LOCK works both in the publishing software and in virtual terminal windows. The L4 function key works as a caps lock key only in the publishing software.

Create a Component (LINE FEED or CTRL-j) creates a new component after the component with the caret.

Open (R5 or CTRL-o) clears space for text insertion. Normally, when you position the caret in text and begin to type, the text that is already there moves to the right to make room for the new characters. The **Open** command moves the text between the caret and the end of the line *down* a line and inserts a carriage return. However, the text caret does not move to the next line as it does when you insert a regular carriage return, so this action clears space where you can insert text. When you have inserted as much text as you like, you can delete the carriage return to close the space.

Creating Special Characters

Special Characters

Several special characters are available through a combination of ESC sequences or function keys. You must repeat the sequence each time you want to use the special character. These characters are available on the Standard and Greek keyboards only (see the keyboard layouts at the end of this chapter). For more information about special characters, see the chapter *Typography and Typefaces.*

Any character available in the publishing software has a decimal value associated with it. Using this value and the **decimal character** command key sequence, you can produce characters that are not mapped to specific keys. This can be especially useful if you have purchased optional fonts. This command only works in text (including microdocuments). It will not enter characters in property sheets, stickup menus, or text strings in frames.

Character	Keystroke	Result
Double Opening Quotation Mark (")	ESC ["]	Inserts a double open quotation mark.
Single Opening Quotation Mark (')	[~]	Inserts a single open quotation mark. (This is not an ESC sequence.)
Hyphen (-)	ESC [-] *or* R4	Inserts a hyphen.
En Dash (–)	[-]	Inserts an en dash (same as a figure dash or minus character). (This is not an ESC sequence.)
Em Dash (—)	ESC [±]	Inserts an em dash.
Cent Sign (¢)	ESC [$4]	Inserts a cent sign.
Bullet (·)	ESC [|]	Inserts a bullet.

Special Characters (continued)

Force Justify Character (⇥)	ESC RETURN	Inserts a force justify character, which distributes the text preceding that character across the entire line length in a justified component. You can use this to force a short line to justify. In non-justified components, this character has the same effect as a hard return.
British Pound Sign (£)	ESC ⌈L⌉	Inserts a British Pound sign.
Section Mark (§)	ESC ⌈⁚⌉	Inserts a section mark.

Tokens

Certain keystroke sequences in the publishing software, when executed in frames, produce *tokens* that expand into a text strings. If you open a frame in a document and execute the Document Name sequence, the name of that document appears in that frame. If you execute the Page Number sequence in a frame, the correct page number is displayed on that page. If you execute the Page Number sequence in a *header/footer* frame, the correct page number will display on some or all of the other pages in the document, depending upon the page layout you have chosen. If you change the name of document or shift the position of pages, the tokens automatically pick up and display the new information.

Character	Keystroke	Result
Page Number	ESC SHIFT ⌈#⌉⌈3⌉ (ESC #)	Inserts a token for the page number of the page the frame is on.
Document Name	ESC SHIFT ⌈@⌉⌈2⌉ (ESC @)	Inserts a token for the name of the document the frame is in.

Spaces

The software provides several different kinds of spaces. When you press the space bar, you get a soft space, which allows the words on either side to be separated from each other by a break at the end of a line. Soft spaces can stretch or shrink to even out spacing in lines of justified text.

Hard spaces treat the words they divide as though they were one word. If *word + hard space + word* will not fit together at the end of a line, the whole group moves to the next line. Note that if on the Page property sheet *Hyphenation* is **On**, the words connected by a hard space may be hyphenated. To force the connected words to move to the next line, you may have to clear all hyphenation points. For more information about hyphenation, see the chapter *Spell Checking and Hyphenation*.

There are four hard spaces of different sizes. All hard spaces are available on the Standard and Greek keyboards (see the keyboard layouts at the end of this chapter). The *natural hard space* is available in every font. Hard spaces do not stretch, shrink or break. Figure 3-3 shows the relative width of the five different spaces in 12-point *Modern*.

```
This space is a soft word space.
This space is a natural hard space.
This space is a 12 pt. thin space.
This space is a 12 pt. en space.
This space  is a 12 pt. em space.
```
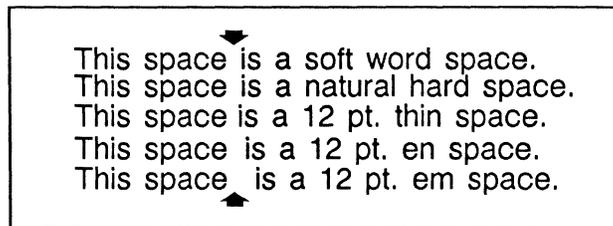
Figure 3-3. Five possible space characters

Character	Keystroke	Result
Natural Hard Space	**ESC SPACE BAR**	Inserts the space determined by Interleaf as the one that gives the best spacing for each font.
Thin Space	**ESC a**	Inserts a space the width of a punctuation mark.
En Space	**ESC s**	Inserts a space the width of the space that a numeral occupies in each font.
Em Space	**ESC d**	Inserts a space the width of the *em dash* in each font.

Fractions

It is possible to combine superscript and subscript numbers from the 10-point Symbols font with a slash from the Math A font to produce fractions. If you expect to use fractions often, we recommend that you create a document containing the fractions you expect to use frequently. Then you can copy the fractions and paste them where you need them.

> The Math A slash has zero width, which means that it is not possible to select it by itself. You must either type the slash every time you need it or create fractions which you can select as a whole.

To make a fraction:

✔ Change the font to Symbols 10-point.

✔ Type a superscript for the numerator.

In Symbols, the shifted numerals are the superscripts.

✔ Type a subscript for the denominator.

In Symbols, the unshifted numerals are the subscripts.

✔ Position the text caret *between* the superscript and the subscript.

✔ Change the font to Math A.

✔ Hold down the SHIFT key, and type 2.

In Math A, this key produces a zero width slash.

Here are examples of commonly used fractions, built using the method suggested above:

½ ¼ ¾ ⅛ ⅜ ⅝ ⅞ ⅓ ⅔

and a few that are less common:

$^{16}\!/_{32}$ $^{75}\!/_{234}$ $^{90}\!/_{143}$

> An equation-building option is available for the Interleaf software. This option allows you to create multilevel equations using property sheets.

Interrupt and Cancel Commands

You have a choice of several different commands to interrupt the publishing software or cancel a process that has not yet been completed. The Interrupt stickup menu, which is summoned by the command CTRL-z, gives you access to a new series of commands that may be executed using the menu or by adding a letter after you press CTRL-z.

In documents, the cancel commands will stop printing, searching/finding, and re-placing. They will also cancel the opening of a document. For multiple selected icons, these commands will cancel cut, copy, and props operations after they have been completed for one icon.

Please read the description of the Interrupt stickup in the chapter *Basic Procedures* before using any of the commands on that menu.

Character	Keystroke	Result
Cancel	L2, CTRL-g, *or* CTRL-z C	Cancels and operation in progress.
Interrupt	CTRL-z	Causes the Interrupt stickup menu to appear. Each option on this menu is followed by a character in angle brackets. You can select an option by typing the corresponding character or by clicking the mouse button on that option. In a terminal window, CTRL z does not interrupt the publishing software but does interrupt other processes running in that window.

Additional Commands

Exit from Virtual Terminal Window (CTRL-d) at the operating system prompt exits from the virtual terminal and closes its window. If your system does not respond to CTRL-d, you may exit from the virtual terminal window by typing **exit** at the operating system prompt. If your system does not respond to either command, see your system administrator.

Save Document (CTRL-x CTRL-s or CTRL-x s) saves a document that has been modified. This command works when the mouse cursor is in a document window and the text caret is visible in regular text or a microdocument. It does not work when a frame is open in the document unless a microdocument is also open.

ESCAPE
SHIFT
NORMAL

Standard

Math A

Spaces | Thin | En | Em

hard space

space

(spaces work with every font)

Symbols

superscript

subscript

Math B

Greek

Math Extension

Spaces | Thin | En | Em

Chapter 4

Text Processing

All Interleaf documents are made up of **components**, building blocks that give structure and definition to the page. When you create a document on your desktop, the document has a single empty component named *paragraph*. This component has a set of properties — margins, typeface, etc. — that determine both its overall appearance and its relationship to any other components that you add.

From this one component, you can construct a varied format. For example, this *Reference Manual* started with one *paragraph*. Now it consists of a number of different kinds of components that were given the properties appropriate to their functions in the manual.

On this page, there are components named *title2, para,* and *note,* among others. Any time we create a *note* component, we know that it will have a left margin of *1.5* inches and be appropriate for information that we want to stand out from the rest of the text on a page. If we decided that the information in a *note* was not so special after all, we could easily change the *note* into a *para* by using another feature of the editor.

In addition to the editing and text processing features available at the component level, many editing features are also available to you within the text of a component. For example, you can specify a typeface for an entire *component* and then specify a different typeface for *any portion of text* within it. You can cut and paste whole components or any selection of text you choose.

The majority of this chapter describes basic text processing activities applicable to any document, including microdocuments. See the chapter *Text in Frames* for details on microdocuments. Certain special considerations affect text processing in a *multicolumn* document. Multicolumn pagination issues are discussed in the chapter *Page Design*.

This chapter covers the following major topics:

- cursors and carets in a document window
- component master definitions
- the commands available for creating and manipulating components
- the commands available for manipulating text
- the Format and Custom sheets of the Component property sheets
- paging and scrolling through documents
- tabs and the Tab sheet of the Component property sheets

Cursors and Carets in a Document Window

In a document window, there can be three different place-marking devices: the *mouse cursor,* which can be moved anywhere in or outside of the window; the *text caret,* which exists only in the text region of the document window; and the *component caret,* which exists only in the component bar and appears as a white line above or below the name of one component when no components are selected. Both the mouse cursor and the text caret take various forms depending on circumstances. For details on the various shapes of cursors and carets, see the chapter *Fundamentals.*

When the mouse cursor is in the document window, you generally can edit text in that document. You *cannot* edit text in that document when

- the diagramming cursor is visible, indicating that a frame is open
- the text is within a closed frame and/or closed microdocument
- certain property sheets *for that document* are open. If you attempt to edit with one of these property sheets open, you will see the message: *Please close the property sheet first.*

If the mouse cursor is in the component bar, the header, or a scroll bar when you begin editing text, it remains where it is until you move the mouse.

However, if the mouse cursor is on the text page itself, it is merged into the text caret as soon as you use the keyboard. This both prevents the mouse cursor from getting in the way of your editing and lets you know exactly where the mouse cursor is when you want to use it to do something, such as mark text or scroll your document.

If you want to edit text that appears within a frame, see the chapters *Text in Frames* and *Text as a Diagramming Object* for details.

The Component and Text Popup Menus

In the component bar, the Component Location and the Component Selected popup menus are available. In text, the Text Location and the Text Selected popup menus are available.

When the mouse cursor is in the component bar and you hold down the middle button, you see one of the **Component** menus (Figure 4-1).

Component Location menu

- When the component caret appears above or below a component name, you can access the **Component Location** menu.

Component Selected menu

- When one or more components are selected, you can access the **Component Selected** menu.
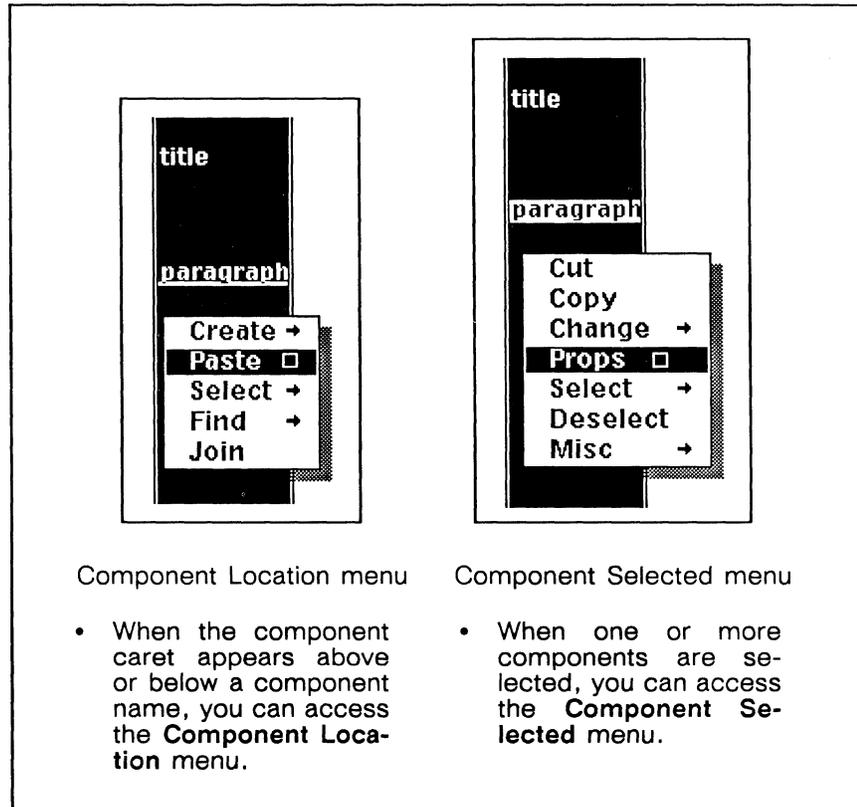
Figure 4-1. Component popup menus

When the mouse cursor is in text and you hold down the middle mouse button, you see one of the **Text** menus (Figure 4-2).
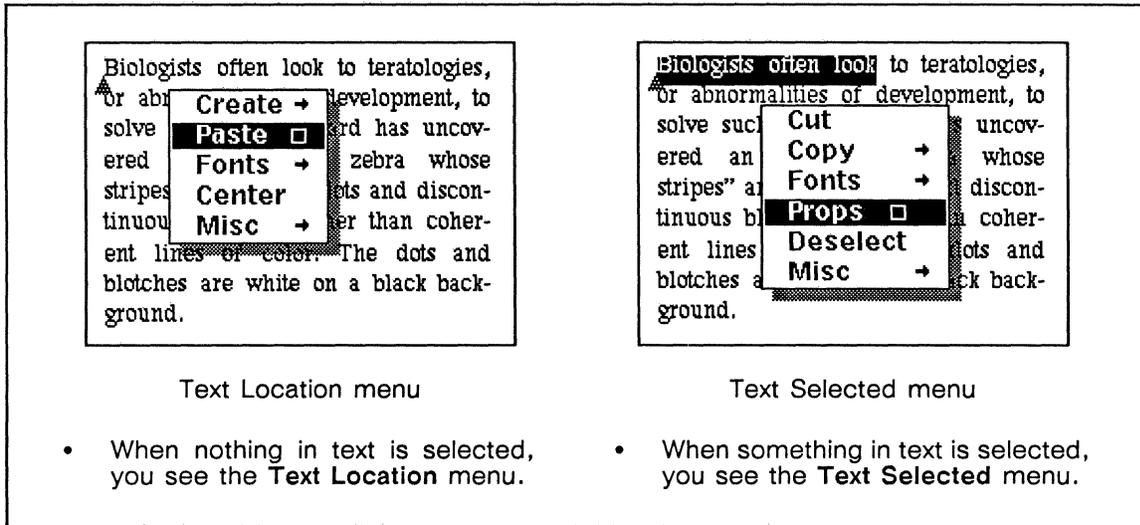
Figure 4-2. Text popup menus

Components

With Release 3.0, the publishing software uses a new method for identifying and storing the **master definition** of a component. In previous releases, component master definitions were derived from the *first* occurrence of each component name in the document. Now, for each uniquely named component the software maintains a master definition that is independent of individual occurrences of the component. Essentially, this changes the way components inherit their properties.

Below are some of the basic concepts pertaining to component master definitions.

- There is no separate property sheet that corresponds to the master definition. You define the properties of a component master using the four Component property sheets.
- The list of component master names appears on certain Component submenus (e.g., the Create submenu).
- To see the master definition of a component, create a new instance of the component name and open its property sheets.
- When you paste a uniquely named component into a document, the master definition for that component in the originating document (from which you cut or copied) becomes the master definition for that component in the destination document (where you paste).
- Even if you cut the last instance of a component, the master definition is saved.
- When you save a document, all master definitions that you have not deleted are saved regardless of whether an instance exists in the document.

For details on how to use component masters to create templates (for sets of documents that you want to structure the same way) see the chapter *Document Templates.*

Creating Master Definitions

The following exercises describe in detail how to create, see, and change master definitions.

To create a new master definition:

✔ **Create** a component and open the property sheet.

The properties of the component you create will become the default properties of the new component master. For convenience, you may want to create a component which has master properties similar to those you want for the new master.

✔ Change the name of the component to something unique
and **Apply** the change.

The status line will display the message: Created the master definition
for: 'componentname'. *You will see this new component listed on the
Component submenus.*

In your document, you now have one instance of the new component you just cre-
ated. If you cut that component and check the Component submenus, you will see
that the component name is still there.

To see the master definition for a component:

✔ **Create** a new instance of the component whose master defi-
nition you wish to see.

✔ Open the Component property sheet for this component.

*The properties displayed reflect the current master definition for the
component.*

To change the master definition of a component, you must first open the property
sheet of a component with the same name as the master. Either you can open
the property sheet of an existing instance or create a *new* instance. The *new* in-
stance will reflect the current master definition whereas an existing instance may
not. Next you make the changes desired and execute either **Global Apply** or
Unify. Both commands will affect the master definition and all instances of the
component in the document. However, **Global Apply** applies only the latest
changes whereas **Unify** applies *all* of the properties of the component with the
open property sheet to the master and all instances.

To change the master definition:

✔ **Create** an instance of the component whose master defini-
tion you wish to change and execute **Props**.

or ✔ Select an instance of the component whose master you wish
to change and execute **Props**.

*If you have changed the properties of this particular instance, the
property sheet may not reflect the current master definition.*

✔ Open the Component property sheet for this component and
change the properties as you wish.

*You can change the properties on any one or all four of the Compo-
nent property sheets.*

✔ **Global Apply** the changes.

The status line will display the message: Updated the master definition
for: 'componentname'.

*All instances of this component will reflect the changes made to the
master.*

Deleting Master Definitions

You may find that you want to remove one or more master definitions. To delete a single master, you must globally apply a name change. The results of this name change depend on two conditions:

- whether the name you enter is unique or the name of another master already defined in the document
- whether the document contains instances of the component master you are deleting

Figure 4-3 shows the four possible combinations of results from a global component name change. In all cases the software informs you that you are deleting a master definition and asks for confirmation.

	Document contains *no** instances of the master	Document contains in-stances of the master
Change to an *existing* master name	• deletes the old master	• deletes the old master • properties of the existing master are globally applied to the instances
Change to a *unique* master name	• deletes the old master • creates a new master with the unique name	• deletes the old master • creates a new master with the unique name • new master name globally applied to the instances; properties of instances do not change

*Except the instance you created to make the global name change.

Figure 4-3. Effects of globally applying component name changes

To delete a single component master definition:

✔ **Create** an instance of the component whose master definition you wish to delete *or* select an instance if one already exists.

✔ Open the Component property sheet for that component and change the name to an existing *or* unique component name.

✔ **Global Apply** the name change.

A stickup menu will appear asking you to confirm the deletion of the master definition for the original component (Figure 4-4).
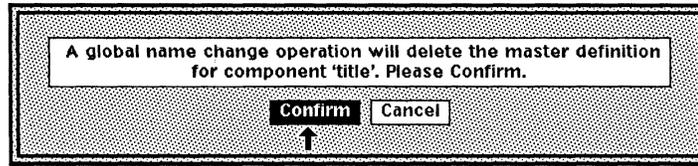
Figure 4-4. Delete Component Master stickup menu

✓ Select **Confirm** on the stickup menu and **Close** the Component property sheet.

Now, if you check the list of components on the Component submenus, you will find that the original component name no longer appears on the list.

Note that you have globally applied the component name you entered to all instances that had the name of the deleted master. In addition, if you entered the name of an existing master, you have globally applied the properties of the existing master to any instances of the deleted master.

To remove several masters you can use the **Purge Components** command, which deletes the component master definitions for all components not used within a document (Figure 4-5).
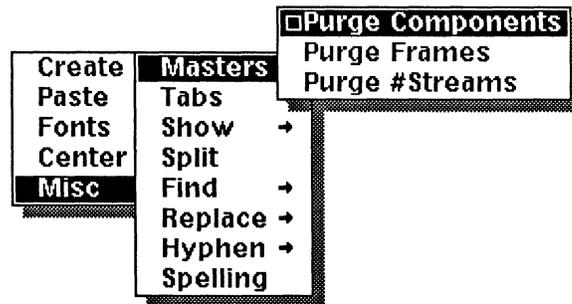
Figure 4-5. Misc Masters submenu on the Text Location menu

Before using **Purge Components** make sure an instance exists for every component master you want to save.

To delete the master definitions for all components without instances:

✓ On the Text Location menu select the `Misc →` `Masters →` `Purge Components` command.

The stickup menu shown in Figure 4-6 will appear.
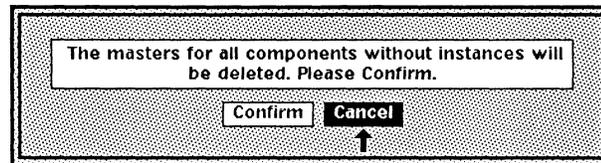
Figure 4-6. Purge Masters stickup menu

↙ Select **Confirm** on the stickup menu.

If you check the Component submenus, you will find that the components without instances no longer appear on the list of components.

A Summary of the Master Definition Concept

Below is a summary of the more advanced concepts pertaining to component master definitions.

- To create a new master definition, create an instance of any component and rename it with a unique name. The properties associated with the previous master become the default properties of the new master.

- To change the properties of a master, open the property sheet of any component with the same name as the master. Modify the properties of the instance and globally apply or unify the changes, depending on your goal.

- To delete a single master, you must globally apply a name change. The full results of this name change depend on the circumstances.

- To delete the master definitions for all components not used within a document, use the **Purge Components** command.

The Component Caret

The **component caret** always indicates the location where an action will occur when you choose a command on the Component Location menu. Figure 4-7 shows how the placement of the *component caret* affects the execution of a command.

When the component caret is *below* the name of a component, a command will have its effect below the component.

If the component bar looks like this:

| title 1 |
| paragraph |
| paragraph |
| list |
| list |
| title 2 |

when you create a new component with **Create**, the new component appears *after* the component with the caret. If you use the Create default, the component bar will look like example (a). However, you can create any kind of component you like. The component bar would look like example (b) if you created a paragraph.

| title 1 |
| paragraph |
| paragraph |
| list |
| list |
| list |
| title 2 |

(a)

| title 1 |
| paragraph |
| paragraph |
| list |
| list |
| paragraph |
| title 2 |

(b)

When the component caret is *above* the name of a component, a command will have its effect above the component.

If the component bar looks like this:

| title 1 |
| paragraph |
| paragraph |
| list |
| list |
| title 2 |

when you create a new component with **Create**, the new component appears before the component with the caret, as in examples (c) and (d).

| title 1 |
| paragraph |
| paragraph |
| list |
| list |
| list |
| title 2 |

(c)

| title 1 |
| paragraph |
| paragraph |
| list |
| paragraph |
| list |
| title 2 |

(d)

Figure 4-7. Effect of component caret position on the Create command

When no components are selected, the component caret is always directly above or directly below the name of a component. We call that component *the component with the caret.*

If you execute a command that moves the component caret, the caret appears *above* the new component if it was *above* its previous component; the caret appears *below* its new component if it was *below* its previous component.

Creating Components

You can use either the LINE FEED key or the **Create** command on the Component Location menu to create components, but depending on the situation one is more useful than the other.

The Create Command

The **Create** command has a submenu listing the names of the component masters defined in the document. With the **Create** command you can:

- create at any location a new instance of any component on the submenu; this instance will have the master properties;
- create a component that is different from the component in which the *text caret* is positioned.

When you create a new component, it is selected so that you can:

- easily see which component was created;
- immediately execute the **Props** command on the Component Selected menu, if you want to check or change its properties;
- easily **Cut** or **Change** the component if the type you created is not the type of component you want.

The **Create** submenu is **dynamic**, which means that it grows when you add unique component master names to a document and shrinks when you delete component masters (Figure 4-8).
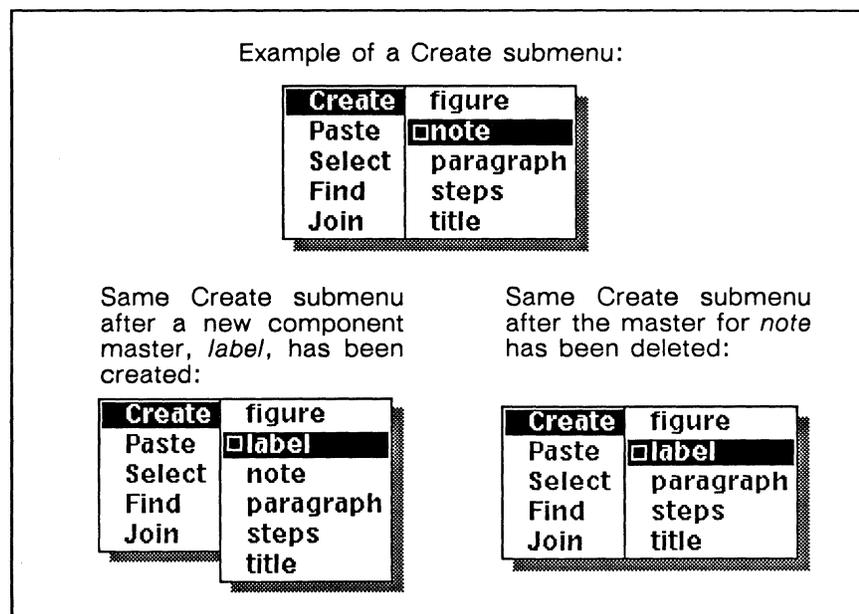
Figure 4-8. A dynamic menu

Note that once you create 26 component masters in a document, the Create submenu splits into two submenus each containing 13 component names (Figure 4-9). To see the component names displayed on the second half of the submenu, slide onto the More submenu.

```
┌──────────┬──────────┐
│ Create   │ chapline │
│ Paste    │ chapter  │──┬──────────┐
│ Select   │ explain  │  │ space    │
│ Find     │ figure   │  │ steps    │
│ Join     │ label    │  │ stepsAT  │
│          │ list     │  │ title1   │
│ title2   │ listAT   │  │ title2   │
│          │ note     │  │ □title3  │
│          │ noteAT   │  │ title3E  │
│          │ noteline1│  │ title4   │
│          │ notelne2B│  │ title4E  │
│ para     │ notelne2T│  │ title5   │
│          │ para     │  │ title5E  │
│          │ More...  │  │ title6   │
│          │          │  │ titlexcer│
└──────────┴──────────┴──┴──────────┘
```
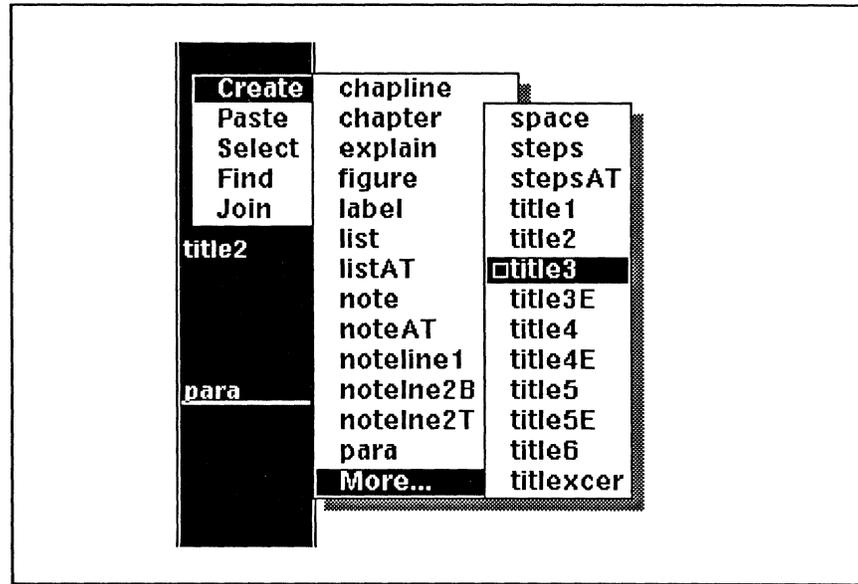
Figure 4-9. Create submenu with 26 component master names

The default on the **Create** submenu is the name of the component with the caret. When you execute the **Create** command, the component created inherits the properties of the master for the component name chosen.

Creating Components Using LINE FEED

You can also create a new component using the LINE FEED key, which enables you to create a series of components without taking your hands off the keyboard. Unlike **Create**, LINE FEED always creates a component immediately *following* the component in which the *text caret* is positioned. The newly created component will have the same properties as the component in which the text caret is positioned. Because this technique does not rely on component caret position, you can use it to create a new component when other components are selected.

Note that the properties of a component created using LINE FEED are not necessarily the same as the properties of the master definition for that component. If you use LINE FEED and then decide to make sure the new component has the properties of the component master, use the **Change** command on the Component Selected menu.

Selecting Components

You can use the left and right mouse buttons to select and deselect components. There are also selection and deselection commands on *Component* menus.

For details about component selection in multicolumn documents, see the chapter *Page Design*.

Using the Select Commands on the Component Location Menu

The **Select** command appears on both of the Component popup menus.

When you execute **Select** within a document that contains closed microdocuments, the software does not select components within the microdocument; likewise, when you execute this command from within a microdocument, the software does not select components outside the microdocument.

On the Component Location menu, **Select** has a submenu that gives you five options (Figure 4-10a), one of which has its own submenu (Figure 4-10b).

Create	All
Paste	All Of →
Select	□Thru Page
Find	To Front
Join	To End

(a) The Select submenu

		chapter
		figure
		□label
Create	All	note
Paste	**All Of** ₀	paragraph
Select	Thru Page	steps
Find	To Front	title1
Join	To End	title2

(b) The Select All Of submenu
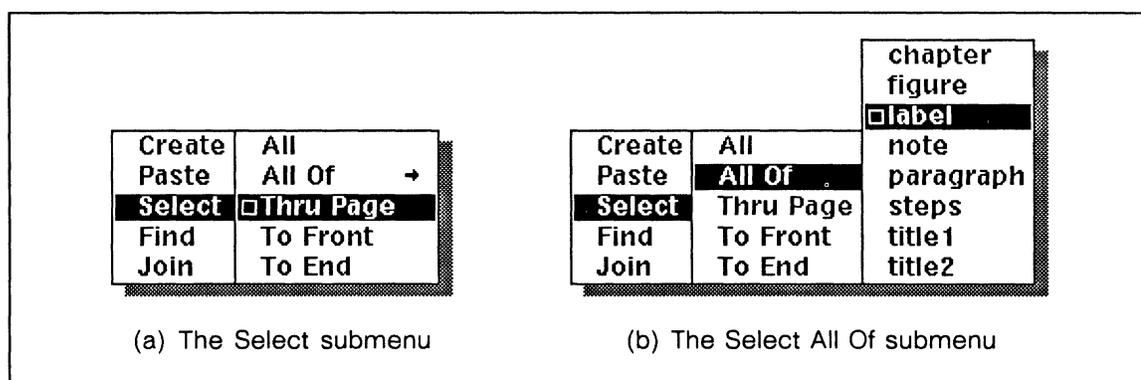
Figure 4-10. Select submenus of the Component Location menu

The Select All Command

Select All selects every component in the document.

The Select All Of Command

Select All Of has a submenu listing the names of the component masters in the document. With **Select All Of**, you can select all of the components with a particular name. The default is the name of the component with the caret.

The **Select All Of** command is useful when you want to do something like copy all of the *note* components in your document. You can **Select All Of** the *notes*, and then use the **Copy** command on the Component Selected menu.

The Select Thru Page Command

Select Thru Page selects all of the components from the location of the component caret to the end of the page you designate on a stickup menu.

There are two things to remember about **Thru Page**:

- The command works in either direction — for example, from the caret position on *page 23* through the last component on *page 25* or from the caret position on *page 25* through the first component on *page 23*.

- Whether a component name is above or below the caret determines whether that component is included in the selection.

The Select To Front and To End Commands

Select To Front selects all of the components between the component caret and the beginning of the document.

Select To End selects all of the components between the component caret and the end of the document.

Selecting Additional Components

Once you have selected components, when you hold down the middle button you see the Component Selected menu. The **Select** command on this menu is used to alter your initial selection. **Toggle** is the default on the Select submenu (Figure 4-11a). **All Of** on the Select submenu has a submenu of its own (Figure 4-11b).

Cut					Cut				figure
Copy					Copy				label
Change	►	All			Change	►	All		list
Props		All Of ►			Props		**All Of**		paragraph
Select		▢Toggle			**Select**		Toggle		▢steps
Deselect					Deselect				title1
Misc	►				Misc	►			title2

The Select submenu   The Select All Of submenu
(a)   (b)
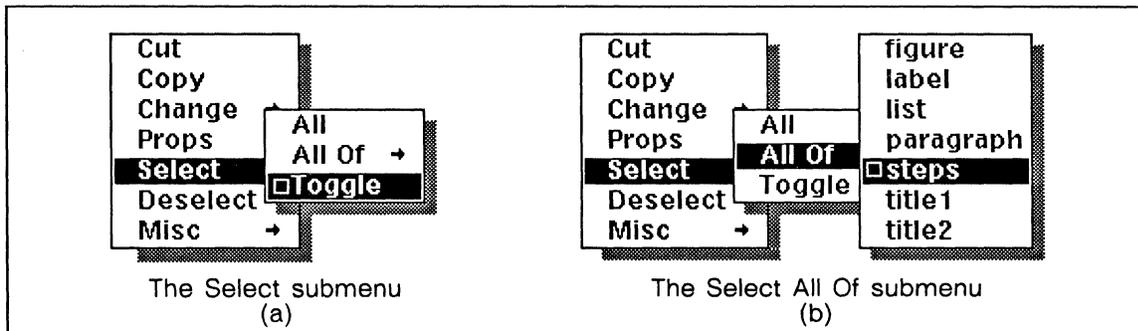
Figure 4-11. Select submenus of the Component Selected menu

The Select All Command

Select All selects the rest of the components in the document, so that every component is selected.

The Select All Of Command

Like the similar command on the Component Location menu, this **Select All Of** command lets you select all of the components of a particular name. In this case, it allows you to add to a previous selection of components.

The Select Toggle Command

Select Toggle deselects the selected components and selects the others. This command is useful when you want to do something like copy 50 components out of a total of 55 components. You can select the 5 you do *not* want to copy, and then use **Toggle** to reverse the selection.

Use of a menu to select components does not deselect any previously selected components. If you are not *adding* to a selection, it is a good idea to start any component selection process by clicking the left button while in a blank area of the component bar. This will prevent you from inadvertently cutting or otherwise acting on a selected component that you cannot see.

Deselecting Components

There are several ways to deselect components. Which one you use depends on circumstances.

The following actions deselect every selected component and place the component caret below the name of the component the text caret is in:

- editing text in any way (inserting or deleting characters at the keyboard or using the **Cut, Create Frame, Paste,** or **Split** commands on the Text menus)
- clicking the left button while the mouse cursor is between components in the component bar (that is, so the component caret appears)

The following actions deselect individual components, leaving selected any other selected components in the document:

- positioning the mouse cursor over a selected component name and clicking the right button
- holding down the right button and moving the mouse cursor over the names of the selected components you want to deselect

When you use the **Deselect** command on the Component Selected menu:

- every selected component is deselected
- the component caret is *below* the name of the component in which the text caret was positioned when you executed the command

The second result makes **Deselect** particularly useful if you have been scrolling through a document selecting components and then decide you do not want to do what you thought you did.

If you use the **Deselect** command and the component with the text caret in it is off the screen, move the mouse cursor into text, and execute the **Center** command on the *Text Location* menu. The line in which the text caret is located will be centered in the document window.

Combining Components

The **Join** command on the Component Location menu unites one component with another. If the component caret is above the component name, **Join** unites this component with the one above it. If the caret is below the component name, **Join** unites the component with the component below it.

There are two things to understand about this command:

- If the components being joined differ, the combined component will have the name and properties of the top component.
- You will usually have to insert a space at the place where the two components are joined. Since the text caret will be at this point, you can simply press the space bar.

Splitting Components

The ⌷Misc →⌷ ⌷Split⌷ command on the Text Location menu causes the component to break into two components at the text caret position. Each new component has the same name and properties as the original. The text caret appears at the beginning of the new component. If you want to undo a **Split** command, immediately execute the **Join** command on the Component Location menu. **Split** is the default command on the Misc submenu.

Finding Components

The **Find** command on the Component Location menu is used to reach a specific component quickly. When the command finds the specified component, it moves the component and text carets to this component. If the component caret was above the component name when the **Find** command was initiated, it will be above the found component name. If the component caret was below, it will be

below the found component name. The text caret always moves to the beginning of the component. Figure 4-12 shows the Find submenu.

Create	First
Paste	Last
Select	□Next
Find	Previous
Join	Next of →
	Prev of →
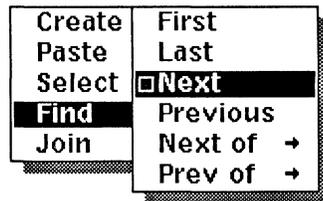
Figure 4-12. Find submenu

If the **Find** command reaches the end or the beginning of the document without finding the component you are looking for, you will see a stickup menu asking if you want to continue the operation by wrapping to the beginning or the end of the document.

When you execute **Find** within a document that contains closed microdocuments, the software does not search the microdocuments; likewise, when you execute this command from within a microdocument, the software does not search for components outside the microdocument.

Find First and Find Last Component

Find First finds the first component in your document, and **Find Last** finds the last one, thus giving you quick ways to scroll to the beginning or the end of your document.

Find Previous Component

Find Previous finds the component immediately before the component with the caret. It is useful when the name of the previous component is not visible in the component bar. This command becomes the temporary default, so you can use it to move quickly backward through a document, component by component.

Find Next Component

Find Next finds the component immediately after the component with the caret. It is useful when the name of the next component is not visible in the window. This command becomes the temporary default, so you can use it to move quickly forward through a document, component by component.

Find Previous of and Find Next of

Find Prev of and **Find Next of** both have submenus that consist of the list of component masters for the document. With either of these commands, you can

find the nearest example of a named component in the specified direction. These commands become temporary defaults, so you can use them to move quickly through a document to find a *specific* component of a given name.

These commands can be useful to examine the format as well as the content of a particular component. For example, you may not be sure whether you want to create a *list* or a *steps* component in a certain location, or whether to create an entirely new component. With **Prev of** or **Next of,** you can find examples of each and look at them and their property sheets to see whether or not you need a new component.

The **Find** commands are especially useful in multicolumn documents, in which component order is more complex.

Changing the Component Type

The **Change** command on the Component Selected menu makes it possible for you to change components with one name into components with another name. By means of this name change, you can alter the properties of components without opening Component property sheets. When you use the **Change** command, the selected components inherit all of the properties of the master for the name you choose on the Change submenu. The **Change** command, like the **Create** command, uses the component master to define component properties.

The command is useful when you want to change a component completely; for example, change a *list* component to a *paragraph* component. Font changes as a result of a change command behave in the same way as font changes brought about by a change on a Component property sheet.

Change is also useful when you want to be sure a component is exactly like the master of its name in the document, especially when you have pasted components from other documents. If the *paragraph* components you paste are in 10-point Modern roman and the *paragraph* components in the document into which you are pasting them are in 12-point Classic roman, you can change the font (and other properties) of the pasted *paragraphs* by using [Change ➔] [paragraph] .

Note that if you change all components of one type to another type, the master component for the first type will not be affected. For example, if you select all of the *title3*'s and **Change** them to *title2*'s, *title3* will continue to appear on the relevant submenus even though there is no occurrence of *title3* in the document.

Defining the Format of a Component

The **Props** command opens the four *Component* property sheets — the **Format** sheet, the **Tab** sheet, the **Page** sheet and the **Custom** sheet. You can see only one of these sheets at a time. Figure 4-13 shows the Format and Custom sheets for the *paragraph* component in the default document. The Tab sheet is discussed later in this chapter in the section *Setting Tabs*. The Page sheet is discussed in the chapter *Page Design*.

```
Component Properties:  Format  [ Tab ][ Page ][ Custom ]

Name           [paragraph]  ↑
Margins
   Top            [ 0.07 ]   inches
   Bottom         [ 0.07 ]   inches
   Left           [  0  ]    inches
   Right          [  0  ]    inches
First Indent      [  0  ]    inches
Line Spacing      [ 1.31 ]   lines

Alignment      Flush Left [ Centered ] Flush Right
Font           [Classic][10][ Bold ][ Italic ]
Attributes     [ Under ][ Strike ][ Revision ]
```

(a) Format sheet

```
Component Properties:  [ Format ][ Tab ][ Page ] Custom

Hyphenation    [Off][1][2][3][4] Normal [6][7][8][9][Full]  ↑
Auto Number
   Name        [            ]
   Level       [ 1 ]
TOC
   Doc Name    [            ]
```

(b) Custom sheet

Figure 4-13. Format and Custom sheets of the Component property sheets

You can see the property sheet for only one component at a time. If you choose **Props** when more than one component is selected, a message in the status line informs you of this.

The Format and Custom Sheets

The Format sheet allows you to define the name, margins, first indent, line spacing, alignment, default font, and attributes of a component.

The Custom sheet enables you to control hyphenation within the component. In addition, you can use this sheet to associate both an autonumber stream level and a table of contents document name with a component.

The Component Property Sheet Popup Menus

When you make changes to properties on the Component property sheets, you have several options for applying the changes. Using **Apply**, you can apply them to a single component; using **Global Apply**, you can apply them to the master and all components with the same name.

When you apply changes globally, you are asked to confirm. Figure 4-14 shows the Apply popup menu with the Global Apply submenu.

```
┌──────────────┬──┐
│ Cancel       │  │
│ Apply        │┌─────────┐
│ Global Apply ││ Confirm │
│              ││□Cancel  │
└──────────────┴─────────┘
```
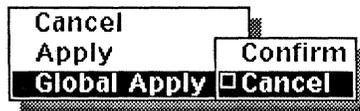
Figure 4-14. Apply popup menu with Global Apply submenu

Global Apply applies only those changes that you have made since you opened the Component property sheets, or since the last time you applied, globally applied, canceled or reset changes.

Cancel on the main *Apply* menu cancels any changes you have made since the last time you applied or globally applied changes.

Cancel on the *Global Apply* submenu cancels only the **Global Apply** command. If you want to cancel the change(s) you made to the property sheet, use **Cancel** on the main Apply menu.

Once you apply changes and then hold down the middle button, you see the *Close* menu. Figure 4-15 shows the Close menu with the Unify submenu.

```
┌───────┬──┐
│ Reset │  │
│ Close │┌─────────┐
│ Unify ││ Confirm │
│       ││□Cancel  │
└───────┴─────────┘
```
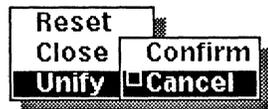
Figure 4-15. Close popup menu with Unify submenu

Unify on the Close menu is a more powerful command than **Global Apply** and also requires confirmation. **Unify** affects *all* properties (not just those you have changed) of all components that have the same name as the component whose property sheet is open. This command makes the properties of all instances identical to *all* the properties shown in the open property sheet.

Before using **Global Apply** and **Unify**, make sure you understand their distinct functions and know what effect the properties you are applying will have on all of the components involved. *Both of these commands update the definition of a component master.*

Reset on the Close menu resets *all* of the properties you have changed on *any* of the four Component property sheets to what they were when you opened the sheets. When you use **Reset**, this message appears in the status line: *Please apply or global apply to reset the component*. This command does not affect properties of other instances of the same component.

If you have *globally* applied any changes, you must use **Global Apply** in order to reset all the components. However, be sure that you do not use **Global Apply** after resetting a property that you have applied to only one component. If you use **Global Apply** in this case, you apply *all* of the properties on that sheet — not just the one you reset — to all of the components of that name in the document.

Reset resets the *properties* of a component, not its *content*. In most cases, this has no serious implications. If, however, you eliminate a distinction between the default text and a special font within it through a property change, that distinction will not be restored by **Reset**. For example, if you have a formula in *Math A* within a *Classic* component and you then accidentally change the component font to *Math A*, all of the text, including the formula, will be reset to *Classic* when you reset the properties.

Component Properties on the Format Sheet

The Name Box

In the *Name* box, you can type the name of a new or existing component. A component name can be up to 9 characters in length and be composed of any combination of letters, numbers, or other characters.

Each time you open a Component property sheet, type a unique name into the *Name* box, and apply the change, you create a new component master. The name of the new component master is then added to the list of component master names used by the **Create**, **Select**, and **Find** commands (Component Location menu) and the **Change** and **Select** commands (Component Selected menu). Component names appear on the list in a sorted order — if all the names contained nothing but alphabetic characters, the list would be alphabetical.

When you use numbers in component names, you should remember that the system sorts the names in the list character by character, with digits coming before alphabetic characters. This means that *title13* will be above *title3* in the list of component names and *10para* will be above *1para*. If you have a long list of component names, you may want to make their order easier to use by renaming *title3, title03* and renaming *1para, 01para*. Figure 4-16 shows three component lists. Figure 4-16c is probably the easiest one to use.

Create	▫10list		Create	▫01list		Create	▫figure
Paste	1list		Paste	01steps		Paste	list01
Select	1steps		Select	02list		Select	list02
Find	20steps		Find	10list		Find	list10
Join	2list		Join	20steps		Join	paragraph
	figure			figure			steps01
	paragraph			paragraph			steps20
	title			title			title
	title1			title01			title01
	title13			title02			title02
	title2			title03			title03
	title3			title13			title13
(a)			(b)			(c)	

Figure 4-16. Software-collated component lists

If you change a component name to the name of an existing component, the software automatically inserts the other properties of the existing component into the property sheet for you as soon as you confirm the name with a TAB or RETURN.

Margins

On the Document Page property sheet, you establish the margins for the entire document. On the *Format* sheet of the *Component* property sheets, you establish the margins for an individual component. The margin settings are mostly self-explanatory, but you should keep in mind the following points:

- The left component margin is the distance from the left page margin to the left edge of the component, and the right component margin is the distance from the right page margin to the right edge of the component. When they are set to *0,* the margins of the component are the same as the margins of the page. In this *Reference Manual,* for example, most of the components have a left margin of *0,* but the component you are reading has a left margin of *1.5* inches.

- Where components meet on the same page, the values for the bottom margin of one component and the top margin of the next are added together. However, the actual space between components may vary if the *Vertical Justification* property on the Document Page property sheet is **On.**

- Top component margins have no effect at the top of a page and bottom component margins have no effect at the bottom of a page. These **soft margins** above and below components prevent misalignment of text at the top of pages throughout a document and of text at the bottom of vertically justified pages.

Vertical justification and the behavior of top and bottom margins under different circumstances is explained in more detail in the chapter *Page Design*.

First Indent

Sometimes, you will want the first line of a certain kind of component indented. With typewriters, you set a tab for the indentation and then have to hit the tab key at the beginning of each paragraph. *First Indent* makes it possible for you to set an indentation and then let the software create it automatically at the beginning of a component.

In a paragraph having the width and type size of this paragraph, a setting of *.25* inches gives a very pleasing indentation, but you can set the indentation of the first line to any width that will leave at least one word on the first line. Typographers often use the **em space** in the font the paragraph is in to produce a first indent.

You can set a *negative* indentation when you want the first line to extend to the left of the other lines in a component (this is often called a **hanging indent**). The bulleted list above is an example of a hanging indent; the bullet *hangs* and the text is left aligned. To use negative indentation, you can set *First Indent* to any value up to the combined values of the left margin of the component and the left margin of the page (i.e., if the left margin of the component is *.25* inches and the left margin of the page is *1* inch, you can make a negative *First Indent* of between zero and *–1.25* inches). If you create a **Left** tab setting at 0, you will be able to use the TAB key to align the first line of a component with a negative first indent to succeeding lines in that component.

Line Spacing

Line Spacing determines the distance between the lines of type in a component. The default setting for every point size is *1.31* lines. Because *Line Spacing* is proportional to point size, this setting has different meanings for different type sizes. When *Vertical Justification* on the Document Page property sheet is **On**, line spacing may vary from the specified amount.

For point-to-inch conversions, please refer to the document named *LeadingToLineSpacing* in the *TextAids* folder in the *Documentation* drawer on your desktop.

Alignment

The *Alignment* settings allow you to change the horizontal alignment of components. There are four possibilities:

- **Flush Left,** which lines up the component on its left margin and leaves a ragged right margin
- **Flush Left** and **Flush Right,** which justifies the text between the left and the right margins; this is the only setting that allows spaces between words to expand
- **Centered,** which centers each line of the text of a component between the left and right margins (when you center a component, the *First Indent* must be *0,* or the first line will not be centered correctly)
- **Flush Right,** which lines up the component on its right margin and leaves a ragged left margin

You cannot insert tabs into center- or right-aligned components.

Font

You can change the family and size of a component's default font and also make it roman, bold, italic and, in some cases, bold italic. Changes to two or more of these properties at once are handled independently.

When a type family is changed, all words in that component that were of the original family are changed appropriately. For example, if the text contains Classic roman, italic, and bold and *Family* is changed to **Modern,** all Classic text, not just the roman characters, will be changed to its Modern counterpart.

When type size is changed, all words that were of the previous default size in that component are changed to the new size regardless of their typeface. Typeface and size of other sizes is unchanged.

The **Bold** and **Italic** settings can be turned *on* or *off* by pointing into the appropriate box and clicking the left button. If you have purchased the typesetter option or certain optional fonts, you can select **Bold** and **Italic** simultaneously.

The Font properties unique to the Interleaf typesetter option are discussed in the typesetting manual. For details on optional fonts see the *Interleaf Typeface Catalog.*

Using the **Bold** and **Italic** boxes on a Component property sheet, you can choose the default type style for that component: bold, italic, roman, or bold italic (optional). Changes on the property sheet affect one or more of these styles in text. For example, when you change the default from roman (bold and italic both unselected) to bold, all roman type becomes bold and any bold type becomes roman. All italic and bold-italic type remain unchanged. Figure 4-17 illustrates how

changes to the Component property sheet bold/italic settings affect the text of a component.

If the Component property sheet setting is changed in either direction between . . .	Roman and **Bold**	Roman and *Italic*	**Bold** and *Italic*	Roman and ***Bold Italic***	***Bold Italic*** and *Italic*	***Bold Italic*** and **Bold**
all Roman text becomes/stays	**Bold**	*Italic*	Roman	***Bold Italic***	Roman	Roman
all **Bold** text becomes/stays	Roman	**Bold**	*Italic*	**Bold**	**Bold**	***Bold Italic***
all *Italic* text becomes/stays	*Italic*	Roman	**Bold**	*Italic*	***Bold Italic***	*Italic*
all ***Bold Italic*** text becomes/stays	***Bold Italic***	***Bold Italic***	***Bold Italic***	Roman	*Italic*	**Bold**

Figure 4-17. Text changes when bold/italic properties change

Keep in mind that changes in font made on the Component property sheet always change the text in a component in the same way, no matter what font changes you have made using the Text Selected menu. For example, if you have used the menu commands to change all of the text in a component from its roman default to italic and then decide to change the default on the property sheet to italic, all of the text will become roman.

Not all keys have been assigned characters in all fonts. If a character that is displayed in a component is not available when you change fonts on the Component property sheet, that character is displayed on the screen as an upside-down question mark.

Attributes

The *Attributes* boxes allow you to make all the text within the component appear with *strikethrough* and/or *underlining*, and to mark the entire component with *revision bars*. You can turn these features on individually or simultaneously.

The following is an example of strikethrough *On: ~~strikethrough is on~~*. <u>This line is an example of text with underlining *On*</u>. The vertical black line displayed in the margin is a revision bar.

Like font characteristics, strikethrough and underlining function on a character-by-character basis; revision bars operate on a line-by-line basis. If you turn any of these properties **On** for the entire component and certain words or lines in the component already have the same properties on, the properties will be turned off for those words or lines.

Attributes are discussed in the chapter *Document Management*.

Component Properties on the Custom Sheet

Hyphenation

If the *Hyphenation* box on the Custom sheet of the Document Page property sheet is turned **On**, you can change the way hyphenation works in a particular component by changing the setting on its Custom sheet. See the chapter *Spell Checking and Hyphenation* for details.

Autonumber

The *Autonumber* boxes enable you to associate one level of an autonumber stream with a component so that when created, each instance of that component begins with an autonumber token. To do this, type the name of a *defined* stream in the *Name* box. If you want a stream level other than 1, which is the default, type the number of a *defined* level within the *Level* box. When you globally apply the information, an autonumber token will automatically be inserted at the beginning of each existing component of that type and at the beginning of each new instance you create thereafter.

For more information, see the chapter *Autonumbering and Referencing*.

Table of Contents

The *TOC Doc Name* box must be used in conjunction with the *book* to create a table of contents for a document. See the chapter *Large Documents in Books* for details on how to use this feature.

The Text Caret

Like the component caret in the component bar, the text caret in text represents a location.

When you start typing, characters are inserted at the position of the text caret. They are in the font and have the attributes displayed in the Font and Attributes boxes of the document header. As you type, the caret moves to the right across a line and then to the beginning of the next line when it reaches the right margin of the component, always moving ahead of the last character typed.

When you choose any of the commands on the Text Location menu, the position of the text caret determines where the action occurs.

When you are selecting text, the text caret represents the starting location for the selection. The selection can go forward or backward in text or in both directions, but the text caret stays where it is first located.

You can move the text caret around in text in several ways:

- When you hold down the left button and move the mouse, the text caret merges with the mouse cursor. This is a quick way to move the text caret and an accurate way to position it. When you move the caret this way, your document scrolls automatically when the mouse cursor reaches a window boundary.

- You can move the mouse cursor to the location where you want the text caret and *click* the left button to bring the text caret there.

- You can use the *arrow keys* and other keyboard commands.

- The **Go To** command on the Page Pulldown menu moves the caret to the top of the page you want to go to.

The Relationship of the Text Caret to the Component Caret

When the component caret is visible, the text caret is *always* in the text of the component with the caret. If you move the caret to another component, the text caret moves to the beginning of that component. If you move the text caret from the text of one component into the text of another component, the component caret moves, too. In this case, the caret will be above the component name if it was above the other component name and below if it was below.

When a component is *selected* (highlighted) the text caret is not necessarily in that component. You can move the text caret independently.

Vertically Centering the Text Caret Location in the Document Window

The **Center** command on the Text Location menu scrolls the document whatever amount is necessary to center the line the text caret is on in the document window. This command is very useful when you are editing near the top or the bottom of a window and would like to see the surrounding text, both above and below the caret.

There is another circumstance in which it is also very useful. Except for **Go To**, the commands on the Page pulldown menu do not change the position of the text caret. This is also true when you use the scroll bar for scrolling. These scrolling methods are particularly appropriate when you want to look through your document but do not want to move the text caret from its location. When you have finished browsing, you can execute the **Center** command, and the page with the text caret on it reappears with the text caret centered in the window.

Selecting Text

To select text, you use both the left and the right buttons. The left button positions the text caret, which acts as an anchor for the selection. The right button does the selecting and the deselecting.

Text can be selected forward, backward, or in both directions from the text caret. It can also be selected across component boundaries. But all the text that is selected at one time must be connected — no unselected text can interrupt it.

Deselecting Text

You can deselect text in a number of ways. The following actions deselect *all* of the selected text:

- editing text in any way (inserting or deleting a character at the keyboard; clearing or setting hyphenation points using the Hyphenation submenu, adding a word to the desktop dictionary using the **Add to Dictionary** command on the Spelling submenu)
- any action that moves the text caret
- executing the **Deselect** command

It is also possible to deselect just a portion of selected text. There are two methods for doing this, both of which are described in the chapter *Basic Procedures*.

Changing Fonts

Most of the commands on the Fonts submenus are available on both the Text Location and Text Selected menus (Figure 4-18). If no text is selected and you execute a **Fonts** command, the font at the text caret location changes. If you first select text and then execute a **Fonts** command, the font(s) of the selected text are changed.

```
┌─────────────────────────────────┐  ┌──────────────────────────────────┐
│ Create │ Last              │     │  │ Cut      │ Current           │    │
│ Paste  │ ◻Pick Up          │     │  │ Copy     │ Last              │    │
│ Fonts  │ Family          → │     │  │ Fonts    │ ◻Pick Up          │    │
│ Center │ Size            → │     │  │ Props    │ Family          → │    │
│ Misc   │ Bold            → │     │  │ Deselect │ Size            → │    │
│        │ Italic          → │     │  │ Misc     │ Bold            → │    │
│        │ Attributes →      │     │  │          │ Italic          → │    │
│                               │  │          │ Attributes →      │    │
│                               │  │                                    │
│ Text Location Fonts Submenu   │  │ Text Selected Fonts Submenu        │
└─────────────────────────────────┘  └──────────────────────────────────┘
```
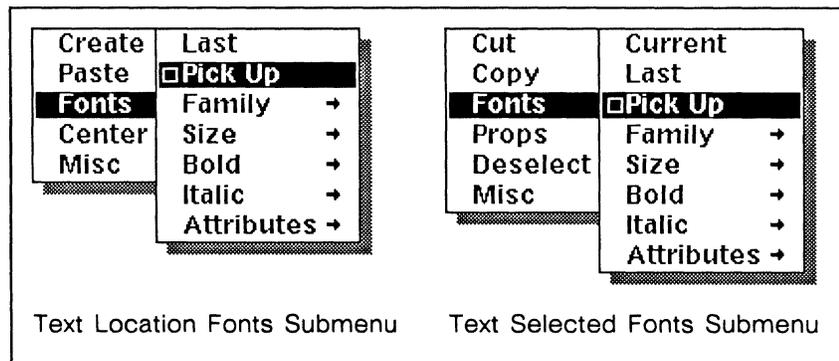
Figure 4-18. Text Location and Text Selected Fonts submenus

For convenience, some of these commands are also available from the keyboard. For example, to toggle Bold you can press **F4** and to toggle Italic press **F5**. See the chapter *The Keyboard* for details.

Current

The **Current** command changes all *selected* text to the font displayed in the Font box and applies the attributes shown in the Attributes box located in the document header. In most cases, the current font and attributes are the same as those at the position of the text caret. **Current** can be useful if you have a number of different fonts in a region and want to make all of the text the same font. This command is available on the Text Selected menu only.

Last Font

Last makes it possible for you to switch fonts at the caret location (and of selected text) and then switch back to the most recent font you used, without going beyond the first submenu. Since there are commands that toggle between bold and roman and between italic and roman on the same submenu, this command is most useful when you are alternating fonts of a different family or size. The caret and all selected text take on the family, size, attributes and bold/italic characteristics of the last font when you execute this command.

Text Processing

Pickup Font

Pickup makes it possible for you to switch easily the font of the text caret and of selected text to any font visible in text in the document window.

> If your document window contains an open microdocument, you cannot use **Pickup** to change to the font of text displayed outside the microdocument; likewise, from the main document you cannot use this command to switch to a font displayed in a microdocument.

To execute the pickup command:

    ✔ Position the *text caret* where you want the new font.

  or ✔ Select the text you want to change.

    ✔ Point the *mouse cursor* at some text that is in the font you want to use.

    ✔ Hold down the **middle** button, and release it when **Pickup** is in reverse video.

        Pickup *is the default on this menu unless you have done something to change it temporarily.*

Caution: Do not confuse changes made using the Text Selected menu with changes to a component's default font. With one exception, font changes to selected text do not affect the component default font. The one exception occurs when you select all text in a document. In this case, a font change on the Text Selected menu also changes the default font for every component.

Font Family

On both of the Text popup menus, **Family** leads to a submenu with the listing of the available font families.

On the Text Location submenu each family has a submenu of available sizes. You can, therefore, choose a family and a size in one action. The default on the Size submenu is the size closest to the one displayed in the Font box in the document header (for example, if the font is *Classic 24-point,* and you choose *Typewriter,* the default size will be *12* because that is the closest size in that typeface).

For selected text, there is no Size submenu on the Family submenu.

When a type family is changed, all selected text is changed appropriately. For instance, if the selected text contains Classic roman, italic and bold as well as

4-30   *Release 3.0*

Typewriter, and the font family is changed to Modern, all of the typefaces present in the selected text will be changed to their Modern counterparts.

You have received a set of standard font families with the publishing software. If you purchase additional fonts, the names of those fonts will also appear on the Family submenu. For information about optional fonts, see the *Interleaf Typeface Catalog*.

Font Size

Size is the option to choose if you want to change the *size* of the font at the caret position or of selected text. The Size submenu varies depending on whether you have selected text or not. The Text Location Size submenu shows the sizes available for the font at the caret location.

The Text Selected Size submenu shows the sizes available for the font with the most options; some of the sizes shown are not available for all of the families. When you select text and change the font size, all selected text is changed to the new size regardless of bold/italic characteristics. However, if the new size is not available in the selected font, the size will be changed to the next smaller size available for that typeface. If the next smaller size is identical to the current type size, it will be changed to the next larger size available for that typeface.

When no text is selected, **Larger** and **Smaller** change the font at the text caret one size in either direction from the size displayed in the Font box of the document header. When you select text and use **Larger** or **Smaller**, each size within the region of selected text is changed independently to its own next larger or smaller size. **Larger** is the default unless you have changed to another setting on the submenu.

Bold and Italic Toggle

Regardless of whether text is selected, both the **Bold** and **Italic** commands toggle the current font. The result of the toggle depends on whether or not *bold italic* is available for the font at the text caret or for the selected text. Bold italic is available with some of the optional fonts but not with the standard fonts.

First consider how **Bold** and **Italic** work on systems that have only the standard fonts.

If no text is selected and the font designated in the Font box in the document header is *not* bold, releasing the mouse button when **Bold** is in reverse video will change the font at the caret position to bold (if bold is available in that family and size). If the font designated in the Font box *is* bold, releasing the mouse button when **Bold** is in reverse video will change the font to roman.

The **Italic** command works the same way as bold when no text is selected.

When text is selected, the Bold and Italic submenus include **On**, **Off**, and **Toggle** commands. This allows you to alter selected text in distinct ways. When you use the **On** command, you turn on the property for the entire region, making it all bold or all italic. When you use the **Off** command, any text in the selected region with the property you turned off becomes roman. Turning **Off** a property that is not already on — for example, using **Bold Off** in a region with only roman and italic type — will have no effect.

When you use the **Toggle** command on the Bold submenu, you make all bold type roman and all roman or italic type bold. Likewise, when you use the **Toggle** command on the Italic submenu, you make all italic type roman and all roman or bold type italic. Figure 4-19 summarizes bold/italic changes made with these commands. Note the difference in effect between these changes and bold/italic changes made on the Component property sheet.

If the command executed on the Text Selected menu is . . .	**Bold** Toggle	**Bold** On	**Bold** Off	*Italic* Toggle	*Italic* On	*Italic* Off
all Roman text becomes/stays	**Bold**	**Bold**	Roman	*Italic*	*Italic*	Roman
all **Bold** text becomes/stays	Roman	**Bold**	Roman	*Italic*	*Italic*	**Bold**
all *Italic* text becomes/stays	**Bold**	**Bold**	*Italic*	Roman	*Italic*	Roman

Figure 4-19. Effect of Bold/Italic commands on Text Selected menu (does not apply to systems using typesetter and certain optional fonts)

If you have purchased the typesetter option or one of the optional fonts, the *bold-italic* typeface may be available to you. If you want bold italic, you can sequentially choose **Bold** and **Italic** on the Fonts submenu or press, either sequentially or simultaneously, both the Bold and Italic keys on the keyboard. Figure 4-20 summarizes the effect of bold/italic changes on selected text for which bold-italic is available.

If the command executed on the Text Selected menu is . . .	**Bold** Toggle	**Bold** On	**Bold** Off	*Italic* Toggle	*Italic* On	*Italic* Off
all Roman text becomes/stays	**Bold**	**Bold**	Roman	*Italic*	*Italic*	Roman
all **Bold** text becomes/stays	Roman	**Bold**	Roman	***Bold Italic***	***Bold Italic***	**Bold**
all *Italic* text becomes/stays	***Bold Italic***	***Bold Italic***	*Italic*	Roman	*Italic*	Roman
all ***Bold Italic*** text becomes/stays	*Italic*	***Bold Italic***	*Italic*	**Bold**	***Bold Italic***	**Bold**

Figure 4-20. Effect of Bold/Italic commands on Text Selected menu (for systems using typesetter and certain optional fonts)

Capitalizing Text

Sometimes you will find that you must change the way a word is capitalized within a document. To minimize the time and chance of error involved in retyping, the software provides commands so that you can change selected text to all capital letters, initial capitals on each word, or no capitals. Figure 4-21 shows the Capitals submenu.
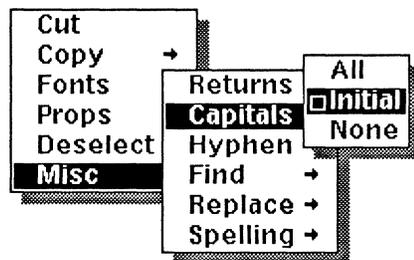
Figure 4-21. Capitals submenu

To change words in selected text to all capitals, no capitals, or initial capitals only:

✔ Select the region of text you want to change.

▯▮▯ Execute the command of your choice on the `Misc ➜` `Capitals ➜` submenu of the Text Selected menu.

This set of commands affects letters only — it does not change numbers or symbols to their related upper- or lowercase characters. There are several other characteristics of these commands that you should be aware of.

- In a compound word divided by a hard hyphen(-), minus sign or en dash (–), em dash (—), ampersand (&), apostrophe ('), or backslash (\), **Initial** capitalizes only the first letter in the first word of the compound. When a standard slash (/) divides the compound word, the initial letters of both parts are capitalized by the **Initial** command.

- If only part of a word is highlighted and you execute the **All** command, only the highlighted letters in that word are capitalized. If you execute **Initial** without having the initial letter in a word selected, no letter in that word will be capitalized.

Cut, Copy, and Paste in Documents

There are **Cut** and **Copy** commands on both the Component Selected and the Text Selected menus, and **Paste** commands on both the Component Location and Text Location menus (Figure 4-22).
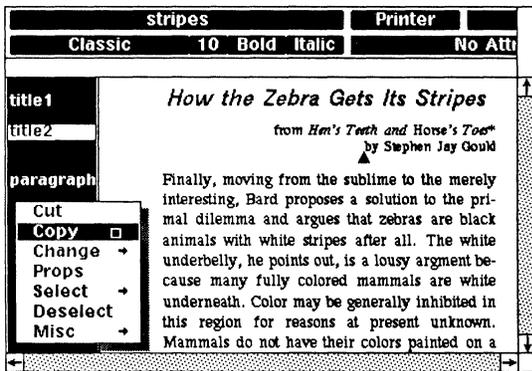
Component Selected menu
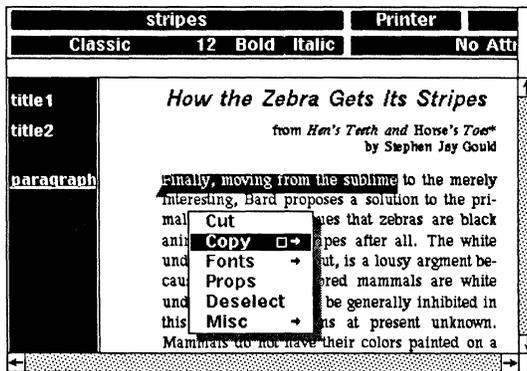
Text Selected menu
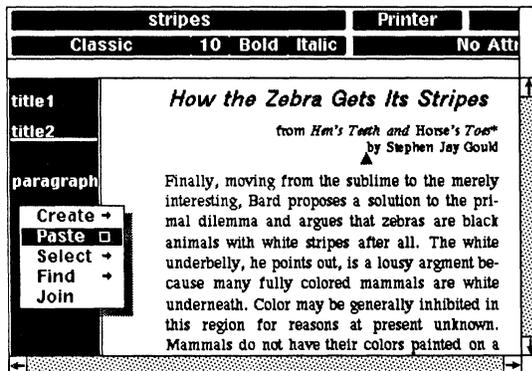
Component Location menu
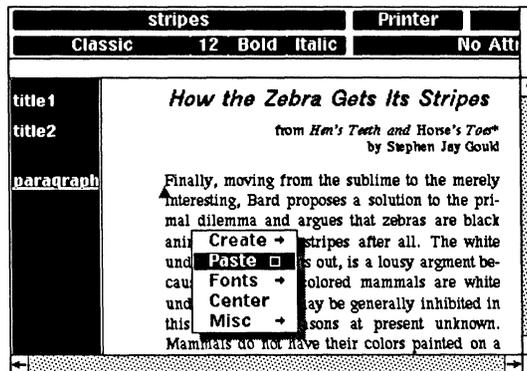
Text Location menu

Figure 4-22. Component bar and text popup menus

There is interaction among these commands — for example, you can cut components and paste them as text or copy text and paste it as a component.

The Cut and Copy Commands

When you execute the **Cut** command, the object is removed from its original location. **Copy** leaves the document from which the command was executed unchanged.

When you **Cut** or **Copy** components or text, the cut or copied material becomes the contents of a document represented on the clipboard by an icon named *from 'documentname'* (where the actual name of the document is substituted for *'documentname'*).

Usually you can have only one icon named *from 'documentname'* on the clipboard. Whenever you cut or copy components or text from an open document, the contents of *from 'documentname'* on your clipboard are overwritten. By cutting *one word* of text you may be overwriting the *20 components* you previously copied. On those rare occasions when there is more than one *from 'documentname'* icon on the clipboard, *all* are vulnerable to overwriting.

Since the contents of *from 'documentname'* on your clipboard is so vulnerable, you must remember to paste material you want to save.

- If you have cut something in order to move it, *immediately* paste into your document or onto your desktop. Otherwise, what is in the document on the *clipboard* will be overwritten the next time you cut or copy text or components.
- If you have cut something that you want to discard, you can forget about it. The next time you **Cut** or **Copy** components or text it will be overwritten.

If you paste a *from 'documentname'* icon on your desktop, you can rename it. But if you put it back on the clipboard as *from 'documentname'*, it is vulnerable to overwriting.

You can select and then cut or copy any number of components at a time. These can be scattered throughout a document or be together in a series.

You can cut or copy only one uninterrupted block of text at a time, but the block can be as long as you like.

Copy as the Default Command

The software takes advantage of the capacity to change defaults temporarily to set up combined **Copy** and **Paste** operations. When you copy components, **Copy** becomes the default on the Component Selected menu. Similarly, when you copy text, **Copy** becomes the default on the Text Selected menu. So that when you **Copy** and then **Paste**, **Copy** is the next action available to you.

The interaction between **Copy** and **Paste** is useful when you have something you want in a number of places. You can copy it and paste it as many times in a row as necessary.

There is no similar interaction between **Cut** and **Paste**. If you **Cut** and **Paste**, the default command on the Component Selected menu or the Text Selected menu returns to **Props**. Figure 4-23 shows the combinations that occur with **Cut** and **Copy** on the Component Selected menu and the Text Selected menu.

In the component bar and in text:

If you choose **Cut** the defaults that follow are **Paste** then **Props**

If you choose **Copy** the defaults that follow are **Paste** then **Copy** then **Paste...**

Figure 4-23. How Cut and Copy work with Paste

Cut as the Default

Usually, when you want to cut a group of components, the easiest method is to select them all and execute a **Cut** command. However, you may sometimes want to cut components one or two at a time. The stickiness of the **Cut** command makes a series of cuts easy to accomplish. Once you choose **Cut**, you can select the component(s) you want to cut and click the middle button.

Cut remains the default on the Component Selected menu until you either choose another command on one of the component bar menus or slide the cursor off the menu to reset the default to **Props**.

The **Cut** command on the Text Selected menu acts in exactly the same way. The stickiness of the **Cut** command is even more useful in text than in the component bar because you cannot select multiple text blocks at once as you can select multiple scattered components.

When you cut objects from an open frame, they become the contents of a diagramming object named *from 'documentname'* on the clipboard. See the chapter *Modifying Diagrams* for a discussion of diagramming objects on the clipboard.

The Paste Commands

Whether you cut or copy text only, components, or entire documents, the resulting document on the clipboard contains enough page and component information to be pasted as text, as a component, or as a document. The **Paste** commands on

the Component and Text Location menus, and on the Desktop Nothing Selected menu can all be used to paste any object from the clipboard.

Pasting Text as Text

You can paste text as text within a single document, or you can cut or copy text in one document and paste it in another. In either case, you simply position the text caret where you want to paste the text and execute the command.

If what you are pasting consists of text from more than one component, you will get a stickup to confirm the execution of the **Paste** command (see Figure 4-24). For example, if you have selected and copied the last sentence in a *paragraph* component and the first sentence in the *list* component following it and are now pasting the selection in text, the system warns you that you will be losing component information if you proceed. This loss of component information does not affect the font(s) of the pasted text.

> You are pasting your selection as text which will destroy the component information. Please confirm.
>
> [Confirm] Cancel
> ↑
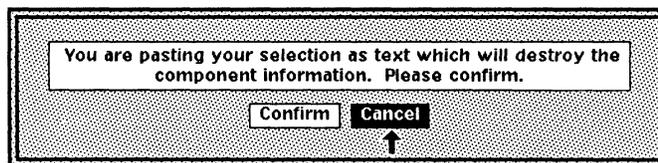
Figure 4-24. Loss of Component Information stickup

Pasting Text as Components

When you paste text in the component bar, the new component assumes the name and other properties of the component the text was cut or copied from. For example, if you have copied four words from a *paragraph* component and you execute a **Paste** command in the component bar, a *paragraph* component with the four words in it will be inserted.

If you have copied four words from a *paragraph* component along with another four from the *list* component that follows it and you execute a **Paste** command in the component bar, a *paragraph* component with the four words in it and a *list* component with the four words in it will be inserted, and all component information will be preserved.

Pasting Components as Components

Pasting components as components is the way to move one or more components from one place to another in the same document or from one document to another. It is also *one* way to insert an entire document into another document.

When you paste components as components, they are inserted at the component caret position in the order in which they originally occurred. They will be pasted contiguously even if they were separated by unselected components before the cut or copy.

Pasting Components or Documents as Text

You will not often want to paste components as text because the pasted compo-
nents lose their identity as components; that is, if you paste 20 components of
different names into the text of a *paragraph* component, the 20 components are
melded together and take on the characteristics of the *paragraph* component.

Because of the potential loss of component information, you are asked to confirm
on a stickup that you really want to execute the **Paste** command if what you are
pasting has more than one component in it (Figure 4-24).

You can also paste *documents* as text, but doing so is seldom a good idea. As
with components, pasting documents as text requires confirmation on a stickup
menu.

Pasting Documents as Components

You can insert one document into another by pasting the document as compo-
nents. To do this, cut or copy the document icon to be inserted (or make sure the
document icon is selected on the clipboard), position the mouse cursor in the
component bar of the document in which you want to paste, and execute the
Paste command.

When you paste a document icon into an open document as components, the
components of the document you are pasting are inserted at the component caret
position in the order in which they originally occurred. The page properties of the
pasted document are lost.

The document in which the **Paste** command is issued is the
document whose page properties, such as page margin and
header style, prevail. When you are going to combine two
documents, be sure that you paste into the document that
has the page properties you want. Please refer to the chap-
ter *Document Templates* for details about this process.

Paging and Scrolling

The **Page** box in the document header gives you page information about your document, and the Page pulldown menu gives you access to several page-related commands.

By looking at the Page box, you can learn the current page number, the number of pages in the document, and what the first page number is.

The current page number that is shown in the box is the number of the page that fills more than half the document window. When the division is about 50/50, it will be very difficult for you to judge this with your eyes.

Figure 4-25a shows the way the box looks when a document starts with page *1*, and Figure 4-25b shows the way the box looks when a document begins with another number (in this case, *5*).

```
┌─────────────────────────┬──────────────────────────┐
│ ■■ Page 8 of 9 ■■    ·  │ ■ Page 12 of 5 to 13 ■   │
│           (a)           │            (b)           │
└─────────────────────────┴──────────────────────────┘
```

Figure 4-25. Page boxes for two different documents with different starting page numbers

Starting page number is one of the properties entered on the Document Page property sheet. You gain access to this sheet by executing the **Properties** command on the Page pulldown menu (Figure 4-26).

```
■■■ Page 8 of 9 ■■■
□ Properties
  Next
  Previous
  Go To
  First
  Last
  Current
```
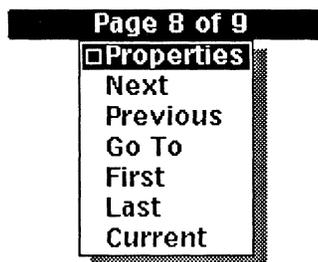
Figure 4-26. Page pulldown menu

Properties is the default on the menu, but when executed **Next, Previous,** and **Go To** become defaults until you execute another pulldown command or move the mouse cursor outside the document header. When you use **First, Last,** or **Current, Properties** remains the default since **First, Last,** and **Current** are absolute rather than relative or variable destinations.

Other than **Properties,** the commands on this menu represent ways to move through a document in a measured and systematic way. They always take you to the top of the specified page.

When you execute the **Go To** command, the component and text carets move to the beginning of the page you specify. When you execute the other commands on this menu, the carets do not move from the original line or page.

Next Page

The **Next** command scrolls the document to the next page as defined by the page number registered in the Page box. When you are on the last page of a document, you get the message *This is the last page* in the status line.

Previous Page

The **Previous** command scrolls the document to the previous page as defined by the current page number that is registered in the Page box. When you are on the first page of a document, you get a message in the status line *This is the first page*.

Go To Page

The **Go To** command produces a stickup menu on which you type the number of the page you want to move to. If the number is greater than the highest page number in the document, this command goes to the last page in the document. The text caret moves to the beginning of the first component on the specified page.

First Page

The **First** command scrolls the document to the first page of the document.

Last Page

The **Last** command scrolls the document to the last page of the document.

Current Page

The **Current** command scrolls the document, so that the current page as defined by the page number registered in the Page box begins at the top of the document window.

A Summary of Scrolling Methods

The following scrolling methods move *both* the component caret *and* the text caret:

- using the **Find** command on the Component Location menu
- using the **Go To** command on the Page pulldown menu

- using any of the **Go To Page, Next,** or **Previous** function key commands described in *The Keyboard* chapter
- using the arrow keys
- holding down the left button and moving the mouse cursor in the component bar
- if no components are selected, holding down the left button and moving the mouse cursor in text (if any components are selected, they remain selected, so the caret does not move with the text caret)

The following scrolling methods move *neither* the component caret *nor* the text caret:

- using the scroll bars
- using the scrolling commands on the Page pulldown menu (except the **Go To** command)

Setting Tabs

There are four kinds of tab settings in Interleaf publishing software documents:

- Left
- Center
- Right
- Decimal

You can insert tabs in any component that is either **left-aligned** or both **left** and **right-aligned**. Because tabs are set with reference to the left margin of a *component*, not the left margin of the page, they have meaning only in components that have a left alignment. Although you cannot insert *tab characters* in the text of a **centered** or **right-aligned** component, centered and right-aligned components can have *tab settings* on the Tab sheet of the Component property sheets.

If you already have tabs in the text of a left-aligned component and then change it to centered or right-aligned you will get an error message requesting you to delete the tabs or change the alignment, but you will be able to make the alignment change. The software will not accept new tabs that you try to insert in the text of a centered or right-aligned component.

When you are setting tabs, consider the left margin of the component to be at tab position *0* inches. Figure 4-27 shows where text appears when **Left, Center, Right,** and **Decimal** tabs are set at *2* inches (that is, 2 inches from the left margin of the component). The arrowhead shows where the tab is set in each instance.

```
┌─────────────────────────────────────────────┐
│                 Left tab at 2 inches         │
│                 ▲                            │
│          Center tab at 2 inches              │
│                 ▲                            │
│   Right tab at 2 inches                      │
│                 ▲                            │
│       Decimal tab with . at 2 inches         │
│                 ▲                            │
└─────────────────────────────────────────────┘
```
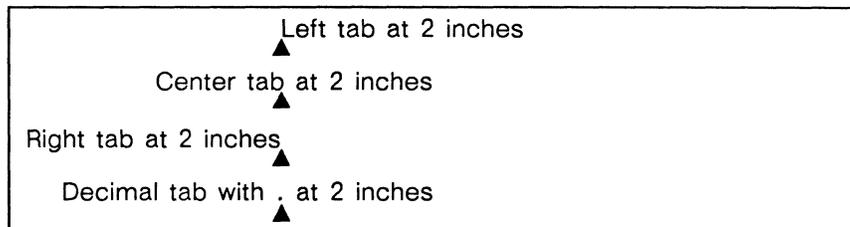
Figure 4-27. Comparison of tab settings

Note that the tab is set in the same place whether it is **Left**, **Center**, **Right**, or **Decimal**. The difference is in the relationship of the text to the tab setting.

- When you use a **Left** tab setting, text grows to the right of the tab setting.

- When you use a **Center** tab setting, the text is centered around the tab setting (in Figure 4-27, the *C* of *Center* and the *s* of *inches* are equidistant from the tab setting).

- When you use a **Right** tab setting, text grows to the left of the tab setting.

- When you use a **Decimal** tab setting, text grows to the left until a decimal point is encountered. From the decimal point on, text grows to the right, so that the decimal points in a column line up properly. In other words, a decimal tab behaves like a right tab up to the decimal point and like a left tab from the decimal point on. If no decimal occurs in the text, a decimal tab behaves like a right tab. If the first character typed at the tab stop is a decimal, a decimal tab behaves like a left tab. (A close parenthesis acts like a decimal point in relation to a decimal tab.)

The Tab Sheet of the Component Property Sheet

Tabs for each component can be set on the Tab sheet, which is one of the Component property sheets. Figure 4-28 shows the Tab sheet with its default settings. As with any component property, tabs can be applied to one or all components with the same name.

You will find tabs useful in the following circumstances:

- setting up numbered lists
- making outlines
- making tables
- setting up table of contents templates

| Component Properties: | Format | **Tab** | Page | Custom |

Location	Tab Type				
0	inches	**Left**	Center	Right	Dec.	Remove
0.75	inches	**Left**	Center	Right	Dec.	Remove
1.50	inches	**Left**	Center	Right	Dec.	Remove
2.25	inches	**Left**	Center	Right	Dec.	Remove
3	inches	**Left**	Center	Right	Dec.	Remove
3.75	inches	**Left**	Center	Right	Dec.	Remove
4.50	inches	**Left**	Center	Right	Dec.	Remove
5.25	inches	**Left**	Center	Right	Dec.	Remove
6	inches	**Left**	Center	Right	Dec.	Remove
6.75	inches	**Left**	Center	Right	Dec.	Remove
7.50	inches	**Left**	Center	Right	Dec.	Remove
8.25	inches	**Left**	Center	Right	Dec.	Remove
9	inches	**Left**	Center	Right	Dec.	Remove
9.75	inches	**Left**	Center	Right	Dec.	Remove
10.50	inches	**Left**	Center	Right	Dec.	Remove
11.25	inches	**Left**	Center	Right	Dec.	Remove
12	inches	**Left**	Center	Right	Dec.	Remove
12.75	inches	**Left**	Center	Right	Dec.	Remove
13.50	inches	**Left**	Center	Right	Dec.	Remove
14.25	inches	**Left**	Center	Right	Dec.	Remove
15	inches	**Left**	Center	Right	Dec.	Remove
15.75	inches	**Left**	Center	Right	Dec.	Remove
16.50	inches	**Left**	Center	Right	Dec.	Remove
17.25	inches	**Left**	Center	Right	Dec.	Remove
18	inches	**Left**	Center	Right	Dec.	Remove
18.75	inches	**Left**	Center	Right	Dec.	Remove
19.50	inches	**Left**	Center	Right	Dec.	Remove
20.25	inches	**Left**	Center	Right	Dec.	Remove
21	inches	**Left**	Center	Right	Dec.	Remove
21.75	inches	**Left**	Center	Right	Dec.	Remove

Figure 4-28. Tab sheet of the Component property sheet

You can enter tab settings in any order (Figure 4-29a), and the system will reorder them by numerical value after you apply them (Figure 4-29b). You can have negative as well as positive tab settings. If you use a negative tab setting, you must include a tab setting at *0* inches, the left margin of the component.

To remove a tab setting, you click the left button in the **Remove** box on the line next to it. You need to remove only those settings you do not want to use and ones that fall within the range of the tabs you want. For example, in Figure 4-29a, new tabs are set at *0, 2, 4,* and *5.5* inches, so that only the settings between *0* and *5.5* inches have to be removed.

Component Properties:	Format	**Tab**	Page	Custom

Location		Tab Type				
0	inches	**Left**	Center	Right	Dec.	Remove
2	inches	**Left**	Center	Right	Dec.	Remove
4	inches	Left	**Center**	Right	Dec.	Remove
5.50	inches	Left	Center	**Right**	Dec.	Remove
3	inches	Left	Center	Right	Dec.	Remove
3.75	inches	Left	Center	Right	Dec.	**Remove**
4.50	inches	Left	Center	Right	Dec.	**Remove**
5.25	inches	Left	Center	Right	Dec.	**Remove**
6	inches	**Left**	Center	Right	Dec.	Remove
6.75	inches	**Left**	Center	Right	Dec.	Remove
7.50	inches	**Left**	Center	Right	Dec.	Remove
8.25	inches	**Left**	Center	Right	Dec.	Remove
9	inches	**Left**	Center	Right	Dec.	Remove
9.75	inches	**Left**	Center	Right	Dec.	Remove
10.50	inches	**Left**	Center	Right	Dec.	Remove
11.25	inches	**Left**	Center	Right	Dec.	Remove
12	inches	**Left**	Center	Right	Dec.	Remove
12.75	inches	**Left**	Center	Right	Dec.	Remove
13.50	inches	**Left**	Center	Right	Dec.	Remove
14.25	inches	**Left**	Center	Right	Dec.	Remove
15	inches	**Left**	Center	Right	Dec.	Remove
15.75	inches	**Left**	Center	Right	Dec.	Remove
16.50	inches	**Left**	Center	Right	Dec.	Remove
17.25	inches	**Left**	Center	Right	Dec.	Remove
18	inches	**Left**	Center	Right	Dec.	Remove
18.75	inches	**Left**	Center	Right	Dec.	Remove
19.50	inches	**Left**	Center	Right	Dec.	Remove
20.25	inches	**Left**	Center	Right	Dec.	Remove
21	inches	**Left**	Center	Right	Dec.	Remove
21.75	inches	**Left**	Center	Right	Dec.	Remove

(a)

Tab sheet with user's entries

Component Properties:	Format	**Tab**	Page	Custom

Location		Tab Type				
0	inches	**Left**	Center	Right	Dec.	Remove
2	inches	**Left**	Center	Right	Dec.	Remove
4	inches	Left	**Center**	Right	Dec.	Remove
5.50	inches	Left	Center	**Right**	Dec.	Remove
6	inches	**Left**	Center	Right	Dec.	Remove
6.75	inches	**Left**	Center	Right	Dec.	Remove
7.50	inches	**Left**	Center	Right	Dec.	Remove
8.25	inches	**Left**	Center	Right	Dec.	Remove
9	inches	**Left**	Center	Right	Dec.	Remove
9.75	inches	**Left**	Center	Right	Dec.	Remove
10.50	inches	**Left**	Center	Right	Dec.	Remove
11.25	inches	**Left**	Center	Right	Dec.	Remove
12	inches	**Left**	Center	Right	Dec.	Remove
12.75	inches	**Left**	Center	Right	Dec.	Remove
13.50	inches	**Left**	Center	Right	Dec.	Remove
14.25	inches	**Left**	Center	Right	Dec.	Remove
15	inches	**Left**	Center	Right	Dec.	Remove
15.75	inches	**Left**	Center	Right	Dec.	Remove
16.50	inches	**Left**	Center	Right	Dec.	Remove
17.25	inches	**Left**	Center	Right	Dec.	Remove
18	inches	**Left**	Center	Right	Dec.	Remove
18.75	inches	**Left**	Center	Right	Dec.	Remove
19.50	inches	**Left**	Center	Right	Dec.	Remove
20.25	inches	**Left**	Center	Right	Dec.	Remove
21	inches	**Left**	Center	Right	Dec.	Remove
21.75	inches	**Left**	Center	Right	Dec.	Remove
0	inches	Left	Center	Right	Dec.	**Remove**
0	inches	Left	Center	Right	Dec.	**Remove**
0	inches	Left	Center	Right	Dec.	**Remove**
0	inches	Left	Center	Right	Dec.	**Remove**

(b)

Tab sheet after user's entries are applied

Figure 4-29. Entering tab settings

Tab Appearance

On the Tabs submenu, you can see the different appearances tabs can have in a document (Figure 4-30).

Create	Masters	**Blank**
Paste	**Tabs**	
Fonts	Show	
Center	Split	
Misc	Find →	
	Replace →	
	Hyphen →	
	Spelling	
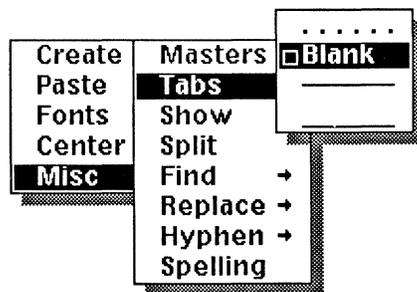
Figure 4-30. Tabs submenu

You can mix tab styles within the same document. The following example uses all four possibilities—blank, underline, line, and dots.

Here   is_____ an example of——————— four . . . . . kinds of tabs.

To accomplish this, all you need to do is choose what you want a particular tab to look like *before* you press the TAB key. If you do not like the way a tab looks, delete it, choose the appearance you want on the Tabs submenu, and then enter another tab.

Even if you are using blank tabs, you may want to see the tab settings on the screen while you are entering and editing text. The **Show** submenu on the Text Location menu makes it possible for you to choose whether or not arrows appear on your screen to represent blank tabs (Figure 4-31).

Please refer to the chapter *Setting Tabs* in the *Training Manual* for instructions on using tabs and on setting the way your tabs will look on the printed page.

Additional Features

The Show Submenu

On the Misc submenu of the Text Location menu you will find the **Show** command. All of the items on the Show submenu (Figure 4-31) are toggle settings.
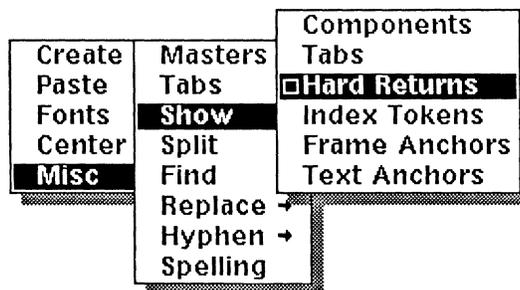
Figure 4-31. Show submenu

Each of these items can be set so that you see a symbol on the screen that represents it. None of these symbols is printable. The default in each case is *on*.

- If *Components* is on, the component bar with the names of the components is visible.
- If *Tabs* is on, blank *Tabs* are represented by dotted lines and arrowheads that indicate the type of tab.
- If *Hard Returns* is on, both *Hard Returns* (represented by ?) and *Force Justify* characters (represented by ↲) are visible.
- If *Index Tokens* is on, each index token is represented by ⅰ.

- If *Frame Anchors* is *on,* each frame anchor is represented by 回 (except for At Anchor frames, which cover the anchor symbol).

- If *Text Anchors* is *on,* each unused text string anchor in a frame is represented by ⊬ , ⊬⊢, or ⊣ and each empty microdocument is represented by a bounding box ⌐⌐⌐. These symbols are always visible in an open frame.

Returns in External Documents

When external documents □ are opened on your desktop, they often have RE-TURNS at the ends of lines. Sometimes you want these to remain and sometimes you do not.

Both the Component Selected and Text Selected menus have commands enabling you to delete or insert returns. On the Misc submenu of the Component selected menu you will find **Delete Returns** and **Insert Returns**; the Misc submenu of the Text Selected menu contains the **Returns** command with a submenu of **Delete** and **Insert**.

When you do not want the returns, select the components or text containing the unwanted returns and execute the appropriate **Delete** command. The hard returns are replaced by soft spaces.

With the **Insert** command, you can replace the soft spaces at software-generated line breaks with hard returns. When you use the **Insert** command in a component or selected text that you have deleted returns from, the software inserts returns corresponding to Interleaf software line breaks, not the line breaks that were in place before you deleted them. Figure 4-32a shows a paragraph from an external document. Figure 4-32b shows it after editing during which the returns have been removed, and Figure 4-32c shows it with returns put in again. Note that a return is not inserted at the end of the last line.

We at **Widget International** are often asked⊋
for some background on the origin of⊋
the word widget. Everyone here knows what⊋
we mean when we say widget, but anyone⊋
who has looked the word up in a dictionary knows⊋
that it doesn't appear in any of the⊋
standard dictionaries.⊋

(a) With software-generated line breaks

We at **Widget International** are often asked for
some background on the origin of the word widget.
Everyone here knows what we mean when we say
widget, but anyone who has looked the word up in a
dictionary knows that it doesn't appear in any of the
standard dictionaries.

(b) After deleting returns

We at **Widget International** are often asked for⊋
some background on the origin of the word widget.⊋
Everyone here knows what we mean when we say⊋
widget, but anyone who has looked the word up in a⊋
dictionary knows that it doesn't appear in any of the⊋
standard dictionaries.

(c) After inserting returns

Figure 4-32. Text with returns deleted and reinserted

Props

The **Props** command on the Text Selected menu is used primarily to gain access to the *Frames* property sheet (discussed in the chapter *Text-Anchored Frames*) or to token property sheets (discussed in the chapters *Auto Numbering and Referencing* and *Large Documents in Books*). If you choose **Props** when you have selected anything other than a single frame anchor, autonumber or autoreference token, or multiple index tokens, you get a message about the selected text (such as: *The current selection is a range of text*) in the status line.

This command is also useful if you want to know which of the five kinds of space available to you is between two words. If you select a space and execute the **Props** command, the message in the status line will tell you precisely which kind of space this space is. See *Spaces* in the chapter *Typography and Typefaces* for a description of the different kinds of spaces.

The **Props** command also identifies different types of tabs and returns. This is especially useful in documents in which you are not showing these characters.

Chapter 5

Spell Checking and Hyphenation

The Interleaf publishing software includes an 80,000-word on-line dictionary based on the *American Heritage Dictionary* (©Houghton-Mifflin). To supplement this *master* dictionary, you can create two additional dictionaries — a *desktop dictionary* and a *system dictionary*. The software then uses these combined dictionaries as a reference for two purposes:

- to check the spelling of words in a document

- to ensure proper hyphenation of most common words

Since your documents may contain words that do not appear in the master dictionary or in the additional dictionaries, an algorithm is provided to hyphenate these words in a reasonable way.

Hyphenation and spell checking are effective for all text in documents and microdocuments, but *not* for text strings in frames. The software excludes microdocuments in header and footer frames from document-wide spell checking operations. However, you can spell check such microdocuments individually.

To understand hyphenation and spell checking, you must learn some basic terminology and concepts.

- The term **dictionary**, unless otherwise specified, refers to the combination of dictionaries that you can currently access from your desktop.

- Stored with each word in a document are pieces of information invisible to the user that describe the spelling status and hyphenation points of the word.

- When the software *marks a word* as correct or incorrect, it sets the spelling status, or spelling **flag**.

- As you type words into a new document, the software inserts invisible **hyphenation points**. Hyphenation points are *not* hyphens; they merely tell the software where it is permissible to hyphenate words.

Spell Checking

The spelling checker allows you to locate words that are marked incorrect and do not appear in the dictionary. When you execute the **Spelling** command and the software encounters such a word, it highlights the word. In response you can:

- request a list of spelling alternatives and select one to replace the misspelled word
- skip to the next incorrect word
- add the highlighted word to your *desktop dictionary*
- edit the word yourself

When you add a word to the desktop dictionary, the software no longer highlights it as an error.

Spelling Conventions

Words that require all capital letters because they appear in the dictionary with all capitals are flagged as spelling errors when they appear in mixed or lower case letters.

The spelling checker does not flag symbols, mathematical characters, greek characters, or any single letters.

If one part of a compound word (whose parts are joined by a minus sign, en-dash, em-dash or slash) is misspelled, only the section containing the error is highlighted.

How to Search for Spelling Errors

You can access the spelling checker by using commands on two text menus, which are shown in Figure 5-1. For details on how to invoke the spelling checker using keyboard commands, see the chapter *The Keyboard*.


```
   Create  | Masters →          Cut
   Paste   | Tabs    →          Copy      →
   Fonts   | Show    →          Fonts       | Returns →
   Center  | Split              Props       | Capitals →
    Misc   | Find    →          Deselect    | Hyphen →
           | Replace →           Misc       | Find        Alternatives
           | Hyphen  →                      | Replace    □ Next Error
           | Spelling □                     | Spelling     Add to Dictionary
```

(a)	(b)
Text Location menu	Text Selected menu
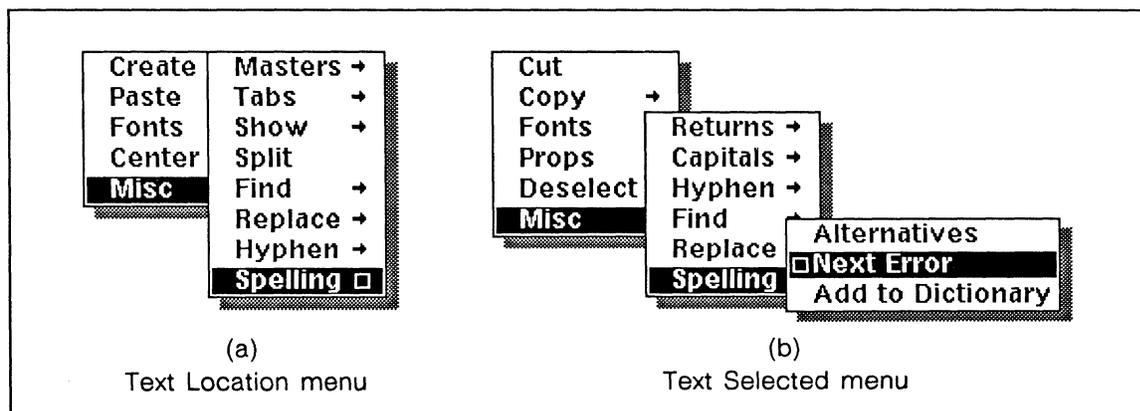
Figure 5-1. Spelling commands on the text menus

To check for spelling errors within a document:

✓ Position the text caret at the beginning of the document, or if you want to check only part of a document, at the point where you want to begin.

✓ On the Text Location menu, execute the ⌈Misc →⌉ ⌈Spelling⌉ command.

As it scans downward through the document, the spelling checker will highlight the first word it encounters that is marked incorrect and is not in the dictionary.

✓ When a word is highlighted, you have four choices.

1. You can correct the word using ⌈Misc →⌉ ⌈Spelling →⌉ ⌈Alternatives⌉. When you execute this command a Spelling Suggestion sheet may appear, providing alternative ways to spell the highlighted word. If the software is unable to provide an alternative, you will see a message to that effect.

2. You can execute the ⌈Misc →⌉ ⌈Spelling →⌉ ⌈Next Error⌉ command to move to the next word not found in the dictionary. **Next Error** is the default on the Spelling submenu. (The word will remain marked incorrect.)

3. You can execute the ⌈Misc →⌉ ⌈Spelling →⌉ ⌈Add to Dictionary⌉ command, which enters the word into a dictionary that resides on your desktop. Once this is done, the software will not highlight as incorrect this or any other instance of the word.

4. You can edit the word using the mouse and keyboard.

✓ If you select any option other than Next Error, execute the **Spelling** command (still the default) to resume checking.

✓ When the software passes the last incorrect word before the end of the document, a stickup menu gives you the choice to wrap to the beginning of the document and continue spell checking or to cancel the spelling operation.

You can also check to see whether one particular word is in the dictionary rather than search for the next error.

To see whether an individual word is in the dictionary:

✓ Position the text caret anywhere in the word you want to check (do not highlight the word).

🖤 On the Text Location menu, execute the [Misc →]
[Hyphen →] [Show] command.

You will see the hyphenation points (as determined by the dictionary or the algorithm) in the status line at the top of the document. If the word is not in the dictionary, you will see a message to that effect.

Spell Checking in Microdocuments

When you search a document for spelling errors, the software will highlight all incorrect words located in microdocuments. During the spell checking, the software opens only those microdocuments that contain an incorrect word.

To locate a spelling error in a microdocument, the software: *first* searches each frame until it locates a microdocument, and *second* searches each microdocument until it finds a spelling error. Note that the frame search path is determined by the order of the frame anchors, not by the order in which the frames appear.

If the software locates a frame that contains several microdocuments, the order in which it searches the microdocuments depends on their relative front-to-back positioning. The back-most microdocument is searched first and the front-most microdocument is searched last. By default, the first microdocument created is the back-most; the most recently created one is the front-most. You can use the diagramming **Front** and **Back** commands to control their front-to-back positioning.

Beginning a Check in a Microdocument

If you begin a spell check in a microdocument, a stickup will appear when the search reaches the end of the microdocument. In most cases, the stickup provides three options: you can wrap to the beginning of the microdocument, continue to the end of the main document, or cancel the search (Figure 5-2).

```
Do you wish to Wrap to the beginning of the microdocument,
         Continue to the end of the main document,
              or Cancel the search?

        [ Wrap ]  [ Continue ]  [ Cancel ]
                       ↑
```
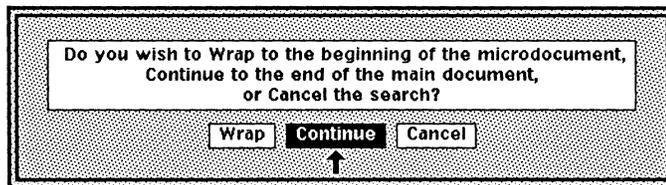
Figure 5-2. Microdocument Search Wrap stickup

If you choose to continue, this stickup will not appear when you spell check subsequent microdocuments *unless* you interrupt the spell checking procedure.

If the microdocument is in a header/footer frame, you can only wrap or cancel.

When you begin spell checking in a microdocument, the software will not search every microdocument within that frame unless you start from the back-most microdocument. However, it will search every microdocument in the frame if you wrap and spell check the document completely.

Spell Correction

When a word is highlighted and you execute the [Misc →] [Spelling →] [Alternatives] command, the software searches the dictionary for likely alternatives to the word marked incorrect. If it finds one or more possible replacements, a Spelling Suggestion sheet will appear displaying these alternatives. You can view additional suggestions by selecting the box labeled **More**.

Once you select a replacement word (displayed in reverse video), you can execute the **Replace** command on the Suggestion Sheet popup menu (Figure 5-3) to replace the incorrect word with the chosen word. Otherwise, execute the **Cancel** command to close the Spelling Suggestion sheet without replacing the incorrect word.

> □**Replace**
> **Cancel**

Figure 5-3. Suggestion Sheet Replace popup menu

To correct spelling errors:

🖝 Execute the [Misc →] [Spelling] command to check for spelling errors.

🖝 When a word is highlighted, execute the [Misc →] [Spelling →] [Alternatives] command. There are two possible results.

1. A Spelling Suggestion sheet will appear, providing alternative ways to spell the highlighted word.

2. The message: *There are no alternative spellings available* will appear in the status line.

🖝 If the Spelling Suggestion sheet appears, you will see the misspelled word and one or more alternatives.

For example, if the spell checker encountered the word definte, *the Spelling Suggestion sheet shown in Figure 5-4 would appear.*

Spelling Suggestions		
Suspect Word:	definte	
Suggestions:	**define**	
	MORE	
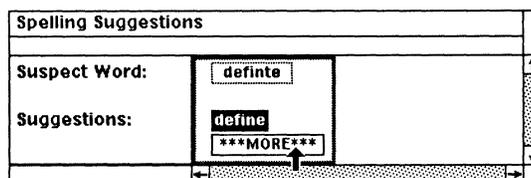
Figure 5-4. Example of a Spelling Suggestion sheet

🖝 To replace the highlighted word in text, *definte*, with the word displayed in reverse video on the Spelling Suggestion sheet, *define*, execute the **Replace** command.

or ✔ To see additional alternatives, point the mouse cursor at the box labeled **More** and click the **left** button.

As Figure 5-5 shows, the list of alternatives expands.

```
┌─────────────────────────────────────────────┬──┐
│ Spelling Suggestions                         │  │
├─────────────────────────────────────────────┼──┤
│                                              │ ↑│
│ Suspect Word:    ┌┄┄┄┄┄┄┄┄┄┐                 │▓▓│
│                  ┆ definte ┆                 │▓▓│
│ Suggestions:  ↑  └┄┄┄┄┄┄┄┄┄┘                 │▓▓│
│                  ┌─────┐                     │▓▓│
│                  │define│                    │▓▓│
│                  ├──────┤                    │▓▓│
│                  │definite│                  │▓▓│
│                  ├────────┤                  │▓▓│
│                  │***MORE***│                │ ↓│
├──────────────────┬────────────────────────┬─┼──┤
│                  │←│▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓│→│ │  │
└──────────────────┴────────────────────────┴─┴──┘
```
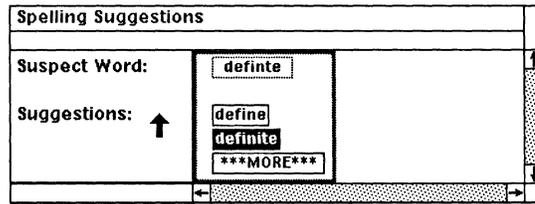
Figure 5-5. Expanded list of spelling suggestions

✔ You can repeat the previous step to request more alternatives until you exhaust the list of alternatives (Figure 5-6).

```
┌─────────────────────────────────────────────┬──┐
│ Spelling Suggestions                         │  │
├─────────────────────────────────────────────┼──┤
│ No more alternative spellings are available. │  │
├─────────────────────────────────────────────┼──┤
│ Suspect Word:    ┌┄┄┄┄┄┄┄┄┄┐                 │ ↑│
│                  ┆ definte ┆                 │▓▓│
│ Suggestions:  ↑  └┄┄┄┄┄┄┄┄┄┘                 │▓▓│
│                  ┌──────┐                    │▓▓│
│                  │define│                    │▓▓│
│                  ├──────┤                    │▓▓│
│                  │definite│                  │▓▓│
│                  ├────────┤                  │▓▓│
│                  │defiant│                   │ ↓│
├──────────────────┬────────────────────────┬─┼──┤
│                  │←│▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓│→│ │  │
└──────────────────┴────────────────────────┴─┴──┘
```
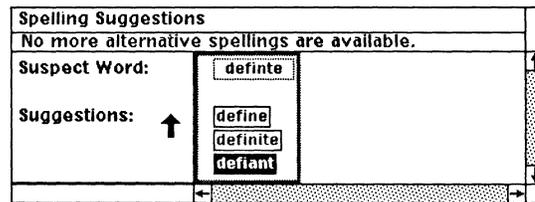
Figure 5-6. Complete list of spelling suggestions

✔ To choose the replacement word, point the mouse cursor at the word and click the **left** button. The word will display in reverse video.

✔ You can replace the suspect word with the word displayed in reverse video by executing the **Replace** command.

or ✔ If you do not find a suitable replacement on the Spelling Suggestion sheet, execute the **Cancel** command.

Types of Dictionaries

There are three types of dictionaries:

- the *master* dictionary, which is included with the software
- the *system* dictionary, which your system administrator must create
- the *desktop* dictionary, which is created automatically the first time you use the **Add to Dictionary** command

The **master dictionary** is an 80,000-word on-line dictionary based on the *American Heritage Dictionary*, published by Houghton-Mifflin. Although you cannot directly access the master dictionary, you can supplement it by creating other dictionaries.

Your system administrator can create a **system dictionary**. This dictionary is designed to store in one place the proper names and site-specific terminology used frequently by many individuals on a network. Individual users can create a link from their *system* cabinet to the system dictionary.

The **desktop dictionary** allows each user to store on his or her desktop a personal dictionary. To this dictionary you can add words not included in the other dictionaries. You can also use it to set the hyphenation points of any word.

We recommend that you add no more than a combined total of 15,000 words to the system and desktop dictionaries. Once you have filled the dictionary to capacity, you will see the following message when you use **Add to Dictionary**: *Desktop dictionary is full.*

A hierarchy exists among the three types of dictionaries. Any time that you perform an operation that requires looking up a word in the dictionary, the software looks first at the desktop dictionary, then at the system dictionary (if one exists), and finally at the master (Houghton-Mifflin) dictionary. Once it finds a spelling match, it stops looking for the word.

Although the search order is not significant during the spell checking, it enables you to set hyphenation points in the desktop dictionary that override the hyphenation defined in the other dictionaries.

For details on using the dictionary to set hyphenation points, see the section *Hyphenation Points in the Desktop Dictionary* later in this chapter.

Desktop Dictionary

The first time you execute the [Misc →] [Spelling →] [Add to Dictionary] command, the software begins compiling a **desktop dictionary**. Once you close a document in which you have executed this command, the dictionary icon (Figure 5-7) will appear on your desktop.
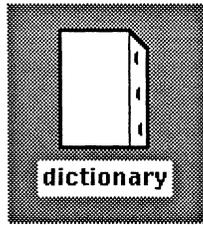
Figure 5-7. Dictionary icon

The **desktop dictionary** is a document-like file that stores words permanently. There are two ways to alter the contents of this dictionary:

- You can use the **Add to Dictionary** command to add words to it. Entries made using this method take effect immediately.

- You can open the dictionary file and type or **Cut, Copy,** and **Paste** words just as you would in a document. You must **Save** the dictionary to make these changes effective.

Each desktop can have only one active *desktop dictionary*. For the software to recognize and use a desktop dictionary, the following conditions must be met:

- the **icon type** and **icon name** must be *dictionary* (see Figure 5-8)

- the icon must remain on the desktop
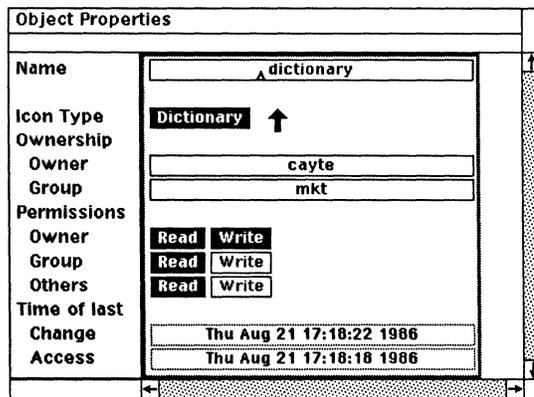
Figure 5-8. Dictionary Object property sheet

If you fail to observe these conditions by:

- placing more than one dictionary icon named *dictionary* on your desktop, the software considers one to be your active dictionary and does not recognize the contents of the others.

- renaming your desktop dictionary to something other than *dictionary*, the software will begin compiling a new dictionary. As you type new instances of words stored in the original dictionary, they will be marked as incorrect.

- moving your dictionary off the desktop and into a subdirectory, the software will not recognize it and will begin compiling a new one.

Using Add to Dictionary to Update the Dictionary

The simplest method for updating the dictionary is by using the **Add to Dictionary** command. If your desktop dictionary is open, we recommend that you close it before using this command.

To add a word to the desktop dictionary:

➤ If the word was not already highlighted by the spelling checker, select it.

Note that you can add several words to the dictionary at once by selecting a group of words.

➤ Execute the [Misc →] [Spelling →] [Add to Dictionary] command.

If the word is already in the dictionary, you will see a message to that effect.

The **Add to Dictionary** command has the following effects:

- It adds the selected word to the desktop dictionary, although the word does not appear in the dictionary until you close the document containing the word.
- It marks each word added as correct.
- As you type new instances of words you have added, they are marked as correct.
- During subsequent spell checking, as the software passes instances of words you have added it sets their flags to correct.
- When you save a document, the software saves all changes to the spelling status of words in the document.

If the dictionary is open on your desktop when you use **Add to Dictionary**, you must close it *without saving* and then reopen it to see your additions. If you save your changes, you will lose all the **Add to Dictionary** entries made since you opened the dictionary.

How to Edit the Dictionary File

You can open the dictionary icon and edit the contents just as you would a document. However, you must follow certain guidelines for the dictionary to operate correctly.

- Place dictionary entries in **components** named *"words"*.
- Follow each dictionary **entry** by a *hard return*; words not separated by a hard return are not recognized.
- Indicate **hyphenation points** with *asterisks*.

If you open your desktop dictionary, you will notice that the software uses these guidelines. As the sample dictionary in Figure 5-9 shows, the software sorts the contents of the dictionary alphabetically. A new component exists for each letter of the alphabet for which there is a dictionary entry. Also, hyphenation points have already been set on certain words. The algorithm attempts to set hyphenation points for each word that you add to the dictionary.
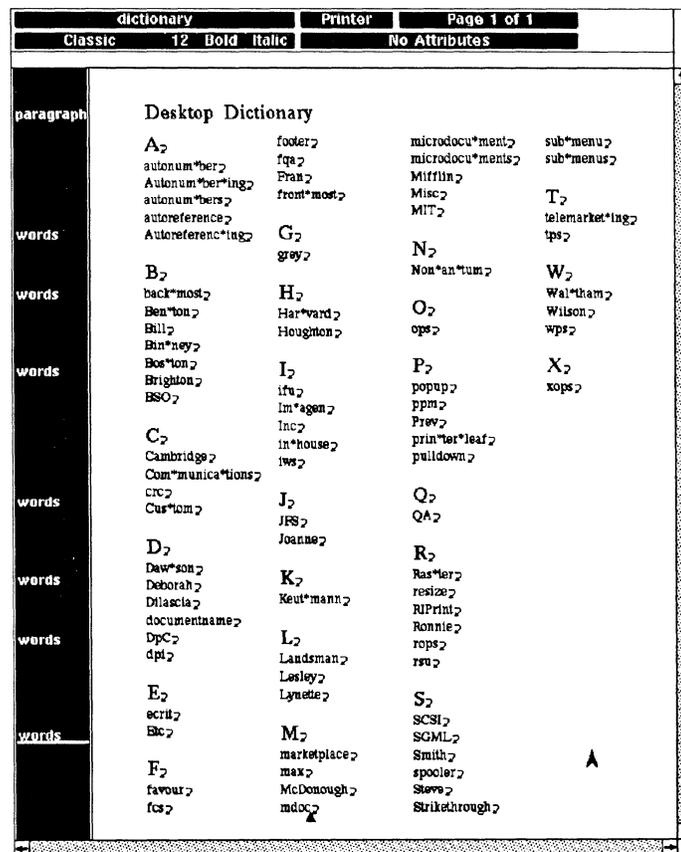
Figure 5-9. Sample open desktop dictionary

The desktop dictionary has certain default page, component, and text properties. For example, the page is divided into four columns, the components are named

words, and the text appears in Classic 12-pt. roman type. If you modify these properties, your changes will be lost the next time the dictionary is sorted.

To edit the desktop dictionary:

✔ Select and **Open** the dictionary icon.

✔ Use the keyboard and the **Cut, Copy,** and **Paste** commands to add or delete dictionary entries as you like. Be sure to follow the formatting guidelines.

> *You need not place entries in alphabetical order. The software re-sorts the entries the next time you execute* **Add to Dictionary.**

✔ **Save** your changes.

> *Changes to the dictionary file do not take effect until you save them.*

When you **add** a word to the dictionary and then

- type new instances of the word, they are marked as correct.
- continue spell checking, existing instances of the word are marked as correct when encountered by the spell checker.

When you **delete** a word from the dictionary and then

- type new instances of the word, they are marked as incorrect (unless the word exists in the system or master dictionary).
- continue spell checking, some instances of the word may be highlighted as incorrect while others may not. If an instance was typed or spell checked while the word was in the dictionary, then it will not be highlighted. Otherwise, the instance will be highlighted.

If you plan to edit the contents of the dictionary file, you should understand how the dictionary works and the implications of your actions. For more information, read the section *When the Spelling Flag and Hyphenation Points are Set.*

System Dictionary

In addition to the desktop dictionary, users can create a link in their *system* cabinet to a **system dictionary.** By placing the names and terminology used by many individuals on the network in the system dictionary, you can save space. First the system administrator must create the system dictionary and store it in the *system* cabinet on his/her desktop. Then other users wanting to access the system dictionary must create a link to this system dictionary. To be effective, the link must

exist in the user's *system* cabinet. A link created in any directory other than the *system* cabinet will not be recognized or used by the software.

Only the owner of the system dictionary, the system administrator, can edit the contents of the system dictionary. To do this, he or she must open the system dictionary file and use the keyboard and the **Cut, Copy,** and **Paste** commands to add or delete dictionary entries. The format of the system dictionary is identical to that of the desktop dictionary.

For information on how to create and edit the system dictionary and how to create a link to the system dictionary, see the *Administration* manual.

Hyphenation

In this section, we discuss the following topics:

- how the software sets hyphenation points
- how to see where the hyphenation points are
- how to set and remove hyphenation points on a single instance of a word and on all instances of a word in a document
- how to use the dictionary to set hyphenation points
- how the software hyphenates
- how to control hyphenation

In this chapter, when we refer to hyphens, we mean the character inserted *by the software* at the end of a line, not the hyphen character that the user can type from the keyboard. The keyboard hyphen is referred to as a *hard hyphen*.

The purpose of hyphenation is to make blocks of text as aesthetically pleasing as possible. In ragged right text, hyphenation helps keep lines of text in similar components relatively uniform in length. In justified text, it helps keep the spaces between words relatively uniform.

When a component has alignment that is **Flush Left** and **Flush Right**, both margins of the component are **justified**; i.e., every line of the component except its last is exactly the same length and stretches from the left-hand margin to the right-hand margin. If words in a justified component cannot be hyphenated, the spaces between words sometimes have to be stretched out to unattractive lengths so that the text will extend from the left margin to the right margin.

When a component has an alignment of **Flush Left** only *or* **Flush Right** only, one of the margins of the component is **ragged**. When words in a ragged component cannot be hyphenated, the lengths of lines can vary widely, particularly in narrow columns or when long words occur towards the end of lines.

When a component has an alignment of **Centered**, both margins of the component are ragged. It is rare to use hyphenation in a centered component; however, hyphenation for centered components is handled the same way that it is for other components with ragged margins.

Proper use of hyphenation is especially important in multicolumn pagination, where narrow columns can cause unhyphenated text to take on unacceptable word spacing or line lengths. The *Page Design* chapter includes more information on hyphenation in multicolumn documents.

Hyphenation Points vs. Hyphens

As you type words into a new document, the software inserts invisible **hyphenation points** in those words. When you open documents created with version 2.0 or earlier of the Interleaf publishing software and external documents that are copied to your desktop, the software also inserts hyphenation points in the text of these documents.

Hyphenation points are not hyphens; they merely tell the software where it is permissible to hyphenate words if two conditions are met. The first condition: *Hyphenation* must be turned **On** for the whole document on the Custom sheet of the Document Page property sheet (Figure 5-10). The default setting is **On**.

```
Page Properties  [ Normal ]  Custom

Hyphenation          ↑        On  Off
  # Consecutive               Any 1 2 3 4

Revision Bar Placement        Left  Automatic

Gutter Width                  Default  2   picas

Balance Columns               On  Off

Vertical Justification        On  Off
  Margin Stretch                200  %
  Margin Shrink                 50   %
  Frame Margin Stretch          10   %
  Feathering, Maximum         On  Off   8   %
  Vert. Just. Pages           On  Off
  Depth, At Page Break          95   %
  Depth, No Page Break          90   %
```
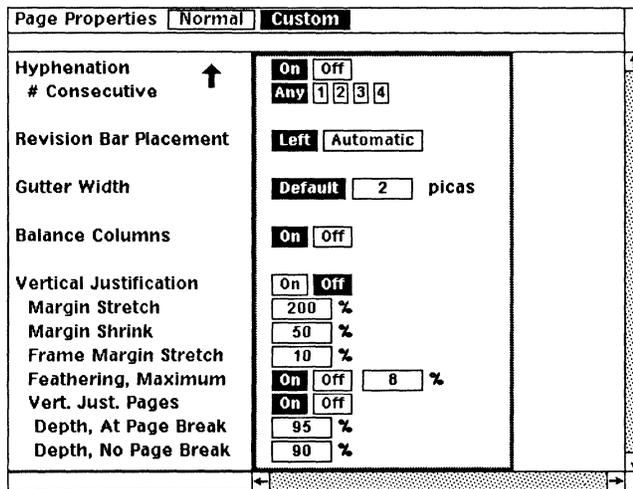
Figure 5-10. Page Custom property sheet with hyphenation setting for the entire document

The second condition: the *Hyphenation* setting on the property sheet for the particular component must be something other than *Off*. Figure 5-11 shows a Custom sheet for the *Component* property sheet with the default setting of **Normal**.
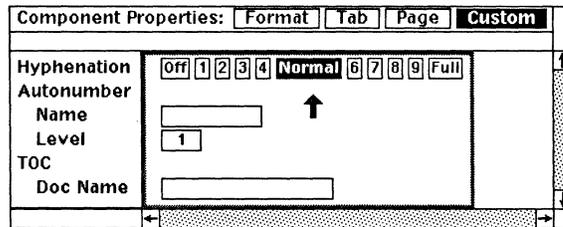
Figure 5-11. Component Custom property sheet with hyphenation settings

How Hyphenation Points are Set

The first source of software-generated hyphenation is the dictionary. If a word is not in the dictionary, the software sets that word's hyphenation points based on an algorithm of Donald E. Knuth.* The rules used in this algorithm are conservative so that the system will almost never hyphenate incorrectly, although it may not hyphenate as frequently as you would like.

If you do not like the way the algorithm, or even the dictionary, sets hyphenation points in a word, you can change them by:

- using the **Misc Hyphen** commands on the Text Location menu
- setting the hyphenation points in the desktop dictionary

With the **Misc Hyphen** commands you can set the hyphenation points for single instances of words. We refer to these as *manually-set* hyphenation points. All other hyphenation points, including those you define in the desktop dictionary, are referred to as *software-generated* hyphenation points.

Hyphenation Conventions

There are several guidelines for setting and clearing hyphenation points:

- you cannot set a hyphenation point immediately after the first letter in a word
- you cannot set a hyphenation point immediately before the last letter in a word
- you cannot set a hyphenation point in a word that is fewer than five characters long
- you cannot set a hyphenation point before a *hyphen character* (-); an *apostrophe* ('); a *slash* (/); an *em dash* (—); or an *en dash* which is the same character on the keyboard as the *minus sign* (-)

---

* *T$_E$X and Metafont: New Directions in Typesetting*, Digital Press, American Mathematical Society, 1979, pp. 180-186.

- the software sets a hyphenation point after a *hyphen character* (-), after a *slash* (/), after an *en dash* (–), and after an *em dash* (—); as with other hyphenation points, you can set or clear these points

- compound words found in the dictionary (e.g. great-grandmother) have hyphenation points at the hard hyphen only

- if you edit a word containing a manually-set hyphenation point, the hyphenation point will be cleared

- if you cut or copy a word containing a manually-set hyphenation point, the point will be preserved when you paste the word as long as the word is not changed

The Hyphenation Commands on the Text Popups

You can remove hyphenation points from a single instance of a word and set other ones if you want.

There are hyphenation commands on both the Text Location popup menu (Figure 5-12a) and the Text Selected popup menu (Figure 5-12b). These make it possible for you to manipulate hyphenation in text in several ways. **Set** and **Clear** appear on both submenus, but they have different meanings on the two popups.
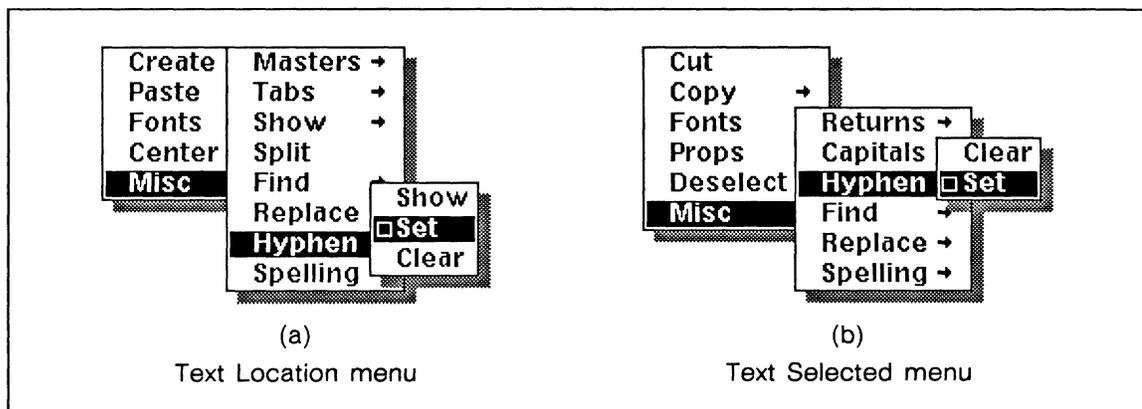
Figure 5-12. Hyphenation submenus on the text popup menus

The Hyphenation Commands on the Location Menu

There are three hyphenation commands on the Text Location menu.

With the **Show** command, you can find out where there are hyphenation points in a particular word.

The **Set** command is the default and makes it possible for you to put a hyphenation point at a place in a word where the system does not normally hyphenate.

The **Clear** command on this menu takes hyphenation points out. With the **Clear** command, you can remove both software-generated and manually-set hyphenation points one at a time.

To see the hyphenation points in a word:

↙ Position the text caret anywhere in the word.

↙ On the Text Location menu, execute [Misc →] [Hyphen →] [Show].

The word appears in the status line. If hyphenation points have been set in it, by either the software or the user, these points are marked.

To set a hyphenation point:

↙ Position the text caret where you want the hyphenation point.

↙ On the Text Location menu, execute [Misc →] [Hyphen →] [Set].

If there is already a hyphenation point at this location, a message in the status line informs you that there has been no change. It also shows you all the hyphenation points in the word.

To clear a hyphenation point:

↙ Position the text caret at the hyphenation point you want to remove.

↙ On the Text Location menu, execute [Misc →] [Hyphen →] [Clear].

If there is no hyphenation point at this location, a message in the status line informs you that there has been no change. It also shows you all the hyphenation points in the word.

If there is a hyphen at the end of a line, you cannot position the text caret after it on the same line. To clear the hyphenation point, which in this case removes an actual hyphen, position the text caret at the *beginning of the next line* before executing [Misc →] [Hyphen →] [Clear].

Preserving Manually-Set Hyphenation Points

Discretionary (manually-set) hyphens are removed from a word if the word is edited. For example, the word *rehydrated* is not hyphenated by the software, but you may want to set a hyphenation between *re* and *hydrated*. If you later delete the *d* from the end of the word, the hyphenation point will be removed.

You can preserve hyphenation points by avoiding unintentional editing. For example, if you want to insert *was* before *rehydrated*, make sure there is a space between the text caret and *rehydrated* before you type *was*.

A similar strategy applies when you have cut or copied a word with a discretionary hyphenation point in it and are pasting it somewhere. If you make sure there is a space on both sides of the text caret when you execute the **Paste** command the hyphenation point will be preserved. Otherwise, the manually-set hyphenation points in the word will be removed and the software-generated points reset according to the dictionary or algorithm.

The Hyphenation Commands on the Text Selected Menu

There are two hyphenation commands on the Hyphenation submenu of the Text Selected menu.

When you execute a **Clear** command on selected text, the system removes both software-generated and manually-set hyphenation points from the selected text. This is useful when you want hyphenation in a document and in a component but you do not like the way a portion of the component looks when it is hyphenated.

When you execute a **Set** command on selected text, the system puts back *only* the software-generated hyphenation points. You would need to use this command:

- if you had used one of the **Clear** commands and wanted to reset the software-generated hyphenation points
- if you wanted to get rid of all manually-set hyphenation points in a section
- if you wanted to update the hyphenation points and spelling status of the text according to the latest information in the desktop dictionary

Hyphenation Points in the Desktop Dictionary

In some cases you may want to change the hyphenation points of all future instances of a word, or of existing instances within a particular document as well. There are two methods available to achieve these results, but one of the methods requires that the word *not* exist in the dictionary yet.

If the word is not in the dictionary, you can first set the hyphenation points of an instance and then enter that word in the desktop dictionary. The software will use the new hyphenation for all subsequent entries of the word. In addition, a stickup will appear asking if you want to rehyphenate all instances of the word in your open document.

To set the hyphenation points of a word not yet in the dictionary:

✔ If the word was not already highlighted by the spelling checker, select it.

✔ On the Text Selected menu, execute the ⌐Misc →⌐ ⌐Hyphen →⌐ ⌐Clear⌐ command to clear the existing hyphenation points.

✔ Position the text caret where you want a hyphenation point.

✔ On the Text Location menu, execute [Misc →] [Hyphen →] [Set].

Perform this and the previous step to create each hyphenation point.

✔ Select the word.

✔ Execute the [Misc →] [Spelling →] [Add to Dictionary] command.

If the word is already in the dictionary, you will see a message to that effect. Otherwise, the stickup shown in Figure 5-13 will appear, displaying both the original and new hyphenation points.

```
┌─────────────────────────────────────────────────────────┐
│  ┌───────────────────────────────────────────────────┐  │
│  │        Do you wish to change the hyphenation of all │  │
│  │ occurences of 'wordtochange' in this document to be 'word*to*change' ? │  │
│  └───────────────────────────────────────────────────┘  │
│                         [No] ▐Yes▌                       │
│                              ↑                           │
└─────────────────────────────────────────────────────────┘
```
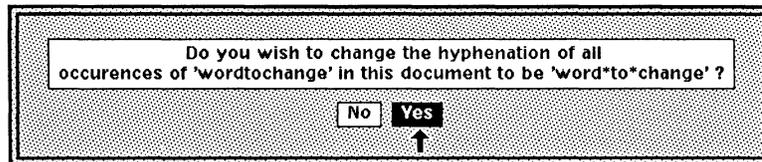
Figure 5-13. Hyphenation Points stickup menu

✔ If you want to change the hyphenation points of all existing instances of the word in your document, select **Yes**. If you select **No**, existing instances will not be affected.

Whether you select **Yes** *or* **No**, *the word will be added to the dictionary with the new hyphenation points.*

If the word is already in the dictionary, you can override its software-defined hyphenation points by editing the desktop dictionary and redefining the hyphenation points there. Note that you can also edit the dictionary to add a new word and define its hyphenation points.

Within the dictionary file each hyphenation point is indicated by an asterisk; a word without asterisks has no hyphenation points. To redefine the hyphenation points of a word, add or remove asterisks as you want. Each instance of the word you type thereafter assumes the new hyphenation points.

To set the hyphenation points of a word by editing the desktop dictionary:

✔ Select and **Open** the dictionary icon.

✔ If the word already exists in the desktop dictionary, find the word and remove all existing hyphenation points (asterisks) that you do not want.

or ✔ If the word does not exist in the desktop dictionary, create a new *words* component and type the word. Press RETURN.

✔ Position the text cursor where you want a hyphenation point and type an asterisk.

✔ **Save** the dictionary file.

Once you have set the hyphenation points of one or more words in the dictionary, you may want to reset all instances of those words in your document.

To change the hyphenation points of all instances of a word in a document:

↳ Set the hyphenation points of the word in the desktop dictionary.

↳ **Close** and **Save** the desktop dictionary.

↳ **Open** the document containing the words whose hyphenation points you want to change.

↳ Select all of the *text* in the document.

↳ Execute the ⌈Misc →⌉ ⌈Hyphen →⌉ ⌈Set⌉ command.

The software updates both the spelling status and hyphenation points of all words in the file according to the latest dictionary information.

↳ **Save** the document.

How Hyphenation Works

Hyphenation works in slightly different ways in ragged and in justified text. However, you can switch from ragged to justified or justified to ragged text, and the software will make the necessary adjustments for you.

Hyphenation settings that appear on the Custom sheets of the Component property sheet and Document Page property sheet affect hyphenation frequency. The default setting on the Component property sheet, *Normal,* is designed to give a reasonable hyphenation frequency in both ragged and justified components. As the settings move towards *off,* the frequency of hyphenation in the component decreases until none occurs when *off* is the setting. In the other direction, as the settings move towards *full,* the frequency of hyphenation in the component increases.

All of the initial explanations here assume that the *# Consecutive* setting on the Page property sheet is **Any**. The section *Limiting Consecutive Hyphenation* explains how other settings alter the way the software hyphenates.

Hyphenation in Ragged Text

In this section, we use the hyphenation of flush left / ragged right text as the example, but hyphenation in ragged left and centered text behaves in the same way.

The hyphenation setting determines how much hyphenation can take place when ragged text is being hyphenated. The setting establishes a zone at the end of each line of text. If a *spaceband* (the space produced when you press the space bar on your keyboard) occurs within this zone, the line breaks at the spaceband and

hyphenation does not take place. If no spaceband occurs before the maximum width of the component is reached, the software will attempt to hyphenate.

The lower the number selected on the property sheet, the wider the hyphenation zone, and therefore the less hyphenation will occur. The hyphenation zones vary from 0 to 40 percent of either 3 inches or column width (whichever is greater).

Spaces between words are not stretched to fill the line in ragged text.

Hyphenation in Justified Text

When text is justified, the spacebands often have to be made bigger in order to stretch the line out from the left to the right margin. If the software used a fixed hyphenation zone to determine when to hyphenate justified text, a line with only two or three words would have very wide spaces between those words and would look ugly (narrow, justified columns in newspapers often have this problem).

Instead of using a fixed zone approach to hyphenating justified text, the Interleaf publishing software uses the concept of *spaceband stretch*. The same settings that determine the width of the hyphenation zone for ragged text determine the amount that each spaceband on a line can be stretched beyond its natural width before the software looks for a word to hyphenate. In effect this means that the amount of space added to a justified line before hyphenation takes place varies depending upon the number of spacebands on that line. (The *natural width* of a spaceband is the width between words in ragged text that has been determined to give the most aesthetic and readable effect.)

For example, when the hyphenation setting for a component is *Normal,* every spaceband on a line can be up to *1.5* times its natural width before the system looks for a word to hyphenate. When the setting is *off,* there will be no hyphenation in the component, and the spacebands on each line will be stretched as far as necessary to justify the text. At the other end of the spectrum, when the setting is *Full,* the spacebands are not supposed to stretch beyond their natural width.

Since it is not always possible to find a legal hyphenation point where it is needed, spacebands in justified components are sometimes stretched more than their hyphenation setting suggests. Spacebands are also stretched beyond the hyphenation setting when the number of consecutive hyphens allowed is fewer than the number that would otherwise occur.

Limiting Consecutive Hyphenation

You may further control the hyphenation of your documents by limiting the maximum number of consecutive hyphenated line endings. This property is set on the Page property sheet. You can select from 1 to 4 lines, or instruct a document to accept any number of consecutive hyphenated line endings.

**To set the maximum number of consecutive
hyphenated line endings in a document:**

    ✔ Open the Page property sheet.

    ✔ Select the Custom sheet.

    ✔ At the *# Consecutive* line, select the choice you want to acti-
vate and then **Apply** the change and **Close** the property
sheet.

In many components, limiting the number of consecutive hyphenated lines will
not have any effect on the component. If the component is set for normal to full
hyphenation, however, limiting the number of consecutive hyphenated lines to
one, for instance, is likely to change the appearance of that component.

◈ *The Algorithms that Determine the Frequency of Hyphenation*

The following information is for users who want to know more precisely what val-
ues the *Hyphenation* settings on the Component property sheet represent.

For components with ragged text, the settings correspond to a percentage of the
column width if the component is *3* inches wide or wider. If the component is
narrower than 3 inches, the hyphenation zone is a percentage of *3* inches.
Figure 5-14 shows how these settings work.

Ragged Text	
Setting	Percentage
Off	——
1	40.0%
2	20.0%
3	13.3%
4	10.0%
Normal	8.0%
6	6.7%
7	5.7%
8	5.0%
9	4.4%
Full	0.0%

*Figure 5-14. Translating hyphenation settings to percentages of
column width*

For justified components, the settings correspond to the amount the spacebands
can be stretched before the system attempts to hyphenate. For example, if the

setting is *Normal*, each spaceband on a line can be increased by one-half of the natural width. Figure 5-15 shows how these settings work.

Justified Text	
Setting	Maximum stretch
Off	——
1	4
2	3
3	2
4	1.75
Normal	1.50
6	1.38
7	1.28
8	1.18
9	1.09
Full	1.00

Figure 5-15. Translating hyphenation settings for justified components into the amount each spaceband on a line will stretch

When the Spelling Flag and Hyphenation Points are Set

Each word has an associated *spelling flag* and *hyphenation points* that carry information about the word. The spelling flag identifies whether the word is recognized by the dictionary; the hyphenation points identify where it is permissible to hyphenate the word.

Spelling flags can be set:

- when you type or edit a word
- when you check the spelling of a word
- when you execute the **Add to Dictionary** command
- when you select a word and reset its hyphenation points by executing the [Misc →] [Hyphen →] [Set] command
- when you first open in software version 3.0 any ASCII document or any document created with version 2.0 or earlier

Hyphenation points can be set:

- when you type or edit a word
- when you execute the **Add to Dictionary** command
- when you execute the [Misc →] [Hyphen →] [Set] or [Misc →] [Hyphen →] [Clear] commands on either text menu
- when you first open in software version 3.0 any ASCII document or any document created with version 2.0 or earlier

Since the software sets the spelling flags and hyphenation points on specific occasions and does not immediately update existing instances every time you make a change, a single document can contain information for a single word that varies from one instance of that word to the next.

For example, suppose you add a word to the dictionary, spell check a document containing the word, remove the word from the dictionary, and then type new instances of the word in the same document. When you spell check again only those instances added *after* you removed the word from the dictionary will be highlighted as incorrect.

Similarly, if you change the hyphenation points of a word by editing the word in the dictionary, existing instances of the word will continue to have the original hyphenation points while the new instances will carry the new hyphenation points.

There are several ways to ensure that your documents contain consistent spelling and hyphenation information. The simplest method is to select all of the text in your document and execute the [Misc →] [Hyphen →] [Set] command.

Chapter 6

Search and Replace

Using the **Find** command on the Misc submenu of the Text Location menu, you can search for:

- strings of text
- frame anchors
- index tokens
- autonumber tokens
- autoreference tokens

You can start a search from anywhere in a document or microdocument, search forward or backward, and switch the direction of the search whenever you like.

Figure 6-1 shows the **Find** command and its submenu. Every time you execute **Find** on the Text Location menu, you establish the search object and direction.
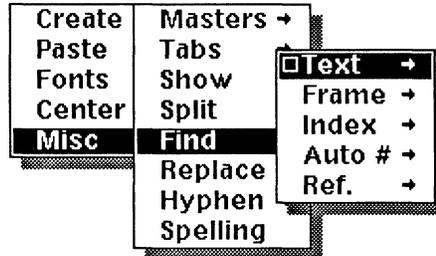
Figure 6-1. Find submenu on Text Location menu

Once the software finds an instance of the object being searched for, it selects and highlights that instance. You can use the **Find** command on the Text Selected Misc submenu to find the **Next** or **Previous** instance of that same object and to change the direction of the search (Figure 6-2). By default, **Find** searches for the same object in the same direction.

To resume a search that has been interrupted, select any text and execute **Find**. The software remembers the object and direction of your last search.
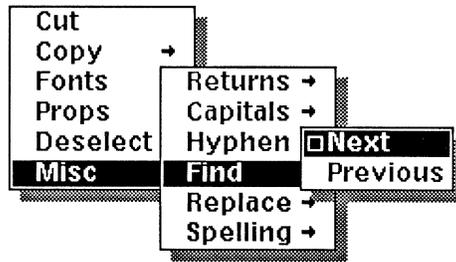
Figure 6-2. Find submenu on Text Selected menu

You can use the **Replace** command to replace one string of characters with another (for example, to replace *elephant* with *zebra* or *American businesses* with *United States corporations*). You can replace the search string one instance at a time, globally, or from the current location to the end of the document.

Before replacing a single instance, you must use the **Find** command. When replacing globally or to the end of the document, you can execute **Replace** without using **Find** first.

Note that you cannot use **Replace** to replace frames, index tokens, autonumber tokens, or reference tokens.

Choosing Search Objects

The Find submenu on the Text Location menu has five options, each with a submenu. Each submenu enables you to choose the direction of the search using either **Forward** and **Reverse** or **Next** and **Previous**. In addition:

- the *Index* and *Ref.* submenus enable you to search for the **First** or **Last** index token and autoreference token in the document (Figure 6-3b)

- the *Frame* and *Auto#* submenus enable you to search for the **First** or **Last** frame and autonumber token in the document, or search for the **Next of** or **Prev of** a specific instance, such as a particular frame master or a token from a specific autonumber stream (Figure 6-3c)

(a) Text submenu   (b) Index and Ref. submenu   (c) Auto# submenu
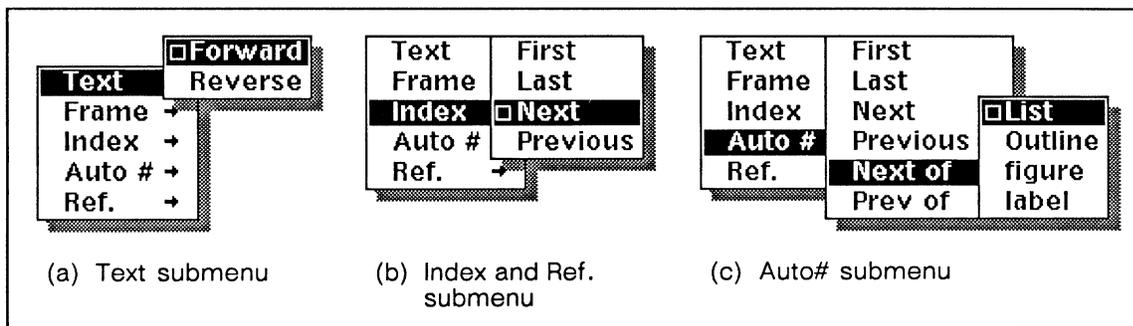
Figure 6-3. Submenus of Text Location Find submenu

Finding Invisible Anchors and Tokens

Using the [Misc →] [Show →] command on the Text Location menu, you can turn the display of frame anchors and index tokens on or off. In addition, you can use the *Show* line on the Autonumber Stream property sheet to turn the display of each autonumber level on or off. By default, the display of frame anchors, index tokens and autonumber tokens is *on*.

When you search for a particular object, the software will locate and select the next instance of that object regardless of whether the display is on or off. However, if the display is off, the selected object will not be visible.

Search Paths

All search paths begin at the text caret and continue to end of the document during a forward search, or to the beginning of the document during a backward search. When the search reaches the beginning or end, a stickup will appear presenting two options. If you were searching forward, you can cancel the search or wrap to the beginning of the document (Figure 6-4a). If you were searching backward, you can cancel the search or wrap to the end of the document (Figure 6-4b).

> Do you wish to continue by wrapping to the beginning of the document?
>
> [Wrap] [Cancel]

a. Forward search

> Do you wish to continue by wrapping to the end of the document?
>
> [Wrap] [Cancel]
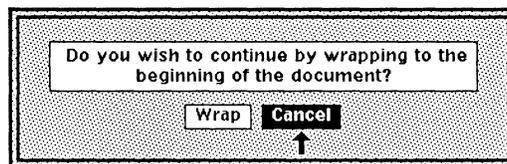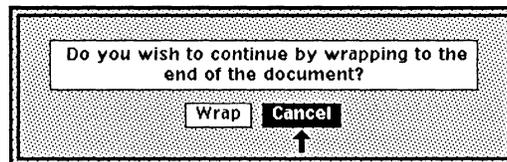
b. Backward search

Figure 6-4. Document Search Wrap stickups

The software excludes header and footer frames and the objects they contain from all document-wide search and replace operations. If you begin a search or replace in a header/footer frame, you can only wrap or cancel.

Frame Search Path

The *frame* search path is determined by the order of the frame anchors, not by the order in which the frames appear. When the anchor is found, both the frame and anchor are highlighted (Figure 6-5).
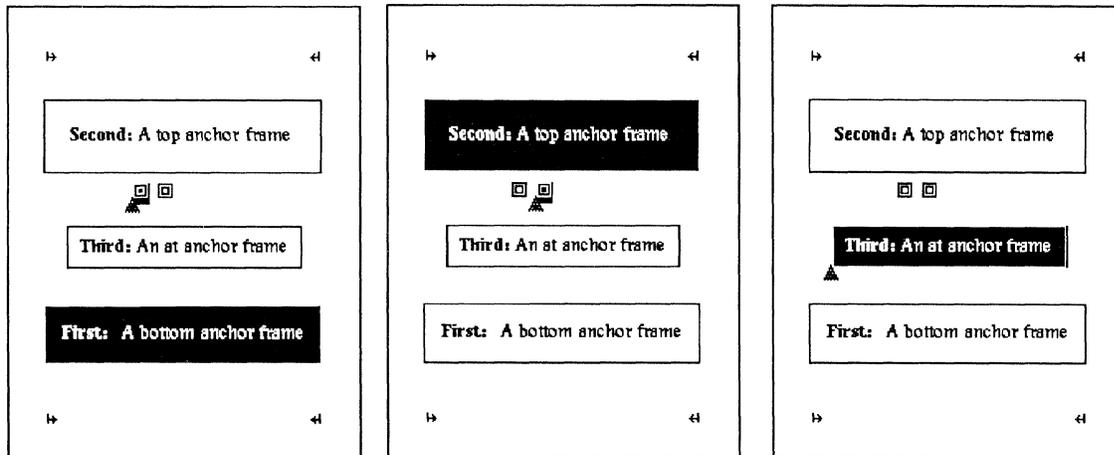
Figure 6-5. Frame search path determined by frame anchors

Microdocument Search Path

When you search for any object *except* a frame, the software will find all instances of the object both in the main document and in microdocuments. During the search, the software opens only those microdocuments that contain an instance of the object.

When you begin a search in the main document, the software performs two levels of searches to locate an instance in a microdocument: *first* it searches each frame until it locates a microdocument, *second* it searches each microdocument until it finds an instance of the object you are searching for.

If the software locates a frame that contains several microdocuments, the order in which it searches the microdocuments depends on their relative front-to-back positioning. By default, the first microdocument created in a frame is the back-most; the most recently created one is the front-most. In a forward search, the back-most microdocument is searched first and the front-most microdocument is searched last. Conversely, in a backward search the front-most microdocument is searched first and the back-most microdocument is searched last.

You can use the diagramming **Front** and **Back** commands to control the front-to-back positioning of microdocuments.

Beginning a Search in a Microdocument

The Forward Search

If you begin a forward search in a microdocument, a stickup will appear when the search reaches the end of the microdocument. The stickup provides three options: you can wrap to the beginning of the microdocument, continue to the end of the main document, or cancel the search (Figure 6-6).
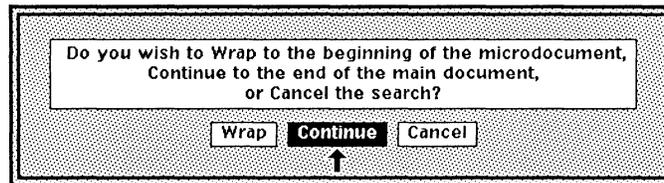
Figure 6-6. Microdocument Forward Search Wrap stickup

The Backward Search

If you begin a backward search in a microdocument, a stickup will appear when the search reaches the beginning of the microdocument. The stickup provides three options: you can wrap to the end of the microdocument, continue to the beginning of the main document, or cancel the search (Figure 6-7).
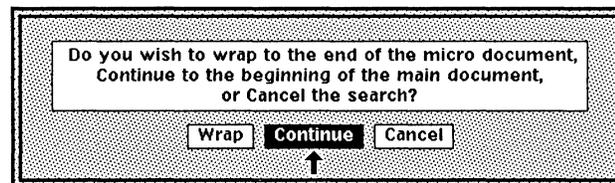
Figure 6-7. Microdocument Reverse Search Wrap stickup

Search Options in a Microdocument

If you choose to **Wrap**, the search will be confined to the open microdocument. If you then search the microdocument completely from beginning to end, or from the end to the beginning, the software will provide another chance to wrap, continue, or cancel.

If you choose to **Continue**, the software will continue searching forward or backward through the main document. As the software encounters subsequent microdocuments, it will search them from beginning to end. The software will not present the Microdocument Wrap stickup again *until* you interrupt the search procedure and then continue the search in a microdocument, or you cancel the procedure and initiate another search in a microdocument.

When you begin a search or replace operation from a microdocument, the software may not search every microdocument within the same frame, since the microdocument search path depends on the relative front-to-back positioning of the microdocuments. To ensure that all microdocuments within a frame are searched, either position the text caret before the frame anchor and begin a forward search, or continue to search the main document and then wrap to search the document completely.

Searching for Text

When you initiate a text search, the software asks you what you want to search for; that is, it asks you for the search string.

A **search string** is a group of characters. It can include spaces, tabs, and hard returns. A good search string is one that is inclusive enough to find what you are looking for and exclusive enough to ignore a search string that is embedded in other strings; for example, one that will find *is*, but will not include *this* or *history*.

You can search for almost anything you can type on your keyboard. For example, you can search for minus signs (–) or you can search for hyphens (-). You insert hyphens into a search string the same way you insert them in text: by pressing the ESC key and then typing a minus sign. (See the chapter *The Keyboard* for other sequences that use the ESC key.)

With the Interleaf publishing software, a search string can be 63 characters long, but users seldom approach this limit.

If you keep several considerations in mind, you will be able to use the search operation efficiently.

- If, in response to the question on the stickup, you type a search string with only lower case letters in it, the search routine will find any example of the string. Thus, you can type *elephant* if you want to find *elephant, Elephant,* and *ELEPHANT.*

- The search operation is sensitive to uppercase letters. If you use uppercase letters in a search string, the same string will be found in text only if *at least* those same letters are uppercase. The word also will be found if additional letters are uppercase.

For example, if you type *Elephant,* you will find *Elephant, ELEPHANT,* and any other version of the word that has the initial *E* capitalized, but it will not find *elephant.* If you type *ELEPHANT,* the operation will find only *ELEPHANT.* If you type *ELephant* or *elepHaNt* or any other version of the word that has upper case letters in any but the first position, it will find only *ELEPHANT* unless it finds an exact match in text.

- When your search string includes characters produced by ESC and another key, the escape sequence is usually represented on the Find stickup menu as ♠ *character* or as an *<abbreviation>.* For example, a hyphen appears as ♠–, and a hard space appears as *<hsp>.*

- Because the RETURN key can be used to confirm a search string, you must use a special key sequence to indicate that you are searching for a hard return: BACKSPACE ESC RETURN. On the Find stickup, the hard return appears as *<ret>.*

- When you want to search for a word like *is* but not for all the words that have *is* in them, such as *this* and *history,* type a space before *is.* (Do not type a space after the word or you will miss any instances of the word followed by a punctuation mark.)

The software will not find or replace any instance of the search string that is split by an at anchor frame, a frame anchor, or an index token.

Forward and Backward Searches

There are several ways to start a forward or reverse search and to confirm that the search string you have typed is the correct one. We present one method for the forward search and one for the backward search. For information on the other methods, see the section *Keyboard Alternatives.*

The Forward Search

A forward search begins at the position of the text caret and moves towards the end of the document.

To begin a forward search:
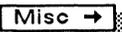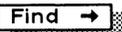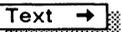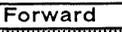
✔ Execute [Misc →] [Find →] [Text →] [Forward] on the Text Location menu. Figure 6-8 shows the Forward Search stickup menu.

```
┌─────────────────────────────────────────────────────┐
│                ┌─────────────────────────┐           │
│                │ Search forward for: " ?: │           │
│                └─────────────────────────┘           │
│      ████████████████████████▲███████████████████    │
│                   ┌───────┐┌────────┐                │
│                   │ Enter ││ Cancel │                │
│                   └───────┘└────────┘                │
│                         ↑                            │
└─────────────────────────────────────────────────────┘
```
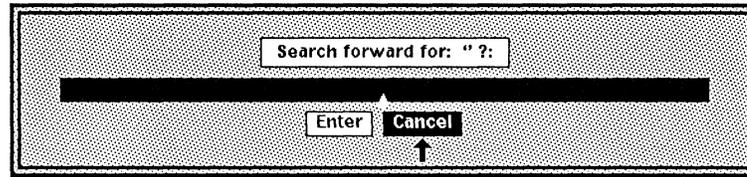
Figure 6-8. Forward Search stickup menu

⌨ Type the search string and **Enter**.

The search begins at the text caret position. If an instance of the search string is found, it is highlighted.

To continue a forward search:

✔ Execute ▭Misc →▭ ▭Find →▭ ▭Next▭ on the Text Selected menu.

If you have not done anything that edited or deselected the previously located string, the search for the next occurrence will begin immediately. If you have deselected or edited the previously located string, the Forward Search stickup will appear with that same string as the default.

The Backward Search

The process you use to search backward in a document is identical to the process you use to search forward. The only difference is the command you use.

To begin a backward search:

✔ Execute ▭Misc →▭ ▭Find →▭ ▭Text →▭ ▭Reverse▭ on the Text Location menu.

⌨ On the stickup menu, type the search string and **Enter**.

The search begins at the text caret position. If an instance of the search string is found, it is highlighted.

To continue a backward search:

✔ Execute ▭Misc →▭ ▭Find →▭ ▭Previous▭ on the Text Selected menu.

Replacing Text

You can replace a string of text:

- one instance at a time
- all instances to the end of the document
- all instances globally

Before replacing one instance at a time using ⌐Misc →⌐ ⌐Replace →⌐ ⌐Single →⌐ , you must search for the text you want to replace. However, you can replace all instances to the end of the document or all instances globally without using the search operation first.

Because a global replace can be difficult to undo, we recommend that you save your document before you replace text globally. Then you can use the **Revert** command to undo the results, if necessary. We also recommend that you define the replacement string very narrowly. For example, if you want to replace *is* with *are* globally, be sure to put a space before *is* so that you do not end up with *haretory* in place of *history*.

Replacement Text

When you initiate the first replace operation in a document, the search string becomes the default replacement string. If you mistakenly accept this default, your document will not be affected since the search and replacement strings are identical. Each time you type a new replacement string, that string becomes the default replacement string for the document. Closing a document clears the default.

As with search strings, when your replacement string includes characters produced by ESC and another key, the escape sequence is usually represented on the Replace stickup menu as ♠ character or as an *<abbreviation>*.

Replacing the Replacement String with a Null String

Sometimes you may want the replacement string to be nothing, that is a **null string**. For example, if you were changing all instances of *Interleaf, Inc.* to *Interleaf.* You could search for, *Inc.* and delete it by making the replacement string a *null string*. The following sequence of keys produces a null string: BACK-SPACE ESC DEL. On a Replace stickup menu, a null string appears as *<nil>*.

If you use this method to delete an entire word, you should delete the word and a space to avoid leaving two spaces between the remaining words.

Replacing Extra Spaces

Some typists use two spaces after punctuation marks like periods and colons. Generally, double spaces are not used in this way in proportionally spaced type such as Interleaf's *Modern* and *Classic*. When a double space occurs at the end of a line, the software converts one of the spaces into a soft return and holds the

other outside the column margin where it will not disrupt the appearance of the column. But the software does not adjust double spaces within a line.

You can search for two spaces (nothing will appear on the stickup menu), and replace them globally with one space. Before globally replacing these double spaces, be sure that you have not purposely placed two consecutive spaces anywhere in your document.

How to Replace Text

There are several methods for starting the replace routine. We give three sets of instructions: one for replacing one-instance-at-a-time, one for performing a global replace operation, and one for replacing all instances to the end of the document. For information on the other methods, see the section *Keyboard Alternatives*.

If you want to replace one instance at a time, use the **Replace Single** command located on the Text Selected Misc submenu. Before using this command you must search for the text.

To replace text one instance at a time:

   ✔ After you find the search string you want to replace, execute the ⌷Misc →⌷ ⌷Replace →⌷ ⌷Single →⌷ ⌷with New⌷ command. Figure 6-9 shows the Replace stickup menu.

```
┌─────────────────────────────────────────────────────┐
│   ┌───────────────────────────────────────────────┐ │
│   │ Replace string: 'elephant', by string: 'elephant' ?: │ │
│   └───────────────────────────────────────────────┐ │
│   ████████████████████████████████████████████████ │
│                    ▲                                 │
│                ┌───────┬────────┐                    │
│                │ Enter │ Cancel │                    │
│                └───────┴────────┘                    │
│                    ↑                                 │
└─────────────────────────────────────────────────────┘
```
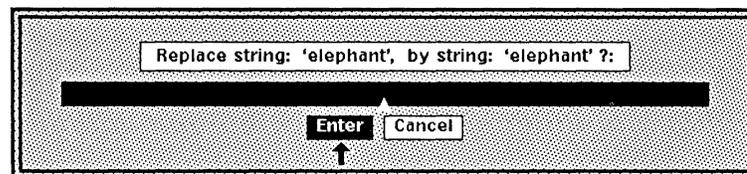
Figure 6-9. Single Replace stickup

   ⌷ On the stickup menu, type the replacement text and **Enter**.
The highlighted text is replaced.

   ✔ If you want to find the next instance of the search string execute the default command, which is ⌷Misc →⌷ ⌷Find →⌷ ⌷Next⌷ or ⌷Previous⌷ depending on the direction of your last search.
The software finds and highlights the next or previous instance.

↙ If you want to replace this instance of the search string execute the default command, [Misc →] [Replace →] [Single →] [with Default] . If you want to skip this instance, execute the [Misc →] [Find →] command.

If you execute the **Replace** *command, the software replaces the search string with the replacement string. If you execute* **Find,** *the software leaves the search string as it is in text and searches for and highlights the next or previous instance. The direction of the continuing search depends on the direction of your last search.*

If you want to replace all instances with one command, you can use the **Replace All** command on either the Text Selected or Text Location menu. The procedure is the same for both menus. You need not use **Find** to search for text before executing **Replace All** or **Replace to End.**

To replace text globally:

↙ Execute the command [Misc →] [Replace →] [All] on either menu.

↙ The first part of the *Global Replace* stickup will appear, prompting for the search string (Figure 6-10). Either **Enter** the default entry or type the search string and **Enter** it.

> Replace all occurrences of string: " ?:
>
> [Enter] [Cancel]
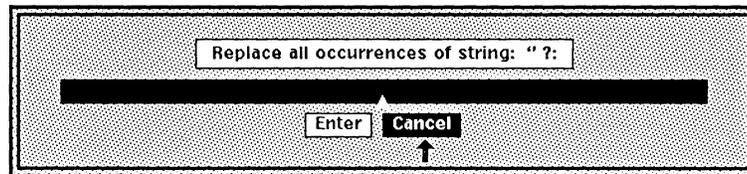
Figure 6-10. First part of the Global Replace stickup

↙ The second part of the *Global Replace* stickup will appear, prompting for the replacement string (Figure 6-11). Either **Enter** the default entry or type the replacement string and **Enter** it.

> Replace all occurrences of string: 'elephant', by string: 'elephant' ?:
>
> [Enter] [Cancel]
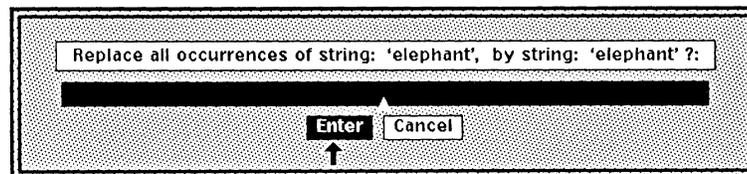
Figure 6-11. Second part of the Global Replace stickup

You will see the message: Replace in Progress. *The software will find every occurrence of the search string and replace it with the replacement string. Once the search and replace operation is completed, you will see the message:* Replaced n occurrence(s), *where "n" is the number of occurrences replaced.*

If you want to replace all instances from the position of the text caret to the end of the document, you can use the **Replace to End** command on either the Text Selected or Text Location menu. The procedure is the same for both menus.

To replace text to the end of the document:

↙ Position the text caret at the point where you want to begin the replacement and execute the command ⌷Misc →⌷ ⌷Replace →⌷ ⌷to End ⌷.

↙ The first part of the *Replace to End* stickup will appear, prompting for the search string (Figure 6-10). Either **Enter** the default entry or type the search string and **Enter** it.

Replace to end occurrences of string: " ?:

Enter Cancel
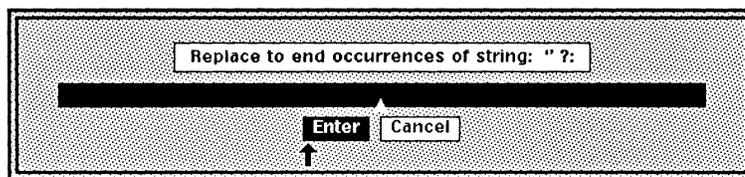
Figure 6-12. First part of the Replace to End stickup

↙ The second part of the *Replace to End* stickup will appear, prompting for the replacement string (Figure 6-11). Either **Enter** the default entry or type the replacement string and **Enter** it.

Replace to end occurrences of string: 'elephant', by string: 'elephant' ?:

Enter Cancel
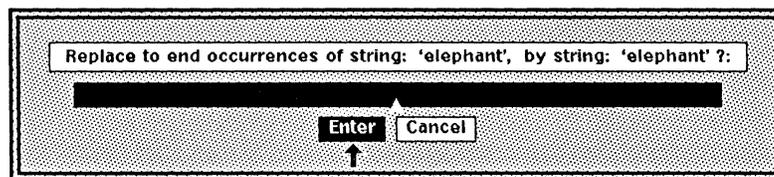
Figure 6-13. Second part of the Replace to End stickup

You will see the message: Replace in Progress. *The software will find every occurrence of the search string between the text caret and the end of the document and replace it with the replacement string. Once the search and replace operation is completed, you will see the message:* Replaced *n* occurrence(s).

Replacing After Searching

If you perform a search operation and then immediately perform a global replace or replace-to-end operation, the software prompts only for the replacement string.

The default search string is the string you entered on the **Find** stickup. Through the course of this operation you will see a total of two stickups.

- When you execute **Find**, a stickup will appear prompting for the search string.
- When you execute **Replace All** or **Replace to End**, a stickup will appear prompting for the replacement string.

Keyboard Alternatives

The following are *forward search* techniques that you can execute from the keyboard:

- To begin a forward search, either hold down the CTRL key and type **s** or press the R3 key. You can also use these keys to confirm a search string or use them repeatedly to continue the search through the document.
- When you receive the wrap stickup, you can confirm that you want to continue the search from the beginning of the document by holding down the CTRL key and typing **s**.

The following are *reverse search* techniques that you can execute from the keyboard:

- To begin a reverse search, either hold down the CTRL key and type **r** or press the BACKSPACE key and then R3. You can also use these keys to confirm a search string or repeatedly to continue a reverse search through the document.
- When you receive the wrap stickup, you can confirm that you want to continue the search from the end of the document by holding down the CTRL key and typing **r**.

The following are *replace* techniques that you can execute from the keyboard:

- To replace one instance at a time, first perform a search. Once you find the string you want to replace, hold down the CTRL key and type **q**. Confirm the default replacement string or type in a new string and confirm the new string. You can continue replacing one instance at a time by repeatedly searching with CTRL **s** and replacing with CTRL **q**. The keys R3 and R6 also enable you to search and replace.
- To replace all instances to the end, hold down the CTRL key and type **x**. Still holding down the CTRL key, type **q**. Confirm the default search and replacement strings or type in new strings and confirm them.

You can use the RETURN key:

- to confirm a search or replacement string
- when you receive the wrap stickup, to confirm that you want to continue searching and/or replacing from the beginning or end of the document.

Stopping a Search or Replace

Occasionally you may begin a search or replace in a large document and then realize that you made a mistake. For example, you may have executed the wrong Find command or mistyped the replacement string. In such circumstances, you may want to cancel the operation and start over. There are two methods of canceling a search or replace in progress. They achieve the same result, but the first is faster.

To stop a search or replace in progress:

⌨ Hold down the CTRL key, and type **g**. (Pressing the L2 key has the same effect.)

or ⌨ Hold down the CTRL key, and type **z**. Then select **Cancel** or type **C** (you must type a capital *C* because lower case *c* represents **Continue**).

When performing a global replace or replace-to-end operation, be aware that the software may have replaced some instances before you canceled the operation.

You can terminate a search or a one-instance-at-a-time search and replace operation after the software has found and selected the object you searched for, by starting to type or by moving the text caret. When you resume a search in the same document, the previous search string will be the default string. You can continue searching for this string by executing ⎍Misc →⎍ ⎍Find →⎍ ⎍Text →⎍ ⎍Forward⎍ or ⎍Misc →⎍ ⎍Find →⎍ ⎍Text →⎍ ⎍Reverse⎍. Or you can search for something new by typing in a new search string and then confirming.

Chapter 7

Autonumbering and Referencing

The autonumbering feature allows you to number a series of objects, like foot-notes, figures, or items in a list. Whenever you change the series, by adding or subtracting an object or by reordering existing objects, the numbers are automatically updated.

With the autoreferencing feature you can create references in text to objects in a numbered series. If the series changes, the references are updated (Figure 7-1).

Autonumbered Section	Reference to Outline

1. Container Icons
   1.1. Folders, Drawers, and Cabinets
   1.2. Special Containers
2. Graphic Icons
	In Section **1.1.** we
	discuss the tools you
	will use to organize
	your documents.

Section Rearranged	Reference Updated

1. Graphic Icons
2. Container Icons
   2.1. Folders, Drawers, and Cabinets
   2.2. Special Containers
	In Section **2.1.** we
	discuss the tools you
	will use to organize
	your documents.

Figure 7-1. Autoreferences change when the referent changes

Autonumber Tokens and Streams

Within a single document, you may want to number many different things, such as simple lists, footnotes, figures, tables, and outlines. In the Interleaf publishing software each of these would be represented by a different **stream** of numbers. For each stream, the numbering scheme, the starting value, and other properties may vary.

In each place that you want a number to appear, you create a **token**. A token looks just like any other number, but its value changes as its position in the stream changes.

Creating Autonumber Tokens

Every document contains two default autonumber streams, *List* and *Outline*. Figure 7-2 shows tokens from both the *List* and the *Outline* stream.

List	Outline
1. Animal Proteins	I. Proteins
	I.A. Red Meat
2. Carbohydrates	I.A.1. Beef
	I.A.2. Pork
3. Fruits and Vegetables	I.A.3. Mutton
	I.B. Fish
4. Dairy Products	I.C. Poultry

Figure 7-2. Sample tokens from the default autonumber streams

One important difference between the two streams is that *List* has only one level, while *Outline* has three. The levels allow you to create three kinds of headings in an outline. Figure 7-2 shows all three levels.

To create a token, execute the `Create →` `Auto # →` `Streamname →` `Insert Level X →` command. As Figure 7-3 shows, the List stream has one level.

Figure 7-3. Creating an autonumber token

The Outline stream has three levels from which to choose (Figure 7-4).
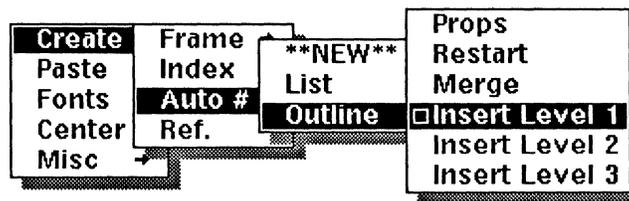
Figure 7-4. Creating a token in a multi-level autonumber stream

Please note that when you are creating tokens from a multi-level stream, you cannot skip levels. For example, if you create a Level 1 token and then try to create a Level 3 token, the software will instead create a Level 2 token.

Properties of Autonumber Streams

Using the Autonumber Stream property sheet, you can change the properties of existing streams and create new ones.

To see the property sheet for an existing autonumber stream:

🖅 Position the mouse cursor anywhere in text.

🖅 Execute the [Create →] [Auto # →] [*Streamname* →] [Props] command.

Figure 7-5 and Figure 7-6 show the property sheets for the List *stream and the* Outline *stream.*

```
Autonumber Stream Properties

Stream Name:          [    List    ]              ↑
Levels:               [ 1 ]
   Level One
      Symbol Type     [ARABIC]              ↑
      Prefix          [          ]
      Suffix          [      .   ]
      Starting Value  [ 1 ]
      Last Only       Yes No
      Show            Yes No
                                         ↓
                   ← ▒▒▒▒▒▒▒▒▒▒▒ →
```

Figure 7-5. List Stream property sheet

```
Autonumber Stream Properties

Stream Name:          [   Outline  ]              ↑
Levels:               [ 3 ]
   Level One
      Symbol Type     [UPPER ROMAN]
      Prefix          [          ]
      Suffix          [      .   ]
      Starting Value  [ 1 ]
      Last Only       Yes No
      Show            Yes No
   Level Two
      Symbol Type     [UPPER ALPHA]       ↑
      Prefix          [          ]
      Suffix          [      .   ]
      Starting Value  [ 1 ]
      Last Only       Yes No
      Show            Yes No
   Level Three
      Symbol Type     [ARABIC]
      Prefix          [          ]
      Suffix          [      .   ]
      Starting Value  [ 1 ]
      Last Only       Yes No
      Show            Yes No
                                          ↓
                   ← ▒▒▒▒▒▒▒▒▒▒▒ →
```

Figure 7-6. Outline Stream property sheet

To open a property sheet for a new autonumber stream:

🖅 Position the mouse cursor anywhere in text.

🖅 Execute the [Create →] [Auto # →] [**NEW**] command.

Figure 7-7 shows the property sheet for a newly created autonumber stream.

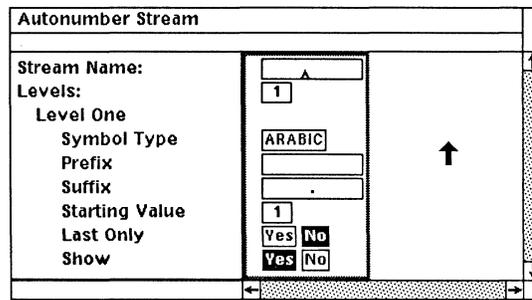| Autonumber Stream	
Stream Name:	
Levels:	`1`
Level One	
Symbol Type	`ARABIC`
Prefix	
Suffix	.
Starting Value	`1`
Last Only	`Yes` `No`
Show	`Yes` `No`

Figure 7-7. Property sheet for creating a new stream

Stream Name

The Stream Name box tells you which stream's properties you are looking at. In the property sheet for a new autonumber stream, this box will be blank.

If you change the name of a stream, all the tokens belonging to that stream will be updated, but the old stream name will still appear on the popup menu. The reason for this is that autonumber streams are shared between all documents in a book, and if the old stream name is used in another document of a book, you would not want it to be changed there. (Please refer to the chapter *Large Documents in Books* for more information on autonumber streams in books.)

If you enter the name of an existing stream, you will see the following error message: *This stream name already exists. Please change it.*

Levels

The number in the Levels box tells you how many levels are defined for this stream. No more than eight levels can be defined for each stream: if you enter a number larger than eight in the Levels box, you will see the following error message: *The number of levels must be between 1 and 8.*

The format of each level is defined separately, so you will see a separate set of properties for each level in the stream (Figure 7-8).

```
Level One
    Symbol Type          UPPER ROMAN
    Prefix               [            ]
    Suffix               [     .      ]
    Starting Value       [ 1 ]
    Last Only            Yes  No
    Show                 Yes  No
Level Two
    Symbol Type          UPPER ALPHA
    Prefix               [            ]
    Suffix               [     .      ]
    Starting Value       [ 1 ]
    Last Only            Yes  No
    Show                 Yes  No
```

Figure 7-8. Properties for two levels of an autonumber stream

The next sections describe the properties you can set for each level of an autonumber stream.

Symbol Type

Use the *Symbol Type* line to determine what kind of numbering scheme you will be using for that particular level.

There are five symbol types: arabic (1, 2, 3...), upper roman (I, II, III...), lower roman (i, ii, iii...), upper alpha (A, B, C...) and lower alpha (a, b, c...). In the Outline stream, for example, the first level is upper roman (I.), the second is upper alpha (I.A.) and the third is arabic (I.A.1.).

To select the symbol type, point the mouse cursor into the Symbol Type box and click the left or right button until the type you want appears in the box.

Prefix and Suffix

Use the *Prefix* and *Suffix* lines at each level to enter text that should always appear before and after that level of autonumber token. You might, for example, want to place a period between each level. Or you might use the prefix "Figure " for a single level stream that you will use to number illustrations. (The space after "Figure" is necessary to prevent the prefix and the number itself from running together.)

The prefix and suffix may each contain up to 15 characters.

Starting Value

Use the *Starting Value* line to specify the number at which that level of the stream will start counting. The starting value for the List stream is 1, so the first three tokens you create in it will be displayed as 1., 2., and 3. If you reset the starting value to 7, the first three tokens will be displayed as 7., 8., and 9.

Whatever the symbol type you have chosen, the number in the Starting Value box will always be displayed in Arabic. The maximum starting value is 9999.

Last Only

When the value of the *Last Only* line for a particular level is **Yes**, the preceding levels of the token will not be displayed.

For example, if you create a third level token from the Outline stream, you will see the current values of the first and second level too, since the setting of *Last Only* for the third level is **No**. If you change that setting to **Yes**, you will only see the value of the last level.

Figure 7-9 shows how you would use *Last Only* to create two different kinds of outlines.

```
┌─────────────────────────────────────────────────────────┐
│                                                         │
│   Last Only   No          Last Only   Yes              │
│                                                         │
│       I.                      I.                        │
│           I.A.                   A.                     │
│       II.                     II.                       │
│           II.A.                  A.                     │
│               II.A.1.               1.                  │
│               II.A.2.               2.                  │
│           II.B.                  B.                     │
│                                                         │
└─────────────────────────────────────────────────────────┘
```

Figure 7-9. Effect of Last Only on a three-level stream

Show

The *Show* line in each level allows you to create invisible tokens. If, for example, the **Yes** box is selected for the *Show* line in the first two levels, and the **No** box is selected for the third level, Level 1 and Level 2 tokens will be visible, but Level 3 tokens will be invisible.

The most common use of the Show property is to create an entire stream of invisible tokens. These invisible tokens are then used with Autoreferencing to create references to items that are not visibly numbered. There is more about this feature in the section *Autoreferences*.

Creating a New Stream: Examples

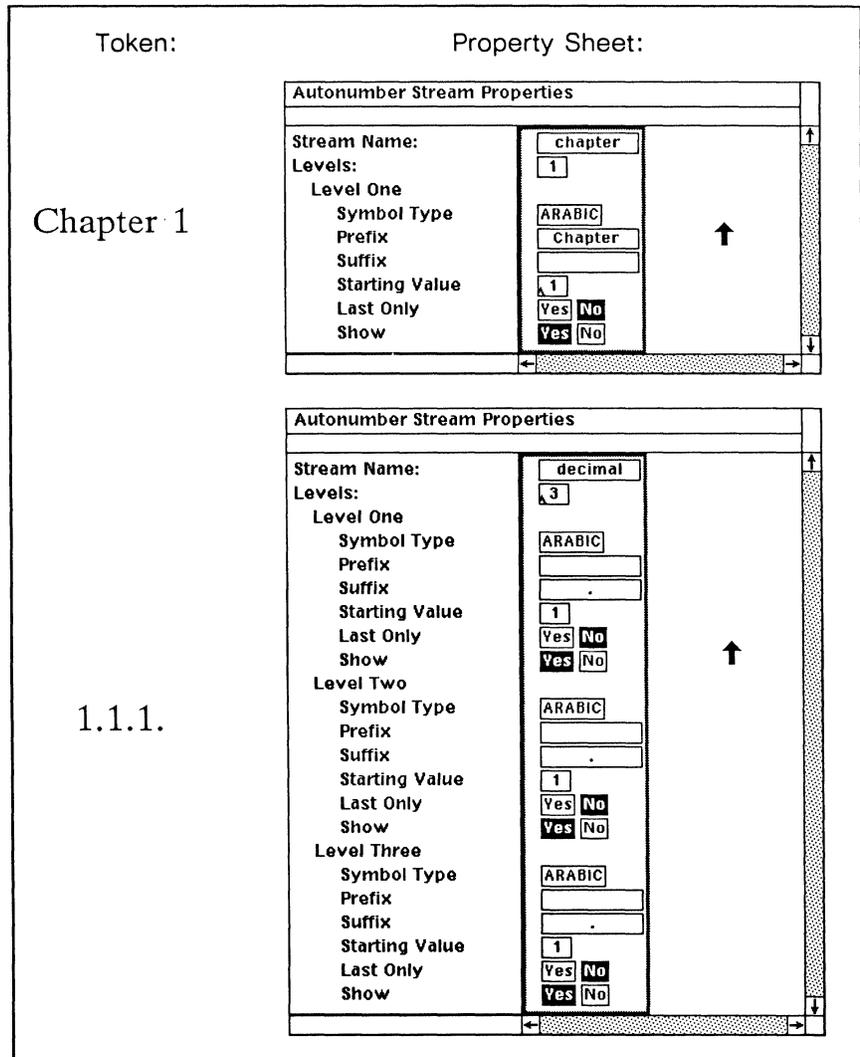Figure 7-10 shows two examples of autonumbered streams and the property sheets that control their appearance.

```
┌──────────────────────────────────────────────────────────────────────┐
│      Token:                      Property Sheet:                       │
│                          ┌────────────────────────────────────────┐    │
│                          │ Autonumber Stream Properties          ┃│    │
│                          │                                       ┃│    │
│                          │ Stream Name:      ┌─chapter─┐        ▲┃│    │
│                          │ Levels:           │ 1 │               ┃│    │
│                          │   Level One                          ▒│    │
│   Chapter 1              │     Symbol Type   ┌ARABIC┐           ▒│    │
│                          │     Prefix        ┌Chapter┐     ↑    ▒│    │
│                          │     Suffix        ┌──────┐          ▒│    │
│                          │     Starting Value  1                ▒│    │
│                          │     Last Only     Yes █No█           ▒│    │
│                          │     Show          █Yes█ No           ▼│    │
│                          │              ←░░░░░░░░░░░░░░░░░░→       │    │
│                          └────────────────────────────────────────┘    │
│                                                                        │
│                          ┌────────────────────────────────────────┐    │
│                          │ Autonumber Stream Properties          ┃│    │
│                          │                                       ┃│    │
│                          │ Stream Name:      ┌─decimal─┐        ▲┃│    │
│                          │ Levels:           │ 3 │               ┃│    │
│                          │   Level One                          ▒│    │
│                          │     Symbol Type   ┌ARABIC┐           ▒│    │
│                          │     Prefix        ┌──────┐           ▒│    │
│                          │     Suffix        ┌ . ───┐           ▒│    │
│                          │     Starting Value  1                ▒│    │
│                          │     Last Only     Yes █No█           ▒│    │
│                          │     Show          █Yes█ No      ↑    ▒│    │
│                          │   Level Two                          ▒│    │
│   1.1.1.                 │     Symbol Type   ┌ARABIC┐           ▒│    │
│                          │     Prefix        ┌──────┐           ▒│    │
│                          │     Suffix        ┌ . ───┐           ▒│    │
│                          │     Starting Value  1                ▒│    │
│                          │     Last Only     Yes █No█           ▒│    │
│                          │     Show          █Yes█ No           ▒│    │
│                          │   Level Three                        ▒│    │
│                          │     Symbol Type   ┌ARABIC┐           ▒│    │
│                          │     Prefix        ┌──────┐           ▒│    │
│                          │     Suffix        ┌ . ───┐           ▒│    │
│                          │     Starting Value  1                ▒│    │
│                          │     Last Only     Yes █No█           ▒│    │
│                          │     Show          █Yes█ No          ▼│    │
│                          │              ←░░░░░░░░░░░░░░░░░░→       │    │
│                          └────────────────────────────────────────┘    │
└──────────────────────────────────────────────────────────────────────┘
```

Figure 7-10. Sample autonumbers and Autonumber Stream property sheets

Associating a Stream with a Component

Sometimes you may want to begin all instances of a particular component with an autonumber token. By adjusting the Component property sheet, you can make the publishing software insert the token.

To associate a stream with a component:

🗸 **Open** the Component property sheet.

🗸 With the **left** button, select the box labeled **Custom**.
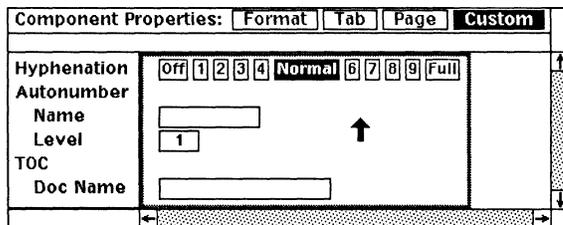
Figure 7-11 shows the Component Custom property sheet.

Figure 7-11. Component Custom property sheet

✔ Enter the name of the stream in the Autonumber Name box.

✔ Enter the level of the stream in the Autonumber Level box.

✔ **Global Apply** and **Close**.

For example, we use this feature in the Interleaf manuals to automatically insert an autonumber token from the *label* stream in the *label* component after each illustration. Figure 7-12 shows the property sheet for the *label* stream in chapter 7, called *07label*, and Figure 7-13 shows the sheet for the *label* component.
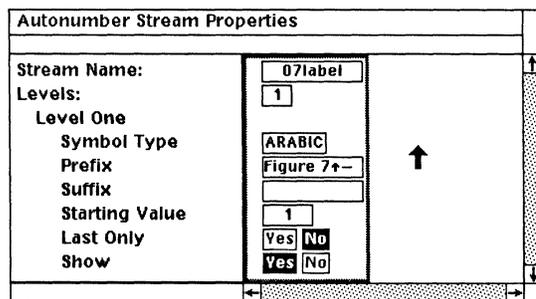
Figure 7-12. Property sheet for the label autonumber stream
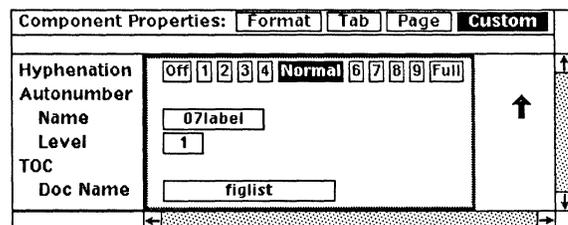
Figure 7-13. Custom sheet for the label component

If you later change the name of the stream associated with the component, the Component Property sheet will not be updated automatically. You will need to change the name on that sheet as well.

When you associate an autonumber stream with a component, an autonumber token will be inserted at the beginning of all existing components of that kind. If you had previously numbered the components by inserting autonumber tokens

manually, you will now find two tokens at the beginning of each component. You can cut either one of the tokens to correct the problem.

Similarly, if you **Change** a component that has an autonumber token associated with it to another component that doesn't, the autonumber token will disappear.

Properties of Autonumber Tokens

In addition to the property sheet for the stream, each autonumber token has its own property sheet. To see it, select the token and execute the **Props** command (Figure 7-14).

Autonumber Token Properties		
Stream:	Outline	
Level:	1	
Tag:		
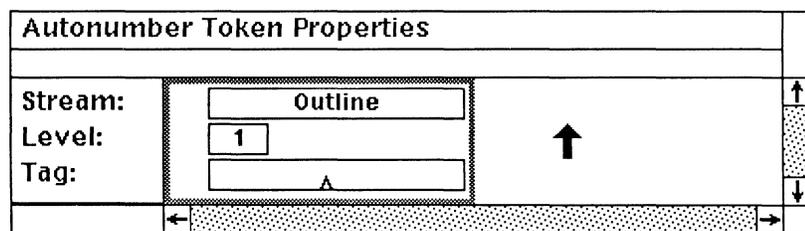
Figure 7-14. Autonumber Token property sheet

If you do not see the Autonumber Token property sheet, you may have selected an invisible token in addition to the autonumber token. Try executing the
[Misc →] [Show →] [Index Tokens] or [Misc →] [Show →] [Frame Anchors] command.

The *Stream* line shows you which stream the token belongs to. You can change the stream with which the token is associated by entering a stream name in the Type box. The stream must already exist; if you type the name of a non-existent stream and **Apply**, you will see the following error message in the status line: *Chosen autonumber stream does not exist.*

And the next time you hold down the middle button you will see the Apply popup menu again.

The *Level* line shows what level the token is. For streams that have only one level (List, for example) the number in the Level box will always be 1. If the token belongs to a stream that has more than one level, you can change the level of the token by entering another number in the Level box.

If you enter a level that does not exist for the current stream, you will see the following error message in the status line: *Level is inappropriate.*

And you will continue to see the Apply popup menu when you hold down the middle button.

The *Tag* line, used with the Autoreferencing feature, is explained later in the section *Autoreferences*. When you create a token, the Tag box is always blank.

Restart and Merge

If you want to number two sets of objects using exactly the same format, you do not have to create two autonumber streams. Instead, you can use the **Restart** command. You can issue the **Restart** command in the middle of a stream, to split the stream in two. Or you can issue it after the last token in a stream, to start a new one.

You can execute the **Restart** command an unlimited number of times within a single stream, creating, for example, as many new lists as you like. All of these lists, though, will still belong to the same stream, and will share that stream's properties.

To split one numbered outline in two:

✔ Position the text caret between the last token of one list and the first token of the other.

The autonumber token after the caret must be Level 1. If it is not, you will see the error message: You can Restart only before a Level 1 token or at the end of a stream.

✔ Execute the [Create →] [Auto # →] [Outline →] [Restart] command.

The tokens that follow the text caret will be renumbered, beginning with the Starting Value *set in the Autonumber Stream property sheet.*

The same procedure can be used with any stream: just substitute the name of that stream for Outline *in the procedure above.*

To begin a new outline with the same format as the preceding one:

✔ Position the text caret after the last token in the list.

✔ Execute the [Create →] [Auto # →] [Outline →] [Restart] command.

The publishing software will insert a Level 1 token displaying the Starting Value you set in the Autonumber Stream property sheet.

The **Merge** command allows you to join two sets of numbers that belong to the same stream. It undoes the **Restart** command.

To combine two numbered lists into one:

    ↙ Position the mouse cursor anywhere after the first token in the second list.

    ↙ Execute the [Create →] [Auto # →] [List →] [Merge] command.

The tokens in the second list will be renumbered to make them continuous with the first list.

This procedure can be used with any stream.

Removing Autonumber Streams

In the process of setting up numbering streams for your document, you may create streams that you decide not to use. As long as the stream is completely unused — there are no tokens belonging to it — you can purge it from the document.

The menu used to purge autonumber streams is the same menu used to purge component and frame masters (Figure 7-15).
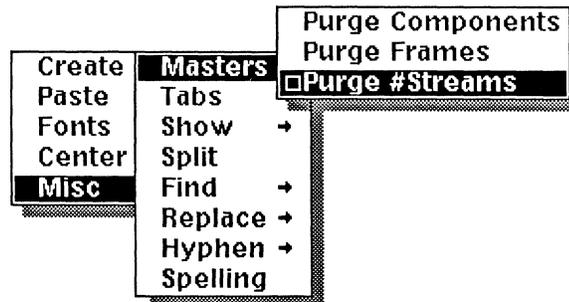
Figure 7-15. Purging unused autonumber streams

When you execute the **Purge #Streams** command, you will be asked to confirm your action (Figure 7-16).
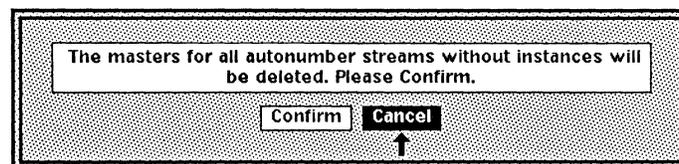
Figure 7-16. Confirmation stickup for purging streams

The *List* and *Outline* streams cannot be purged.

Autonumbers in Microdocuments

Autonumber streams flow throughout a document. If you create two tokens from the *List* stream in the main document, and then create a third in a microdocument, all three tokens are considered part of the same stream, and will be numbered accordingly.

Within a single microdocument, autonumber tokens are numbered according to the flow of the text, just as they are in the larger document. However, if you have several microdocuments in the same frame, each containing autonumbers from the same stream, the microdocument at the back will be numbered first, and the microdocument in front will be numbered last.

If the numbers in your microdocuments do not appear in the sequence you want, you can change the sequence by changing the relative positions of the microdocuments using the diagramming **Front** and **Back** commands (see the chapter *Basic Diagramming Concepts* in the *Reference Manual*, Volume 2).

If you have autonumbers in microdocuments in several different frames, the microdocuments will be numbered in anchor order. If you are using several kinds of frames on a single page you will need to remember this, because the order of the anchors may be different from the order of the frames.

Autoreferences

An autoreference is a token whose value is linked either to the current value of an autonumber token or to the number of the page on which the autonumber currently appears. Autoreferences can be created using either the **Create Ref.** command or the **Copy Ref.** command.

To automatically create a reference to an autonumber:

✔ Select the autonumber token you want to refer to.

✔ Execute the **Copy Ref.** command.

Figure 7-17 shows the location of the Copy Ref. command.

Cut	Normal
Copy	**☐Ref.**
Fonts →	
Props	
Deselect	
Misc →	
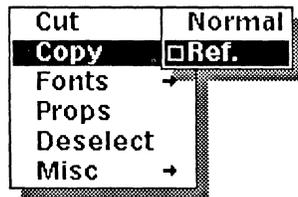
Figure 7-17. Automatically creating a reference

In the document status line you will see the message: The Ref. has been created on the clipboard.

✔ Position the text caret where you want the autoreference and execute the **Paste** command.

The font of the autoreference token will be the same as the font of the text into which you pasted it.

In order to tell which autonumber token a reference is linked to, the publishing software looks at the **tags** to see if they match up. A tag is just a name for a particular token. It can be assigned by the system, or you can choose it yourself. If a reference token has the same tag as an autonumber token, the software knows the reference must refer to that autonumber.

Two autonumber tokens in one document (or one book) cannot have the same tag. If you try to enter a tag that has already been used, you will see the error message: *Tag 'tagname' is already in use. Please pick another name.*

Two references *can* have the same tag, if they refer to the same autonumber. You can make as many reference as you like to one autonumber.

If you have a reference with a tag that doesn't match the tag of any autonumber, the reference will be displayed as "NO TAG".

You can see the tag by opening the property sheet of either the autonumber token or the autoreference token. Figure 7-18 shows the property sheet for an automatically created autoreference.
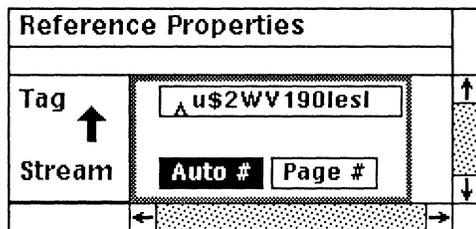
Figure 7-18. Reference sheet showing automatically created tag

The tag consists of twelve characters: eight characters of random letters, numbers, and symbols plus the first four characters of your login name. If your login name has fewer than five characters, your full login name will be added to the tag.

The tags the software assigns are not easy to remember, and there will be times that you need tags that are meaningful and easy to reproduce. In this case you can enter the tags for the autonumber token and the autoreference token in the property sheets.

To create tags for an autoreference manually:

✓ Select the autonumber token you want to refer to and open its property sheet.

Figure 7-19 shows the Autonumber Token sheet.
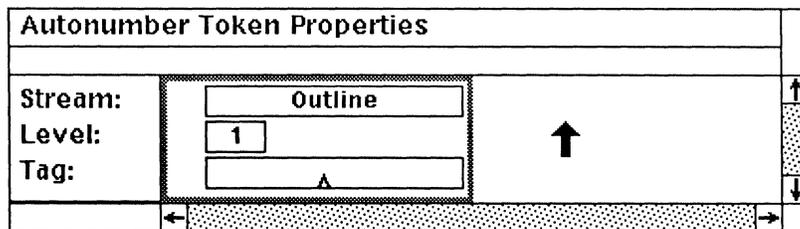
Figure 7-19. Autonumber Token property sheet

✓ Enter a unique name in the Tag box, **Apply** and **Close**.

✓ Position the text caret where you want the autoreference token and execute the **Create Ref.** command.

Figure 7-20 shows the location of the Create Ref. command.
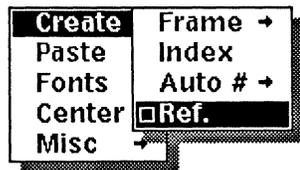
Figure 7-20. Manually creating a reference token

The Autoreference Token sheet will be opened automatically (Figure 7-21). The Tag box will be empty.

```
┌─────────────────────────────────────┐
│ Reference Properties              │ │
├─────────────────────────────────────┤
│         ░░░░░░░░░░░░░░░░░░░░░░░  │ ↑ │
│ Tag   ░│              Λ      │░  │   │
│       ░░░░░░░░░░░░░░░░░░░░░░░ ↑ │░ │
│ Stream ░ Auto #   Page #   ░    │░ │
│       ░░░░░░░░░░░░░░░░░░░░░░░   │ ↓ │
│        ← ░░░░░░░░░░░░░░░░░ →     │   │
└─────────────────────────────────────┘
```

Figure 7-21. Reference property sheet

↙ In the Tag box, enter the same name you entered in the Autonumber Token property sheet.

Another variation on this procedure is to type the tag for the autonumber token by hand, and then to select the token and execute the **Copy Ref.** command.

References to Page Numbers

In the phrase, "Figure 21, on page 47,...", both the figure number and the page number can be references to the same autonumber token. In the previous section we discussed the procedures for creating the reference to the figure number. This section will explain the one additional step necessary to create a reference to the page number.

To create a reference to the current page number of the autonumber:

↙ Create a reference to an autonumber token.

If there is already a reference to the value of the autonumber token, the easiest way to perform this step is to select the reference, execute the **Copy Normal** *command and* **Paste** *it.*

↙ Open the property sheet of the reference and select the Page # box.

References to Any Page Number

You can use referencing and autonumbering to refer to any page in a document by creating invisible autonumber tokens on the page you want to refer to.

To create a reference to any page number:

↙ **Create** a new autonumber stream. Call it *invisible*, give it one level and set Show to **Yes** for that level.

↙ On the page you want to refer to, **Create** an autonumber token from the *invisible* stream.

    ✔ Select the token and execute the **Copy Ref.** command.

    ✔ **Paste** the token where you want the reference and open its prop sheet. In the *Stream* line, select the Page # box.

    ✔ Open the property sheet for the *invisible* stream and set Show to **No** for that level.

For example, suppose you wanted to refer to "the *Procedures* section, beginning on page 13". After you had set up the invisible stream, you would create a token from that stream in the component that headed the *Procedures* section. You would create a reference to that invisible token. Then you would just paste it and edit its property sheet to turn on the Page # box.

If you want to refer to the page on which the reference token will appear, there is an even simpler method.

To create a reference to the current page:

    ✔ Execute the **Create Ref.** command.

        The Autoreference Token sheet will be opened automatically. The Auto# box will be selected, but as long as a reference has no tag, it refers to the current page.

    ✔ **Close** the property sheet without entering a tag.

Chapter 8

Typography and Typefaces

If you are a new user of the Interleaf publishing software, this may be your first exposure to multiple typefaces and type sizes, proportionally spaced type, and varied line and word spacing. These new capabilities open up a tremendous number of possibilities for producing more attractive, more readable documents. They also require choices that are not necessary when you are using a typewriter or a daisywheel printer.

This chapter is divided into two major sections:

- The first section offers guidelines for making basic typographic choices.
- The second section lists the character sets in each of the standard Interleaf typefaces.

If you need to find out how to execute the commands that produce different kinds of type, spacing, and page layout you will find the information in the chapters *Text Processing* and *Page Makeup*.

Typography

Readability and Legibility

Pick up almost any typography book and you will see two principles emphasized: readability and legibility. *Readability* refers to the ease of reading an entire page. *Legibility* refers to the ease of recognizing each letter or word. No matter what else you may read about graphic design, here or in more formal treatments of the subject, these two principles come first.

Units of Measure: Picas and Points

While inches and millimeters are used for some typographic measurements, two additional units of measurement are more common in the graphic arts.

One pica is approximately one sixth of an inch. Picas are used in the graphic arts to measure line width, page length, and sometimes spacing between components or paragraphs.

One point is one twelfth of a pica and thus approximately one 72nd of an inch. Points are often used to specify type size and the spacing between lines.

However, in the Interleaf publishing software, line spacing is specified as a multiple of the line height.

In the *TextAids* folder you will find a document, *PtsPicasToInches*, showing the inch equivalents for a range of pica and point measurements. *TextAids* is in the *Documentation* drawer of the *System* cabinet.

Type Specification

This sentence is set in 12-point Classic italic. In typography, various terms are used, with more or less precision, to describe different kinds of characters and to group characters of similar style.

Size is specified in points and usually varies from 6 to 72 points. However, this does not mean that, for instance, characters in 12-point type are 12 points high. To understand typographic sizing, you must first understand the structure of a character. Figure 8-1 illustrates the common typographic terms used to describe a character's structure.
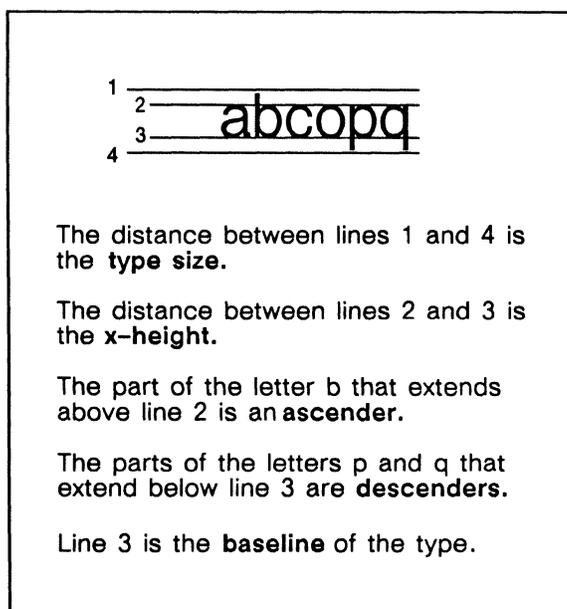
Figure 8-1. Type structure and size

Size is generally determined by the distance from the tallest ascender to the deepest descender plus a small amount of space that varies from style to style. The remaining characters in that style are then designed so that all of the x-heights of the characters in the same size are the same. Uppercase letters are designed to complement their lowercase counterparts. They are constant in height within a particular size and style.

Family is usually used to describe the overall type style, within which there can be variations of slant, weight, and size. In the Interleaf software, *Modern, Classic, Typewriter, Symbols, Greek,* and the various *Math* choices are all type *families.*

Other type families are available. You can purchase the optional families listed in *The Interleaf Typeface Catalog*. And, if you have purchased the typesetter option, you will be able to use the fonts you bought with the typesetter.

Slant and **weight** further distinguish type styles within families. Type of one family may be available in a *book* weight, which is a standard weight used in text, and a *bold* weight, in which the characters are thicker and so appear larger, even though they are the same size. Figure 8-2 shows the difference between bold and book weight in otherwise similar type.

Abc   **Abc**

Book   Bold
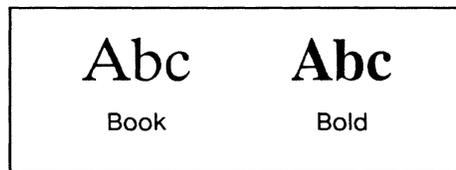
Figure 8-2. Type weight

Also, a type family can have a roman version, in which the characters have no slant, and an *italic* version, in which the characters slant to the right. In general, if no other qualifying words appear after the family name, the type is roman and book weight. Figure 8-3 shows the difference between roman and italic in otherwise similar type.

Abc   *Abc*
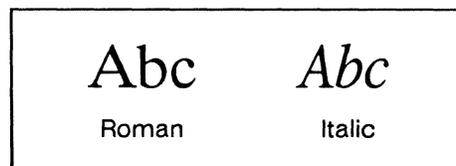
Roman   Italic

Figure 8-3. Type slant

Interleaf's *Modern* and *Classic* type families each have three variations (other than size): roman/book, bold (actually roman/bold), and italic (actually italic/book). Our *Typewriter* type family has roman/book type only.

Combined bold/italic type is available in the optional type families or through the typesetter option.

The term **typeface** is used to describe the family plus the specific slant or weight variation. For instance, *Modern italic* is a typeface.

If someone were to ask you what **font** you are using, you would specify the size of the type, its family, and any variation within the typeface. Thus *10-point Modern italic* describes a particular font and tells someone precisely what he or she is going to see.

Serif and Sans Serif Type

A **serif** is a short ornament or bar at the end of the main stroke of a letter. Serif typefaces, such as this one, have serifs on many of their letters. In the Interleaf publishing software, *Classic* and *Typewriter* are both serif typefaces. **Sans serif** typefaces do not have these ornaments on any of their letters. In the standard Interleaf publishing software, *Modern* is the sans serif typeface. Figure 8-4 shows serif and sans serif typefaces with the same size, weight and slant.

Abc   Abc
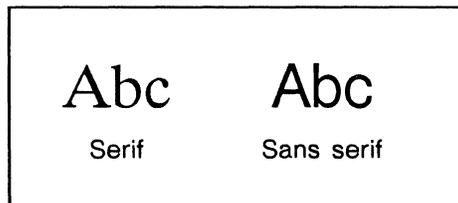
Serif   Sans serif

Figure 8-4. Serif and sans serif typefaces

Choosing Type

At first, you will probably find it easier to create pleasing pages if you use serif type *(Classic)* for text — serif type is considered easier to read than sans serif when used in large blocks of text. Sans serif type *(Modern)* is commonly used for headlines, subheadings, tables, and special display type.

However, many newspapers use serif type for the headlines and subheadings, and some text is set in sans serif. Look through books, newspapers, and magazines that you find visually appealing and easy to read. Notice the typefaces they use and the *combinations* that work best.

You should be able to closely approximate many of the type combinations you see elsewhere. *Helvetica* and *Times Roman* are two of the most commonly used typefaces. Interleaf's *Modern* is similar to *Helvetica* and its variations. Interleaf's *Classic* is similar to *Times Roman* and its variations.

Keep it Simple

You will probably notice that most publications keep the number of type styles on any one page to a minimum. Going from a device where there is only one type style in one size to a system that offers many type styles and sizes can lead the enthusiast into a typographic binge.

After you are through experimenting, fight temptation. Far from being more interesting, frequent type changes usually produce pages that are confusing, ugly, and hard to read.

Choose a text type in the 6- to 12-point range (10 or 12 pt. in most cases) in *Classic*, subheadings in the 12- to 18-point range in *Modern*, and headings in the 18- to 24-point range in *Modern*. If you need larger display type, you can use the

type in the *Fonts* folder of the *Graphics* cabinet on your desktop to build special headings within a frame. Also, if you need more than one level of subheading, select a *sub-subheading* that falls between the size of your major subheading and the size of the text type — 14-point, for instance.

Use bold and italic sparingly for emphasis. Italic type is commonly used to emphasize words that would be underlined in less sophisticated typography; for example figure captions and publication titles are often italicized. Bold type is often used for headings and subheadings, and for special emphasis of particular words. Entire paragraphs of bold type are hard to read. Entire paragraphs of italic type are not quite as hard to read as all-bold paragraphs but are less readable than roman, book-weight paragraphs.

Text Alignment: Justified Columns

Because they are made up of lines of type that all have exactly the same length, justified columns have smooth boundaries on both their left and right sides. Most text columns are either justified or flush left / ragged right. Figure 8-5 shows the same block of text with two different alignments.

Last night's performance by the Leaftones makes it clear that this polka–rock band is now ready to vault from its cult status to the top ranks of the entertainment world.

Flush Left, Ragged Right

Last night's performance by the Leaftones makes it clear that this polka–rock band is now ready to vault from its cult status to the top ranks of the entertainment world.

Justified

Figure 8-5. Text alignment: ragged and justified

Although it is primarily a page makeup issue, the choice of justified or ragged columns also affects hyphenation and word spacing.With justified type in a narrow column width, you want more frequent hyphenation than in ragged right type so that you do not end up with huge spaces between words. But you will not want more than four consecutive lines ending with hyphens, and some designers set a limit of two hyphens.

You have to trade off column width, choice of ragged or justified right margins, and hyphenation control to produce the best overall column. Column margins, column alignment, and hyphenation frequency can be varied on the *Component* property sheet. The number of consecutive hyphenated line endings can be directly controlled on the *Custom* sheet of the *Document Page* property sheet.

More information on justified columns and also about the relationship between hyphenation and column design can be found in the *Page Design* chapter.

Line Spacing

Line spacing is the space from the baseline of one line to the baseline of the line below it.

On many systems, **line spacing** (also called **leading**) is specified in points. For example, 18/24pt. means that lines of 18-point type are spaced, baseline-to-baseline, in vertical increments of 24 points, as in Figure 8-6.
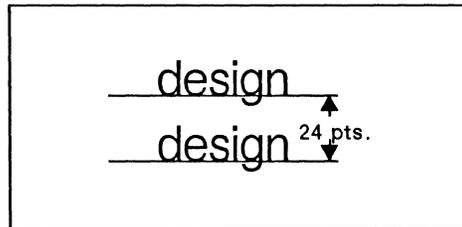
Figure 8-6. Line spacing

In the Interleaf publishing software, line spacing is specified as a multiple of type size.

In the *TextAids* folder in the *LeadingToLineSpacing* document, you will find a table which converts leading in points to leading in line multiples. *TextAids* is in the *Documentation* drawer of the *System* cabinet.

For most text produced using this publishing software, line spacing will be 1.1 to 1.35 times the line height — the default is 1.31. If you want double-spaced type, you enter 2 in the Line Spacing box on the Component property sheet.

Vertical justification is a set of features that is used to adjust certain spaces in an attempt to make all of the pages in a document (and all of the columns on a page) of equal depth. Occasionally, **feathering** is used in the vertical justification process. Feathering involves the expansion of line spacing by no more than 7.5 percent of its specified value. It is not common and is used only when other space adjustments do not accomplish vertical justification.

Character Spacing

Unlike typewriter type, most typeset characters are proportionally spaced; that is, the width that a letter takes up varies depending upon the space the letter actually needs. For instance, the width taken up by an *i* is much less than that taken up by an *m* of the same type size and style. Proportional space eliminates excess letter spacing and character distortion, common characteristics of typewritten copy. These factors help make typeset text not only aesthetically superior but also more economical because of the greater amount of information that can be put into the same space.

All but one of the Interleaf type families are proportionally spaced. So that it more genuinely simulates typewritten copy, Interleaf's *Typewriter* family retains monospacing (all characters take up the same space).

Horizontal Spacing

Word Spacing

As a general rule, spacing between words should not be greater than spacing between lines. Also, the longer the column or the wider the line, the greater the line spacing should be.

Since spacing between words is often variable while spacing between lines is usually constant within a component, you will have to determine the relationship between the two visually on a sample printed page.

Soft and Hard Spaces

The Interleaf software provides five different space characters in two categories: soft spaces, which can vary in width, and hard spaces, which maintain a fixed value. Figure 8-7 shows the relative width of the five different spaces. The font used is 12-point *Modern*.

```
This space is a soft word space.
This space is a hard word space.
This space  is a 12-point em space.
This space is a 12-point en space.
This space is a 12-point thin space.
```
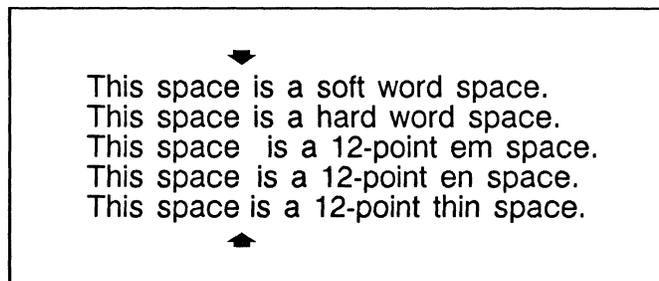
Figure 8-7. Horizontal spaces

Soft Word Space

A **soft word space** varies in width in a *justified* column depending on the number of characters on a line. The range within which it varies is determined by the point size. In a *ragged* column a soft word space has a fixed width chosen to provide the most pleasing spacing in that typeface and size.

When it occurs at the end of a line, a soft space becomes a **soft return** and the word following it begins the next line. In the Interleaf publishing software, you produce a soft word space by pressing the the **SPACE BAR** key.

Soft spaces are the most commonly used spaces and are usually used between words in a sentence.

Hard Word Space

A **hard word space** does not vary in width in a column, even when the column is justified. The width of a hard word space is identical to the width of a soft word space of the same font in a ragged column. In the publishing software, you produce a hard word space by pressing and releasing the ESC key and then pressing the SPACE BAR key.

Hard word spaces are often used to prevent line breaks between parts of a name or other multi-word terms that should be treated as a single word (or string of characters).

Em Space

An **em space** in the Interleaf publishing software is approximately the size of the font; that is, if you are in 10-point *Modern* type, the em space width will be approximately 10 points. In the publishing software, the em space is more directly based on the width of the widest characters in each font (the *em dash* and the *percent symbol*). You produce an em space by pressing and releasing the ESC key and then typing the letter **d**.

The value of the em space is often used for paragraph indents. Figure 8-8 converts em space values for different type sizes to approximate inch equivalents.

An em space at	=	Inches
6pt.		.08
8pt.		.11
10pt.		.14
12pt.		.17
14pt.		.19
18pt.		.25
24pt.		.33

Figure 8-8. Em space values: points to inches

En Space

An **en space** in the publishing software equals the space taken up by a numeral in the same typeface and size. This is approximately one-half the width of an em space. You produce an en space by pressing and releasing the ESC key and then typing the letter **s**.

En spaces are often used to line up columns of figures, since they take up the same space as numerals.

Thin Space

A **thin space** occupies the same width as the thinnest punctuation marks in each typeface and size (periods, commas, colons, and so on). In the publishing software, you produce a thin space by pressing and releasing the ESC key and then typing the letter **a**.

Thin spaces are often used in proper names with initials to assure that the name is treated as one word and therefore not broken between first and last names. They may also be used in place of soft spaces when tighter word spacing is desired.

Column Width

Although column width is primarily a page makeup issue, you should keep it in mind when choosing a typeface and type size. On single-column pages, lines composed of 60 to 90 characters are generally considered more readable than lines containing more or fewer characters. If you must use a certain column width, consider choosing a type size that brings the number of characters per line within the acceptable range. Some designers suggest that you make your columns narrower if you use sans serif type for text.

For multicolumn documents, you will usually want at least 30 characters per line, although this will vary according to the number of columns and their width.

Typewriter Practices vs. Electronic Composition

Certain common typing practices are not appropriate in copy produced using the Interleaf publishing software.

Double Spacing after Punctuation

Some typists are accustomed to typing two spaces after punctuation marks like periods and colons. Double spaces are not used in this way in proportionally spaced type such as Interleaf's *Modern* and *Classic*. *One* soft space is automatically replaced by a soft return when it occurs at the end of a line. As an extra precaution, when two spaces occur at the end of a line, the publishing software converts one of them into a soft return and holds the other outside the column margin where it will not disrupt the appearance of the column.

However, the publishing software does not remove double spaces from *inside* a line. A *single* (soft) space always should be used after the period at the end of a sentence, and after a colon, or semicolon.

Indents

Soft spaces do not produce a consistent indentation on different lines. When you want to indent a paragraph using this publishing software, you should do so with an em space, not with a series of soft spaces. If you want to automatically indent the first line of a component one em space or any other value, you can use the *First Indent* property on the Component property sheet.

New Components vs. Paragraphs

The Interleaf publishing software uses **components** to describe different line widths, typefaces, and type sizes. If you are accustomed to using returns to create new paragraphs, to make lists, to move from the body of a letter to the closing, and so on, you should consider creation of a new component at these places. You will find that you use the RETURN key less in the publishing software, at least within copy, than you have with typewriters or word processors.

Quotation Marks

On many typewriters and daisywheel printers, there is only one double quotation mark ("). This mark is used to both open and close a quotation. In more advanced typography, instead of the single general-purpose character there are separate *opening* (") and *closing* (") quotation characters. This is also true for single quotations (' and ').

In the *Modern* and *Classic* fonts, no generic quotation marks exist. The *close* quotation mark is produced when you press the double quotation mark key (") while holding down the **Shift** key. You produce an *open* quotation mark by pressing and releasing the ESC key before pressing the double quotation mark key ("). The generic marks are available only in the *Typewriter* family.

Printer and Screen Resolution

Computer displays and most printers show type not in continuous form (as you do when you write or use a typewriter or daisywheel printer) but in dots. The number of dots per linear inch that a particular piece of hardware can show is called its **resolution**.

Generally, printed copy looks better than the screen copy does, because, with current technology, printer resolution is higher than screen resolution. For example, the screens on some displays have a resolution of 75 dots per inch (dpi), while a laser printer may have a resolution of 300 dpi. Typesetter output has an even higher resolution.

If you look very closely at the screen of your workstation, you will be able to see the dots. You would have to look much more closely at a printed piece of output to see the dots because there are so many more dots per inch, and the dots are much smaller.

Figure 8-9 shows the difference between the same 24-point character as printed and as displayed on the screen.
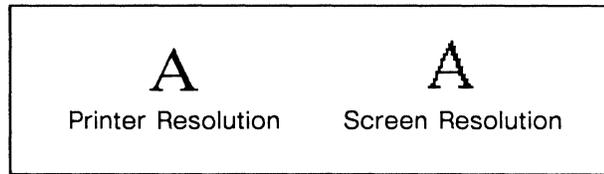
Figure 8-9. Difference between printer and screen resolution

In addition to a difference in resolution, there is a slight size difference between the characters (and anything else) on the display screen and those printed by the laser printer. The screen images appear smaller, and certain characters appear smaller than other characters that are nominally the same size.

For Further Reading

The following books provide additional information on typography, graphic design, and the printing process:

1. Craig, James. *Designing with Type.* rev. ed. Edited by Susan Meyer. New York: Watson-Guptill Publications, 1980.
2. Craig, James. *Phototypesetting: A Design Manual.* New York: Watson-Guptill Publications, 1978.
3. International Paper Co. *Pocket Pal: A Graphic Arts Production Handbook.* New York: International Paper Company, 1984. Pocket Pal Books, P.O. Box 100, Church St. Station, NY, NY, 10008-0100
4. Seybold, John W. *The World of Digital Typography.* Media, Pa.: Seybold Publications, Inc., 1984.

Typefaces

This section lists standard Interleaf typefaces and character sets. Keyboard layouts for the typefaces are shown in the chapter *The Keyboard.* Refer to *The Interleaf Typeface Catalogue* for a list of optional typefaces and character sets.

Classic

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
!@#$%^&=*(1234567890)_-+-[]{·}\|'';:""?/,<>¢-—.
Classic Roman 6pt

Classic Roman, *Italic,* and **Bold** 8pt

Classic Roman, *Italic,* and **Bold** 10pt

Classic Roman, *Italic,* and **Bold** 12pt

Classic Roman, *Italic,* and **Bold** 14pt

Classic Roman, *Italic,* and **Bold** 18pt

Classic Roman, *Italic,* and **Bold** 24pt

Modern

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
!@#$%^&=*(1234567890)_-+~[]{·}\|''"::""?/,<>¢-—.
Modern Roman 6pt

Modern Roman, *Italic,* and **Bold** 8pt

Modern Roman, *Italic,* and **Bold** 10pt

Modern Roman, *Italic,* and **Bold** 12pt

Modern Roman, *Italic,* and **Bold** 14pt

Modern Roman, *Italic,* and **Bold** 18pt

Modern Roman, *Italic,* and **Bold** 24pt

Typewriter

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
!@#$%^&=*(1234567890)_-+~[]{·}\|`´;:""?/,<>¢-—.
Typewriter 8pt

Typewriter 10pt

Typewriter 12pt

Greek 10 and 12pt

The size shown for the Greek and Symbols typeface families is 10 point.

αβξδεφγηικλμνοπθρστυσωχψζABΞΔΕΦΓΗΙΚΛΜΝΟΠΘΡΣΤΥΣΩΧΨZ
!@#$%^&=*(1234567890)_-+~[]{}\|''::""?/,<>¢-—.

Symbols 8, 10, 12, and 14pt

%§©®™♠☞'£ƒ†‡◀▶✱✲☆★□▦□○⦂○◆▲♥♣☞✔℞¹²³⁴⁵⁶⁷⁸⁹⁰✑☞☜♂♀▼□♡
◇◀▶•°●▪■★★₁₂₃₄₅₆₇₈₉₀✳%¶©®™▲⊞"¥/†‡°●①②③④⑤⑥⑦⑧⑨⑩

Mathematics A 10pt

∞◅⊁∅∠CNR⊓⊔⊢⊣↔≪◇≫⊀≐⊰⊱#∇ƒ—⟩∇√•⊖≥≈≤∘≿≡∫Γ{}⟨|
‖⟦ℵ≃≠⟍±∓⊕⊗⊘⊙←⟧⊿⟦⊢↦∽℘ℭ℘ℬ℘ℳℛℜ⊤/~∗≤×⊤÷

Mathematics B 10pt

⊣⊢⊥⊏⊐⊑⊒∈∍∉⋃⋂⊎∧∨⋁⊔⊓⊑⇐⇒⇑⇕⇔∴∀∃

Mathematics Extension

Chapter 9

Page Design

The Interleaf publishing software does not force you to make page makeup decisions in a separate pagination mode. You can make these decisions all at once when you first create a document or you can make any number of changes as you enter text and illustrations.

The page makeup features of the Interleaf publishing software provide you with means for determining the overall structure of a document. Several tools allow you to determine the design of your pages.

- The **Document Page** property sheets make it possible for you to establish such characteristics as the page margins for a document, the number of columns, and the layout of headers and footers.

- On the **Component Page** property sheet, you determine how such matters as page breaks and column straddles will be handled.

- With the **Create Frame** commands, you decide where to put illustrations in relation to text, and, with the **Frame** property sheet, you can determine and change the properties of new and existing frames.

- In **Header/Footer frames**, you enter the text or graphics you want to appear in the top and bottom margins of each page of a document, or on selected pages determined by the layout you established on the Page property sheet.

This chapter describes the property sheets you will use to determine the layout for single- and multicolumn documents.

Creating frames and defining their properties is discussed in the chapter, *Text-Anchored Frames.* Creating headers and footers is covered in the chapter, *Header and Footer Frames.*

Property Sheets and Page Design

The property sheets you will use for page design activities are the Component Page property sheet, the two Document Page property sheets for documents as a whole, and two Page property sheets for *microdocuments* (documents within frames).

The Document Page property sheets and the Component Page property sheet are discussed in this chapter. (See the chapter, *Text in Frames,* for information about the Microdocument Page property sheets.)

Whenever a Page property sheet is mentioned in this chapter, it refers to a full document, not a *microdocument*, unless otherwise specified.

On the Document Page property sheets, you establish properties that determine the overall appearance of your document. These properties can be divided into four groups.

- those that determine page orientation and dimensions
- those that determine the number of columns and the size of the space between columns in a document
- those that determine whether vertical spaces are altered to achieve equal column depth (vertical justification)
- those that determine the layout and appearance of headers and footers

To open a Document Page property sheet:

↙ Point the mouse cursor into the Page box in the document header.

↙ Execute the **Properties** command.

Remember that if you execute the **Properties** *command on the Page pulldown menu while a microdocument is open, the Page property sheet for the microdocument will appear.*

The Normal Sheet of the Page Property Sheet

There are two Document Page property sheets: the Normal sheet and the Custom sheet. You determine the page properties of all documents on the **Normal** sheet and use the **Custom** sheet to refine page layout features, especially those of multicolumn documents.

To select the Page property sheet you want, you turn on the appropriate box in the property sheet header.

Figure 9-1 shows the Normal sheet with the settings for a default document.

Page Properties	**Normal**	Custom	
Orientation	**Portrait**	Landscape	
Columns	1		
Width	8.50	inches	
Height	11	inches	
Margins			
Top	1	inches	
Bottom	1.10	inches	
Left	1	Inches	
Right	1	inches	
Starting Page #	Inherit	1	
Page # Prefix			
Page # Style	**Arabic**	Lowercase Roman	Uppercase Roman
Headers/Footers			
Page Layout	**Single Sided**	Right 1st Page	Left 1st Page
Bleed	**No** Yes		
Diff 1st Header	**No** Yes		
Diff 1st Footer	**No** Yes		

Figure 9-1. Normal sheet of the Page property sheet

Page Orientation

The default setting for the *Orientation* variable is **Portrait** (Figure 9-2a), which is the orientation for pages of text like this one. When you use **Portrait**, the printer orients the *height* of the page along the largest dimension of the paper. The default height is 11 inches.

Landscape orientation (Figure 9-2b) instructs the printer to orient the *width* of the page along the largest dimension of the paper. It is the appropriate setting for tables and diagrams that are wider than 8.5 inches, the default setting for width.

The Origin of the Widget

by Robert J Persimmon,
Widget International

We at **Widget International** are often asked for some background on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, but anyone who has looked the word up in a dictionary knows that it doesn't appear in any of the standard dictionaries.

The president of **Widget International**, Paul Reynolds, has offered one explanation of the origin of the word that should satisfy almost everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget*. People would ask, for example, "Would you please hand me that widget, so that I can fix this widget?"

a. Portrait

House Plan (first floor)
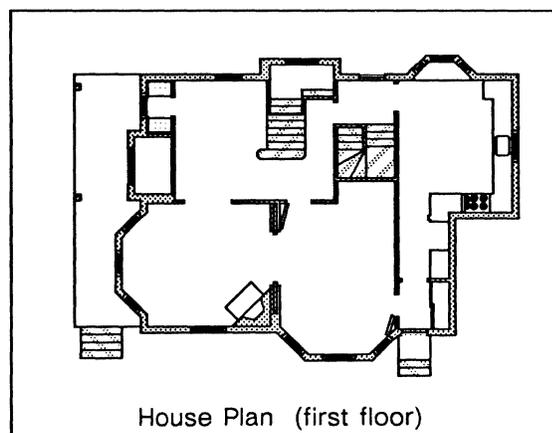
b. Landscape

Figure 9-2. Orientation of pages

The *Orientation* settings are related to the *Height* and *Width* settings on the Normal sheet. When you switch from **Portrait** to **Landscape** or **Landscape** to **Portrait**, the *Height* and *Width* settings on the property sheet are automatically reversed, with one exception. If you change from **Portrait** to **Landscape** when the page height and width are already appropriate for **Landscape**, the dimensions are not interchanged.

Horizontal and vertical page margins do not change when you change from **Portrait** to **Landscape** or from **Landscape** to **Portrait**.

Page Width and Height

The *Width* and *Height* settings control the outside dimensions of the pages in a document. They do not change the orientation in which the document is sent to the printer.

If you do not want to print a document, you can give it a height and width of up to 35 inches. If you do want a printed copy, there are some size limitations determined by the type of printer you use. They are discussed in the chapter, *Printing*, in the *Reference Manual*, Volume 2.

To avoid losing copy when printing on any printer, you should make the page margins at least .25 inches all round, and also make sure that you do not use negative component margins that extend to within .25 inches of the page edge.

Page Margins

The default page margins (1.0 inch for the top, left and right margins of the page and 1.10 inches for the bottom margin) are set so that the white space around text enhances the appearance and readability of your document. You can, however, change these margins to suit your own needs by adjusting the value(s) on the Page property sheet.

Multicolumn Pagination

The multicolumn pagination capabilities of the Interleaf publishing software are made possible by a group of related page layout features. These features allow you to select the number of columns per page for a document and have text and graphics flow into the columns in a predictable way to produce an aesthetically pleasing product.

The key features of multicolumn pagination in this software are:

- You can specify the number of **columns per page** in your document and specify the **gutter** (white space) between columns.

- The software **balances columns** on a page so that they are as near to equal in depth as is possible given the content of the page.

- When you turn on **vertical justification** for a document, the software adjusts certain vertical spacing on the page to attempt to make the columns exactly the same depth for every page in a document. Vertical justification fine-tunes column balancing.

- You can set component properties so that some components **straddle** the width of the text area while others are divided into columns.

Creating a Multicolumn Document

Number of Columns

The *Columns* variable on the Normal sheet allows you to lay out your document with any number of columns as long as the number is within the limitations described in the section, *Soft Component Straddles,* in this chapter.

The greater the number of columns you choose the more difficult it is to maintain an appealing document appearance. For one thing, it is simpler to balance the content of two or three columns than it is to balance the content of nine columns. Also, relatively even word spacing becomes more difficult to maintain the greater the number of columns. Figure 9-3 shows one page of a flush right/left, single-column document (a) and the same page after the number of columns is changed to two (b).

**The Origin of the Wid-
get**

by Robert J Persimmon,
Widget International

We at Widget International are often asked for some background on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, but anyone who has looked the word up in a dictionary knows that it doesn't appear in any of the standard dictionaries.

The president of **Widget International**, Paul Reynolds, has offered one explanation of the origin of the word that should satisfy almost everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget*. People would ask, for example, "Would you please hand me that widget, so that I can fix this widget?"

**The Origin
of the Wid-
get**

by Robert J Persimmon,
Widget International

We at Widget International are often asked for some background on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, but anyone who has looked the word up in a dictionary knows that it doesn't appear in any of the standard dictionaries.

The president of Widget International, Paul Reynolds, has offered one explanation of the origin of the word that should satisfy almost everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget*. People would ask, for example, "Would you please hand me that widget, so that I can fix this widget?"

(a)   (b)

Figure 9-3. Word spacing in single and multicolumn pages

Specifying a multicolumn format when you first create a document has the advantage of seeing your ultimate page design as you enter text. However, you might find the multicolumn format distracting during text entry. In this case, you might want to enter the text and graphics for the document in a single column and change the number of columns in that document later — either during or after editing. You can change the number of columns in a document any number of times. The document will be repaginated immediately.

The Custom Sheet of the Page Property Sheet

Though many of the most important variables on the **Custom** sheet (Figure 9-4) can be applied to single-column documents, most of the settings are especially useful in fine-tuning the appearance of multicolumn documents.

The default settings on the Custom sheet are designed to insure optimal page makeup and appearance in the majority of documents. However, should you have special page design goals, you can adjust these settings according to your preferences.

To see the Custom sheet of the Page property sheets:

    ↙ Open the Document Page property sheet.

    ↙ If the Custom sheet is *not* showing, point the mouse cursor into the **Custom** box in the property sheet header and click the **left** button.

```
┌─────────────────────────────────────────────────────────────┐
│ Page Properties │ Normal │ ███Custom███                      │
├─────────────────────────────────────────────────────────────┤
│                                                          ▲    │
│ Hyphenation              ██On██ │Off│                         │
│  # Consecutive           ██Any██ ①②③④                          │
│                                                               │
│ Revision Bar Placement   ██Left██ │ Automatic │               │
│                                                               │
│ Gutter Width             ██Default██ │  2  │ picas            │
│                                                               │
│ Balance Columns          ██On██ │Off│                         │
│                                                               │
│ Vertical Justification   │On│ ██Off██                         │
│  Margin Stretch          │ 200 │ %                            │
│  Margin Shrink           │ 50 │ %                             │
│  Frame Margin Stretch    │ 10 │ %                             │
│  Feathering, Maximum     ██On██ │Off│ │ 8 │ %                 │
│  Vert. Just. Pages       ██On██ │Off│                         │
│    Depth, At Page Break  │ 95 │ %                             │
│    Depth, No Page Break  │ 90 │ %                        ▼    │
│                          ◄░░░░░░░░░░░░░░░░░░░░░░░░░░░►         │
└─────────────────────────────────────────────────────────────┘
```
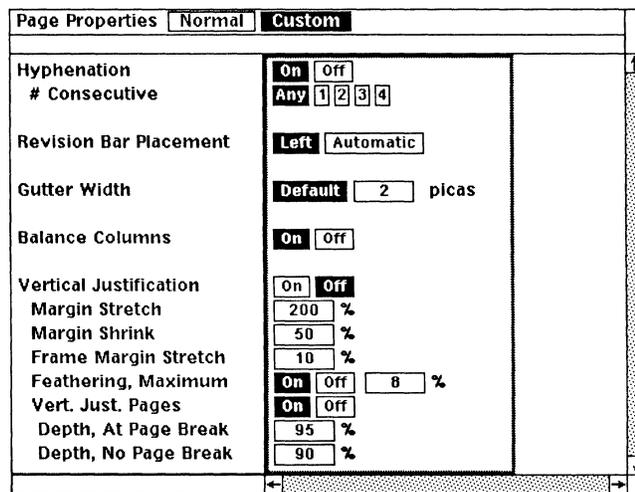
Figure 9-4. Custom sheet of the Page property sheets

Gutter Width

The **gutter** between columns is similar to the break between pages. Text cannot flow horizontally across the gutter, because the gutter acts as a boundary. If you want to keep the multicolumn format for the rest of the document and you also want to create a table with defined horizontal and vertical relationships, use a microdocument with tabs in a straddle component for the table.

The software produces default **gutter widths** when you make the number of columns in a document greater than one. Gutter width is measured in picas (one pica is approximately one-sixth of an inch). For two columns, the default gutter width is two picas. For more than two columns, the default gutter width is one pica.

Adjusting Gutter Width

If you prefer more or less space between columns, you can change the *Gutter Width* setting. You can specify any positive width for the gutter, but you will probably want to observe certain guidelines. If you make a gutter too small, columns can be so close to each other that they may be indistinguishable. If you make a gutter too large, you may create a page that is distracting rather than attractive and readable.

To change the width of the gutter between columns:

    ✔ Position the property sheet caret in the box on the *Gutter Width* line.

    ▣ Type the width you want, confirm it by pressing TAB or RETURN, and **Apply** the change.

Column Balancing

When column balancing is on, the software attempts to match column depth as closely as possible without changing line spacing to meet this goal.

On otherwise empty pages, column balancing does not occur until at least 10 percent of the total column depth can be filled in each column. For example, in a document with two or more columns, text and graphics do not move into the second column until 20 percent of the depth of the first column has been filled. In a document with three or more columns, text and graphics do not move into the third column until 15 percent of the depth of the first and second columns has been filled. Figure 9-5 shows what happens when the setting *Balance Columns* is On and you reach the minimum depth needed to balance two columns.

(a) insufficient depth for balancing   (b) two balanced columns
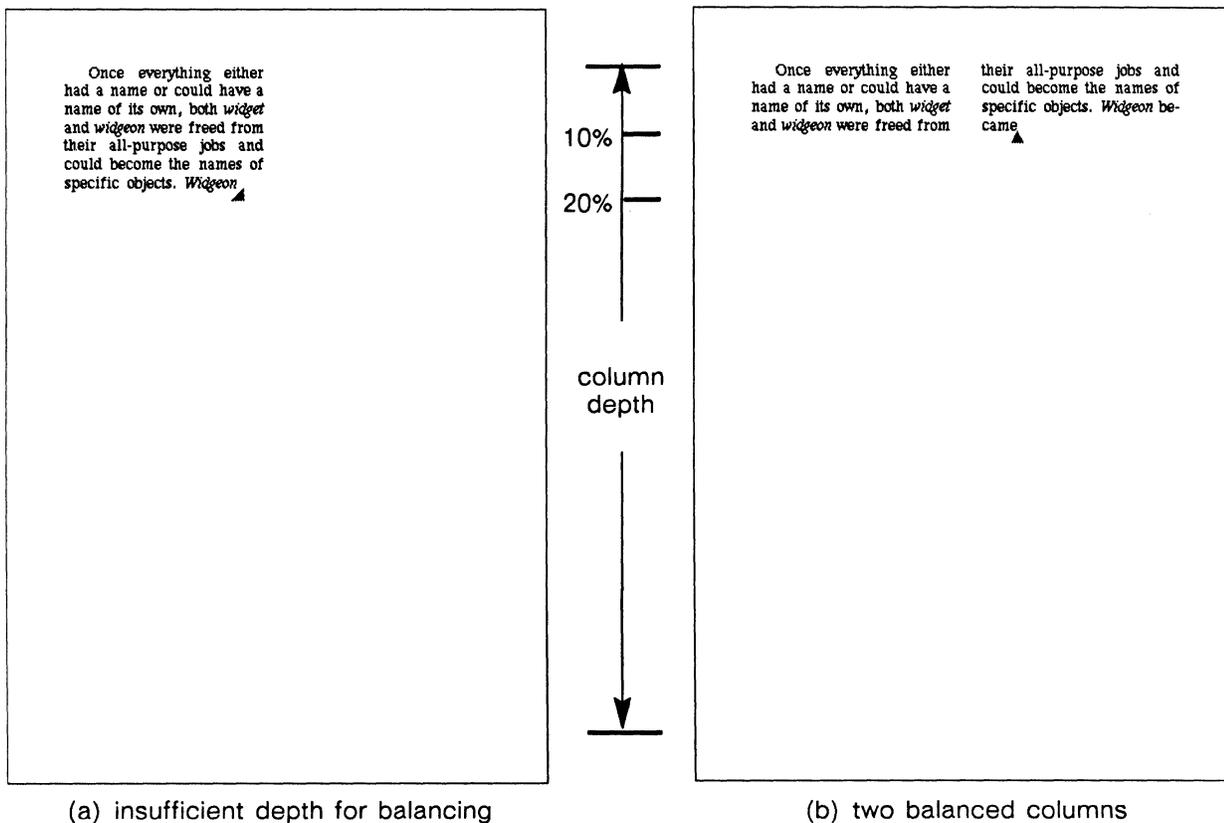
Figure 9-5. Column balancing

Vertical Justification

Vertical Justification and its related functions are important to the final appearance of single-column as well as multicolumn documents. These functions attempt to make **column depth** identical on all pages. On pages with a single column, such as this one, this means the software attempts to make the depth of text and

graphics on each page equal to the *page height minus the top and bottom margins and headers and footers. Vertical Justification* is **On** by default.

Vertical Justification in Multicolumn Documents

The goals and mechanisms of vertical justification are the same in multicolumn documents as they are in single-column documents. When vertical justification is on, the software attempts to ensure that *column depth* on all pages is identical, and that *text depth* on all pages is identical throughout a document. To achieve these goals, it first stretches or shrinks component margins and stretches frame margins — which makes zero margins undesirable. As a last resort, interline spacing, *feathering,* is increased slightly. With more than one column on a page, these functions take on added significance. Stretch, shrink and feathering are each proportional for all vertical spaces they affect in a document.

To turn Vertical Justification on or off for a document:

  ✓ Open the Page property sheet for the document.

  ✓ In the boxes on the *Vertical Justification* line on the Custom sheet, select **On** or **Off** and **Apply** the change.

◈Vertical Justification Variables

The variables related to vertical justification are only effective when vertical justification is on. The first three variables listed under *Vertical Justification* affect the margins between components and frames. The fourth variable affects the spaces between lines of text.

All stretch or shrink on a page is proportional — that is, the fixed value of each margin is adjusted by the same *percentage.* Therefore, if the space between two components was twice that between two other components on the same page, it will remain twice as great no matter what the stretch or shrink percentage is. (Zero margins cannot be altered.)

The first variable is *Margin Stretch,* with the default set at **200%**. This means that the software will stretch the margins between components up to 200% to achieve vertically justified pages. *Margin Shrink* is set at **50%**, which means that the software will shrink the spaces between components as much as 50% to achieve vertically justified pages.

The last variable related to margin adjustment is *Frame Margin Stretch,* and is slightly different than *Margin Stretch.* The default of **10%** means that *10% of the height of the frame* is the amount that can be used to stretch the space above and/or below the frame to accomplish vertical justification. This implied margin is distributed above and/or below the frame according to frame placement. You can adjust this amount from a minimum of 0 (the equivalent of **Off**) to a maximum of 100%.

Increasing the space between lines in a component is called **feathering**. The *Feathering, Maximum* variable allows you to determine the greatest percentage of feathering you want the software to use in order to achieve vertical justification of pages. The default (**On**) is **8%**, the maximum allowable feathering. You can change this value to any number as low as 0%. Feathering is the last resort of the software to accomplish vertical justification. If you select **Off**, the software will attempt to achieve vertical justification, but by means other than feathering.

Setting Vertical Justification Pages

When *Vert. Just. Pages* is **On**, and *Depth, At Page Break* is set at its default, the software will try to justify a page if it is at least 95% filled and the first component on the next page is set for *Begin New Page*. Before the software will attempt vertical justification on a page where *Depth, No Page Break* is set at its default and the component that falls on the next page is *not* set for *Begin New Page*, the previous page must be at least 90% filled. Therefore, if the depth properties are set at the default values and a page is only 70% full, for example, the software will not try to vertically justify them.

Figure 9-6 shows the way the software can *stretch* or *shrink* a page to achieve vertical justification. The page in Figure 9-6a, which has vertical justification off, does not have enough text to reach the bottom page margin. The same page with vertical justification on fills the entire text area.

The page in Figure 9-6c does not have enough text to fill the page as is but also does not have enough space at the bottom to accommodate the two lines of the paragraph on the following page. Since the widow and orphan setting for the paragraph is 2, the page must end short.

In Figure 9-6d, with vertical justification on, the vertical space required for the same text and graphics is reduced by an entire line, so additional text can now fit on the page.

Used for Description

It became clear fairly early that more than one word was needed to describe the objects people used in their daily lives. This is where *widget* and *widgeon* are similar: *widgeon* now describes wild ducks, but the true origins of the word are shrouded in mystery. Probably, *widgeon* was used in the beginning as the name for all birds. The word was obviously derived from *widget* and is our only case of direct derivation.

Once they had *widgeon* as a word people who had been saying, "Look at that widget landing on that widget!" could make a useful distinction, so they began saying, "Look at that widgeon landing on that widget!"

It was only a matter of time before someone began to call the *widget* the *widgeon* was landing on, a *roof* or a *chimney*.

The Widget International logo.

Widget Digest   Page 4

(a) Vertical Justification Off

Used for Description

It became clear fairly early that more than one word was needed to describe the objects people used in their daily lives. This is where *widget* and *widgeon* are similar: *widgeon* now describes wild ducks, but the true origins of the word are shrouded in mystery. Probably, *widgeon* was used in the beginning as the name for all birds. The word was obviously derived from *widget* and is our only case of direct derivation.

Once they had *widgeon* as a word people who had been saying, "Look at that widget landing on that widget!" could make a useful distinction, so they began saying, "Look at that widgeon landing on that widget!"

It was only a matter of time before someone began to call the *widget* the *widgeon* was landing on, a *roof* or a *chimney*.

The Widget International logo.

Widget Digest   Page 3

(b) Vertical Justification On

Stretch

The Origin of the Widget

by Robert J. Persimmon,
Director of Corporate Communications,
Widget International

We at Widget International are often asked for some background on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, but anyone who has looked the word up in a dictionary knows that it doesn't appear in any of the standard dictionaries. Even in *The Oxford English Dictionary* there is no entry between *widgeon* and *widow*.

Prehistory   Medieval Period   1982

Widgeon → Widget → Widget Int'l.

The genesis of the the word widget

We have done some research of our own and come up with a few answers to the pub-

Widget Digest   Page 1

(c) Vertical Justification Off

The Origin of the Widget

by Robert J. Persimmon,
Director of Corporate Communications,
Widget International

We at Widget International are often asked for some background on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, but anyone who has looked the word up in a dictionary knows that it doesn't appear in any of the standard dictionaries. Even in *The Oxford English Dictionary* there is no entry between *widgeon* and *widow*.

Prehistory   Medieval Period   1982

Widgeon → Widget → Widget Int'l.

The genesis of the the word widget

We have done some research of our own and come up with a few answers to the public's questions. We would like to share these with you.

Widget Digest   Page 1

(d) Vertical Justification On

Shrink

Figure 9-6. Page stretch and shrink for vertical justification

Component Page Properties

The settings on the Component Page property sheet allow you to make important decisions about page design.

Figure 9-7 shows the Component Page property sheet with the default settings in reverse video.

Figure 9-7. Page sheet of Component property sheet

Begin New Page

The default setting for *Begin New Page* is **No**. If **No** is in reverse video, the component will start wherever it will fit on a page. If you turn on **Yes**, the component will always begin at the top of a page. You can apply this property to an individual component to make particular pages look better when you are making final page break decisions. You can also apply the property globally so that each of the components of a particular type — major headings, for example — will appear at the top of a new page.

Orphan and Widow Control

An **orphan** is a single line at the bottom of a column, and a **widow** is a single line at the top of a column. These lines usually contain text, but for page makeup purposes it is important to realize that the software will also count apparently empty lines (for example, a line on which only tabs or spaces are present) when attempting to conform to the orphan/widow settings.

With the Interleaf publishing software, you can control the number of lines of a component that must appear at the bottom of one page and at the top of the next page before the component can be broken across page boundaries.

Orphan Control and *Widow Control* makes aesthetically pleasing page makeup possible. The system default for both orphans and widows is two lines. With these default settings, a component of fewer than four lines cannot be split across page boundaries, and in a single-column document, one line will not appear all by itself at the bottom or the top of a page. If both lines of a two-line component or all three lines of a three-line component will not fit at the bottom of one page, the entire component will move to the top of the next page.

A one-line component will be put wherever it will fit unless you give special in-structions. See the discussion of *Allow Break After* for the way to control the placement of one-line components.

Orphan Control and *Widow Control* can be set to any number of lines between *1* and *16*. The greater the number of lines, the less likely the component is to break across page boundaries.

In multicolumn documents, *Orphan Control* and *Widow Control* also determine the number of lines of a component that must appear at the top and bottom of a *column*. However, it will be overridden between columns (but never between *pages*) when necessary to achieve column balancing.

To change the setting for Orphan or Widow Control:

↑ Point the mouse cursor into the box with a number in it, and click the **left** button.

▣ Type a number between **1** and **16**. Press TAB to confirm the number of lines and to move the cursor to the other box in which you may type a value.

or ▧ Point the mouse cursor into the **Default** box, and click the **left** button.

A 2 is entered automatically in the Lines box.

You can apply the changes you make either to a single component or to all com-ponents of a particular name.

Allow Break Within

If *Allow Break Within* is set to **Yes**, the software is allowed to put part of the com-ponent on one page and part of it on the next page as long as this does not vio-late the widow/orphan settings for the component.

If *Allow Break Within* is set to **No**, the component will not, as a rule, break across page boundaries. The exception is a component that is longer than a single page.

If it is necessary to break such a component, the *Allow Break Within* setting and in some cases the *Widow Control* for the component will be superseded. In other words, the software will fit as many lines of the component as possible on a single page even if this means that the number of lines at the top of the next page will be fewer than the number designated by the *Widow Control* setting.

In multicolumn documents, the *Allow Break Within* setting also determines whether a component is allowed to break across columns on the same page.

Allow Break After

The default for *Allow Break After* is **Yes** because it is normally reasonable to let the software decide where page breaks should occur.

If **Yes** is turned on for *component a,* a page break will be permitted between *component a* and the component that follows it.

However, *Allow Break After* may be set to **No** in some circumstances to make sure that components that should appear together on the same page do appear there (at least in part).

For example, if the heading for this section, *Allow Break After,* had appeared all alone at the bottom of a page, the relationship between the heading and the section would have been obscured, if not completely destroyed. In order to avoid this kind of break between a heading and the section it heads, the *Allow Break After* property for the *heading* components in this document was set to **No.**

In multicolumn documents, the *Allow Break After* setting for *component a* also determines whether a column break will be permitted between *component a* and the component immediately following it.

Components in Multicolumn Documents

Soft Component Margins

Top and bottom margins of components are **soft**; that is, the top margin is ignored when it occurs at the top of a page and the bottom margin is ignored when it occurs at the bottom of a page. Soft top and bottom component margins help ensure consistent top and bottom alignment on the page.

Top and bottom component margins are also soft (for columns) where columns and straddles meet.

Component Straddles

A **straddle** is a component that extends horizontally across the entire text area of a multicolumn document. Straddles are likely to be used within multicolumn documents for large illustrations, titles, and subheadings. Note that the need for greater width is not the only reason to use a straddle. If you want a component to be alone on a line all the way across a page, you should use a straddle even if the component would fit in a single column.

The default setting for *Straddle Columns* is **No.** In a multicolumn document, if **No** is in reverse video, the component will assume the column width. If **Yes** is in reverse video, the component will span the entire width of the **text area** (page width minus margins). The setting of this property has no effect in a single-column document.

Figure 9-8 shows a multicolumn document with several components straddling columns. The numerals indicate the order of the text on the page.

Straddle ──────▶

The Origin of the Widget ①

by Robert Persimmon,
Widget International

We at **Widget International** are often asked for some background on the origin of the word ② ③et. Everyone here what we mean when we say *widget*, but anyone who has looked the word up in a dictionary knows that it doesn't appear in any of the standard dictionaries. Even in *The Oxford English Dictionary*, the

most complete dictionary of them all, there is no entry between *widgeon* and *widow*.

We have done some research of our own and come up ③ few answers to the public's questions and we would like to share these with you.

We have found that many people associate *widgets* with *widgeons*, an interesting kind of

wild duck. This is an ingenious and appealing, though inaccurate, explanation that may be accounted for by their latching on to the closest word they can find in their ④ry that is in the bit like *widget*. Actually, *widgeon* and *widget* may have gone through similar transformations over the course of history, as we shall demonstrate here.

Straddle ──────▶

The president of **Widget International,** Paul Reynolds, has offered one explanation of the origin of the word that should satisfy almost everyone: *widget* is a word that has existed si ⑤ e beginning human language before there were individual names for things, everything was a *widget*. People would ask, for example, "Would you please hand me that widget, so that I can fix this widget?"

It became clear fairly early that more than one word was needed to describe the objects people used in their daily lives. This is where *widget* and *widgeon* are similar. *widgeon* no ⑥ bes wild ducks, but true origins of the word are shrouded in mystery. Probably, *widgeon* was used in the beginning as the name for all birds. The word was obviously derived from *widget* and is the only case of direct

derivation that we still have.

Once they had *widgeon* as a word people who had been saying, "Look at that widget landing on that widget!" could ma ⑦ eful distinction, they started saying, "Look at that widgeon landing on that widget!"

It was only a matter of time before someone began to call the *widget* the *widgeon* was landing on, a *roof* or a *chimney*.

Figure 9-8. Straddles in a multicolumn document

Margins between Straddles and Columns

Although they do not necessarily cause a new page to begin, straddles have two characteristics similar to a page break.

- When straddles occur immediately above or below multiple columns, the adjacent top or bottom margins of the *columns* are *soft*. The space between the straddle and the columns is determined strictly by the Top Margin and/or Bottom Margin setting on the Component property sheet of the *straddle*. If vertical justification is on and necessary for the page these components are on, only the straddle margin is altered.

- Like a hard page break, straddles are impermeable. That is, text flows from column to column above the straddle, and text also flows from column to column below the straddle, but text from the bottom of a column above the straddle does not flow to columns below the straddle.

Soft Component Straddles

When the combination of page margins, component margins and first indent, the number of columns, and gutter width does not yield a column at least one-half inch (3 picas) wide, components without *Straddle* turned on will straddle. This condition does not change the straddle setting on the Component property sheet. If you make changes that allow a column of at least one-half inch, the component will no longer straddle.

Soft Frame Straddles

Regardless of the *Straddle* setting of the component it is anchored in, a frame straddles when its width exceeds the column width. This does not change its properties (frames have no straddle property) but only its behavior under these conditions. If you increase the column width to at least frame width or decrease the frame width to column width or less, the frame no longer straddles. (For a full discussion of frames, see the chapter, *Text-Anchored Frames.*)

Using the Component Page Properties

Some page makeup decisions are worth making the moment they come up because they will not change as you revise the document. For example, you will never want a *heading* separated from the section that follows it, and you will almost always want a *chapter* to begin a new page.

On the other hand, when you first enter text in a document, you may find that an exceedingly awkward looking, one-line component falls all alone at the bottom of a page. You may want to rush right in and move it to the next page. If you can, restrain this impulse! As you edit the document it is likely that your one-line component will move. If it does not, you can always move it later, when the text is more stable and the decisions you make are more likely to stick.

Component Bar Behavior in Multicolumn Documents

In a multicolumn document, the component bar always shows shows the names of straddle components. It also shows the names of other components *that begin in the column the text caret is in*. If you want to see the names of the components that begin in a different column, position the text caret in that column.
Figure 9-9 shows how caret position affects the display of components in a two-column document.

Text

	The Origin of the Widget ⋏		The Origin of the Widget ⋏
title		**title**	
byline	*by Robert J. Persimmon, Widget International*		*by Robert J. Persimmon, Widget International*
openpara	We at Widget International are often asked for some background on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, yet anyone looking the word up in a dictionary knows that it doesn't appear in any of the standard dic-		We at Widget International are often asked for some background on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, yet anyone looking the word up in a diction-
	tionaries. Even in *The Oxford English Dictionary*, the most complete dictionary of them all, there is no entry between *widgeon* and *widow*.	**paragraph**	ary knows that it doesn't appear in any of the standard dic-
	We have done some research of our own and come up with a few answers to the public's questions, and we would like to share these with you.		tionaries. Even in *The Oxford English Dictionary*, the most complete dictionary of them all, there is no entry between *widgeon* and *widow*.
rule		**rule**	We have done some research of our own and come up with a few answers to the public's questions, and we would like to share these with you.
paragraph	We have found that many people associate *widgets* with *widgeons*, an interesting kind of wild duck. This is an ingenious and also appealing, though inaccurate, explanation that may be accounted for by their latching on to the	closest word they can find in their dictionary that is in the least bit like the word *widget*. Actually, *widgeon* and *widget* may have gone through similar transformations over the course of history, as we shall demonstrate here.		We have found that many people associate *widgets* with *widgeons*, an interesting kind of wild duck. This is an ingenious and also appealing, though inaccurate, explanation that may be accounted for by their latching on to the	closest word they can find in their dictionary that is in the least bit like the word *widget*. Actually, *widgeon* and *widget* may have gone through similar transformations over the course of history, as we shall demonstrate here.

(a) Text caret in left column   (b) Text caret in right column

Figure 9-9. Component name display in multicolumn documents

When the text caret is in a straddle component, the names of the components in the leftmost column are displayed in the component bar. If you *drag* the caret through a straddle, the names of components that begin in the leftmost column are displayed while the mouse caret is passing through the straddle.

Be especially careful to position your caret in the proper column before you select a component or perform other editing activities within it. Before you execute the **Join** command to combine two adjacent components, you should use the **Find Next** command to see what you are joining.

If you *extend select* from one page to the next or previous page, when you reach the halfway point between the last visible component on one page and the first visible component on the other, you select *all* components from the text caret to the page boundary, including those you cannot see (i.e., columns to the right if you select to the next page and columns on the left if you select to the previous page). This also applies to the **Thru Page, To Front,** and **To End** commands.

If you select a range of text extending across two or more columns, the component names shown are still only those in the column in which the text caret is positioned.

Troubleshooting Pagination Problems

Along with the tremendous increase in capabilities provided by multicolumn pagination comes an increase in complexity. Choices that you never encounter at all in single-column documents must be made on a regular basis if your multicolumn documents are to look well-designed.

If you encounter unacceptable pagination, you should check for certain common causes, including:

- unbreakable text (setting *Allow Break Within* and/or *Allow Break After* to **No** on the Component property sheet)
- empty components
- inappropriate use of the *Same Column as Anchor* setting on the Frame property sheet
- inappropriate use of the *Begin New Page* setting on the Component property sheet
- changing from single- to multicolumn without adjusting first indent and/or left and right component margins.
- use of zero top and/or bottom component margins

What follows is a discussion of some unacceptable pagination situations you might find and the likely causes.

Irregular Column Length

Non-straddling empty components in a multicolumn document can make column balancing appear to be working incorrectly. Even a component with no contents takes up a certain amount of space and has top and bottom margins. If an empty component is at the top or bottom of a column, it is used to balance columns even though it does not contain anything to make a *visual* balance.

An unbreakable component or chain of components can cause pagination difficulties in multicolumn documents. Remember that the settings *Allow Break Within* or *Allow Break After* apply to both *pages* and *columns*. If you make an unreasonably long block of text unbreakable, you force the software to break that block in violation of at least one of the properties you have chosen.

Ambiguous Text Order

An empty *straddle* component causes the columns before it to wrap to the top of the page and balance above it. When an empty component is thin enough and is surrounded by non-straddling text, it can make the order of text extremely confusing (Figure 9-10)
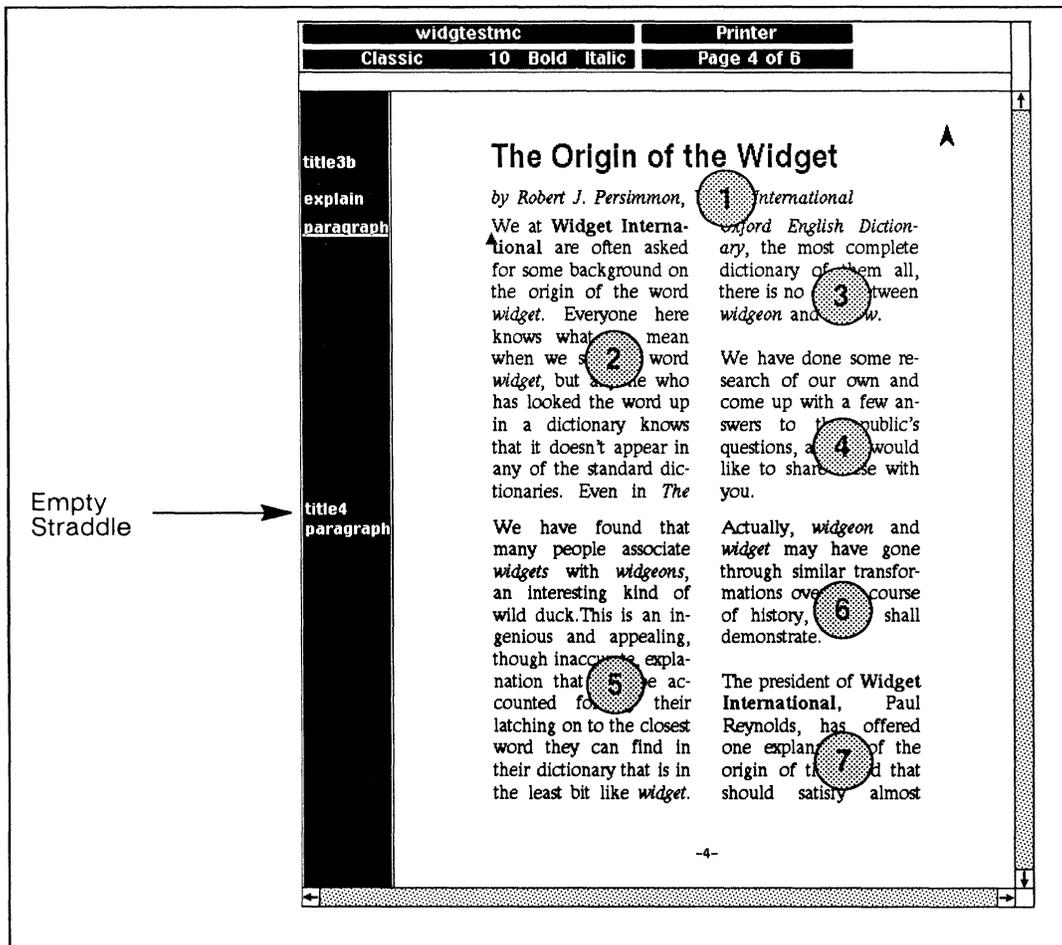
Figure 9-10. Text flow with empty Straddle component

Inappropriate Margins or Indents

When you change from a single column to a multicolumn layout, component properties that made perfect sense before may become unreasonable. For example, a 1.5 inch first indent that works perfectly well in a single-column document with 6-inch columns, becomes awkward in a two-column document with columns that are less than three inches wide, and even more awkward if the same document were put into a three-column layout.

In some cases you may find that there is no space between a straddle component and a multicolumn component. This can occur when the spacing between the straddle and the multicolumn component is determined strictly by the margins of the straddle. To avoid this problem, avoid zero margins on any component. Instead, put at least some margin both above and below a component.

Excessive Raggedness or Word Spacing

When you choose a multicolumn layout for a document, you obviously reduce the width of the columns compared to the same document in a single-column format. The narrower your columns become, the more important it is to have frequent hyphenation of words. If you have hyphenation off or at a low setting on either the Document property sheet or the Component property sheet, you will likely get extremely ragged line endings in ragged right components and too much space between words in justified components

Poor Column Alignment

Vertical justification is especially important in multicolumn documents. In some cases, turning *Vertical Justification* **Off** can make adjacent columns *almost* align at the bottom of the page. These columns would look better either if they lined up exactly or did not come within a few lines of each other so that it is clear that you did not intend to align them — in other words, you don't want it to look like a mistake. In these cases, you should turn *Vertical Justification* back **On** so that your columns align.

Too Much Space between Figure and Caption

If you place a figure caption outside of the frame the figure is in, vertical justification may increase the space between figure and caption to a greater extent than you want. There are two ways to remedy this problem. You can put the caption inside the figure frame, which will guarantee that the distance between the two is unchanged by vertical justification. Another solution is to reduce drastically the top margin of caption and bottom margin of the component containing the frame for the figure. This second solution still allows some stretch and shrink, but is effective in many cases.

Short or Blank Pages

If you set *Same Column as Anchor* to **Yes** on a Frame property sheet when the frame is too large to fit on the the same page, the software will break the text at the line the anchor is in and create a new page. This will generate two short pages. (See the chapter, *Text-Anchored Frames* for a full discussion of frames and their anchors.)

If you globally set *Begin New Page* to **Yes** on a Component property sheet, you may inadvertently create short pages throughout a document. Also, if you have two consecutive *Begin New Page* components and the first one is empty, a blank page will be generated.

Chapter 10

Text-Anchored Frames

There are two major types of frames in the Interleaf publishing software:

- **Text-anchored frames** are the frames you use for illustrations and text inside the normal text area of the page.
- **Header/footer frames** appear on each page of a document to provide information such as chapter name, page number, and document revision dates.

For a detailed discussion of header/footer frames, see the chapter *Header and Footer Frames.*

Text-anchored frames can be created in a variety of sizes, placements, and alignments. For example, a frame can be very small and a part of a line of text like any character, or it can be so large that it is the only object on the page.

In some instances, you will use the Create Frame submenu on the Text Location popup menu to create a frame. In other instances, you may create a new frame by changing the name and properties of an existing frame. Or, the software can create a frame for you. In any case, you can alter the placement of any frame, its alignment, or its size to fit changing circumstances. You can even determine whether or not you want a given frame to share its contents with other frames of the same name.

This chapter includes the following topics:

- frame anchors
- creating frames
- frame master definitions
- frame properties
- frame placement
- frame behavior in multicolumn documents

Frame Anchors

A frame **anchor** ▣ is the reference point in text for a frame, and it changes position if you add or delete text before it.

At anchor frames are placed at their anchors, so you never see their anchors. These frames behave like a character and always move with the surrounding text

and remain in the same relationship to it — that is, text never flows around them.

Frames with **following anchor**, **following text**, and **bottom** placement are always placed after their anchors. Frames with **top** placement can come before or after their anchors. Whether they are placed before or after their anchors depends on where there is room for them and where their anchors are.

You can select the frame anchor (except for at anchor frames) the same way you select any character — i.e., by highlighting it using the left and/or right buttons. To select the frame by selecting its anchor, you must select only the anchor and not any adjacent spaces or characters.

Creating Frames

There are two ways to create a frame: you can use the Create Frame submenu on the Text Location popup menu to choose the frame you want, or you can let the software create a frame for you (by pasting a graphic object directly into your document as text or as a component).

Using the Frame Submenu to Create a Frame

Six default frames are provided by the software and are listed on the original Create Frame submenu (Figure 10-1). Each of these frames has a unique name that corresponds to its placement.
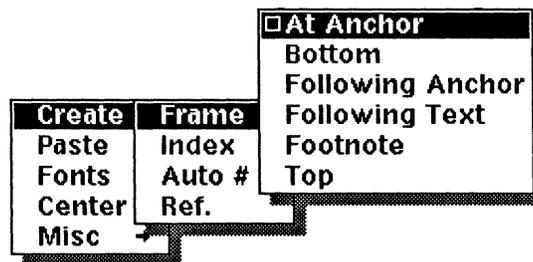
Figure 10-1. Default Create Frame submenu

Throughout this chapter, when we use the terms, "at anchor," "bottom," "following anchor," and so on, we are referring to the placement property of a frame, not to the default names on the Create Frame submenu.

To create a frame:

↙ Position the editing caret where you want the frame anchor, and hold down the **middle** button to see the Text Location popup menu.

↙ Slide onto the Frame submenu, and release the button when the frame you want to create is in reverse video.

A frame (and anchor, when this is appropriate) of the default size is positioned at the appropriate place on the page. The newly created frame is in the selected state.

Automatic Frame Creation

It is not always necessary or convenient to create a frame yourself. For example, when you have a graphic object, such as a plotter object or a screen image, and do not know its size, you can paste it either directly into text or as a component and let the software create a frame of the correct size. Figure 10-2 shows the icons of desktop objects suitable for automatic framing.
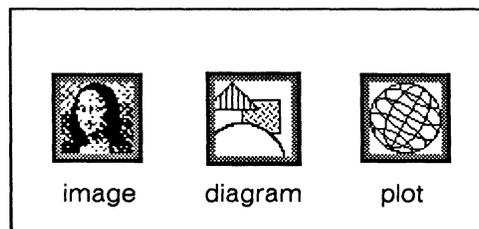
Figure 10-2. Icons for graphic objects

When the object is larger than the largest frame that will fit within the page margins of your document, the software will create a frame the full width of the page (ignoring margins) and will adjust the object proportionally to fit this frame. If you are not satisfied with the placement, size, or alignment of a frame created by the software, you can change these properties on the Frame property sheet once the object has been pasted.

A frame created automatically by the software is named **Auto**. This name is added to the frame submenu.

Pasting a Graphic Object into Text

You can paste a graphic object into the text of a component that already has contents or into the text area of an empty component. The type of frame you get depends on several conditions.

If you paste the graphic object into a component that already has contents, the software creates either a frame with at anchor placement or a frame with following anchor placement and center horizontal alignment. An at anchor frame is cre-

ated if the graphic object is no taller than the height of the default font of the component in which you are pasting the object.

A following anchor frame is created if the graphic object is taller than the component font.

If you paste a graphic object into an *empty* component, the software creates an at anchor frame with its horizontal position determined by the component alignment (flush left, centered, or flush right). If the object requires a frame wider than the component into which you are pasting it, the frame is centered on the page.

Pasting a Graphic Object as a Component

If you paste a graphic object using the **Paste** command on the Component Location popup menu, the software first creates a component of the same type as the component with the caret and places that new component at the location of the caret. Then, it makes the alignment of this component **Center** and creates an at anchor frame into which it puts the graphic object.

Frame Master Definitions

Every frame in the publishing software has a name and gets its original properties from a **master definition** associated with that name. Frames of the same name share a single master definition unless you alter the properties of an individual instance of the frame. Any time you change the name of a frame to a unique name (a name that is not on the Create Frame submenu), you create a new master definition. You will find more often than not, that when you change the name of a frame, you are doing so because you are also changing its properties, and you want the new name to distinguish this frame from others with different properties. This new frame name will appear on your frame menu until you purge its master definition. (See the section *Deleting Master Definitions*, later in this chapter.)

Each master definition exists independently of any occurrences of the frame in a document; you can cut all instances of a frame and its master definition will still be available to you.

If you change the properties of a frame and **Apply** those changes without changing the name of the frame, you will have a single instance of the frame with different properties. The master definition will not be affected.

The major aspects of the frame masters feature are:

- You create and update master definitions by making changes to the Frame property sheet.
- The list of frame masters appears on the Create Frame submenu.
- Even if you delete the last instance of a named frame, the master definition remains until you delete it. Master definitions are saved with a document.
- When you paste a uniquely named frame into a document, its master definition is pasted with it and its name appears on the frame submenu of the document it was pasted into.
- Changing the name of a frame to a unique name, and executing the **Apply** command on the property sheet creates a new master but leaves the old one in place. Executing **Global Apply** creates a new master and deletes the old one.

Creating New Master Definitions

The frame names on the default Create Frame submenu represent frame master definitions from which you can create new master definitions. There is no limit to the number of frame master definitions you can have. Frame names can be up to twenty characters long.

For example, if you create the default frame named *Following Anchor*, and on its property sheet change its name to *largeframe*, and apply the change, *largeframe*

will appear on the Create Frame submenu (Figure 10-3). Note that *largeframe* appears on the menu in addition to *Following Anchor*. Whenever you execute the **largeframe** command on the popup, the frame you get will have all of the properties of the original *largeframe* you created.
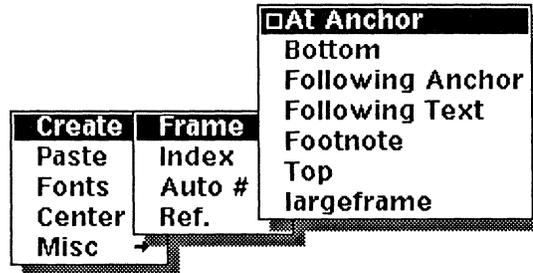
Figure 10-3. Create Frame submenu with an added frame name

To create a new master definition:

✓ **Create** a frame and open its property sheet.

✓ Change the name of the frame and any properties you care to and **Apply** the change.

The status line in the Frame property sheet will display a message that a master definition for framename *has been created.*

Anytime you want to see the master definition for a frame, create a new instance of the frame and open its property sheet. (This is safer than looking at the property sheet of an existing frame of the same name because you might have changed the properties of that instance.)

Changing Master Definitions

When you want to change a frame master definition, first create a new instance of the frame whose master definition you want to change.

To change a master definition:

✓ **Create** an instance of the frame whose master definition you want to change and open the property sheet.

✓ Make the changes you wish and execute **Global Apply**.

The status line of the property sheet will display a message stating that the master definition for framename *has been updated. All existing instances of this frame and all new instances that you create will reflect the properties of the new master.*

If you want to change the properties of only a single instance of a frame without affecting the master definition, simply make your changes on the property sheet and apply them. Do not change the frame's name.

Deleting Master Definitions

There are two ways to delete master definitions. One method allows you to delete the master definitions for *all* frames of which there are no instances in the document. The other method allows you to delete individual masters whether or not there are any instances of the frame(s) in the document. With the latter method, the results you get depend on whether the name change involves a new or existing frame name.

The following instructions direct you to use the **Purge Masters** command. Before executing this command, be sure your document contains at least one instance of every named frame you want to keep.

To delete the master definition for all frames without instances:

🖋 On the Text Location popup menu execute the `Misc →` `Masters →` `Purge Frames` command.

A stickup will appear asking you to confirm that you want to delete the masters for all frames without instances.

🖋 Execute **Confirm** on the stickup that appears.

If you execute the **Purge Masters** command and there are any instances of frames in the document, those master definitions will remain on the submenu.

If you use the **Purge Masters** command when there are *no* frames in your document, your Create Frame submenu will list the default at anchor frame. This single master is provided so that you will be able to create new frames should the need arise.

When you rename a frame in order to delete a master definition, the results you get will vary depending on whether you choose a unique name or a frame name that already exists in your document.

To delete individual master definitions:

✔ **Create** an instance of the frame whose master definition you wish to delete,

or ✔ Select an instance of the frame whose master definition you wish to delete.

✔ Open the property sheet for the frame and change the name to a unique frame name or to a name that already exists in your document.

If you assign a name that does not already exist in your document, you will delete the master definition by creating a new master definition.

If you assign a name that already exists in your document, all frames of the deleted name will now have the new name and the properties that go with that name.

✔ **Globally Apply** the name change.

A stickup menu will appear asking you to confirm the deletion of the master definition for the original frame.

✔ Execute **Confirm** on the stickup menu that appears and **Close** the Frame property sheet.

If you check the list of frames on the Create Frame submenu, you will find that the frame whose name change you globally applied no longer appears there.

	If document contains *no** instances of the master	If document contains instances of the master
Change to an *existing* master name	• deletes the old master • properties of the new master are globally applied to the instances	• deletes the old master • properties of the new master are globally applied to the instances
Change to a *unique* master name	• deletes the old master • creates a new master with the unique name	• deletes the old master • creates a new master with the unique name • new master name globally applied to the instances; properties of instances change
*except the instance you created to make the global name change.		

Figure 10-4. Effects of globally applying frame name changes

Pasting Frames Between Documents

If you paste a frame into a document and the document into which you are pasting contains no master for that frame name, the software will create a master definition from the master of the frame you are pasting.

If you are pasting a frame into a document and the document into which you are pasting contains a master definition for that frame name, the master definition remains unchanged. (For information on pasting shared contents frames from one document into another, see the section *Pasting Shared Contents Frames Between Documents*, later in this chapter.)

Master Definitions of Pre-Release 3.0 Frames

Frames saved in previous versions of the software and opened in Release 3.0 are assigned the software default frame names based on their placement property. For example, frames with at anchor placement will be named At Anchor in Release 3.0; frames with following text placement would be named Following Text. (See the section *Frame Placement,* later in this chapter.) The instance of the frame(s) will remain unaffected. The master definitions in the document will be the 3.0 default masters. You may change the masters according to your needs and preferences, or just delete them.

The Frame Property Sheet

Figure 10-5a shows the Frame property sheet for a default at anchor frame, and Figure 10-5b shows the Frame property sheet for the other default frames. (The property sheet shown in Figure 10-5b is for a default Top frame, but the property sheets for all text-anchored frames except at anchor frames have this basic property sheet, except for the placement property.)

```
┌─────────────────────────────────────────────────────────────┐
│ Frame properties:                                           │
├─────────────────────────────────────────────────────────────┤
│                                                             │
│  Name              ┌──────At Anchor──────┐                  │
│                                                             │
│  Width             ┌0.41┐ inches              ↑             │
│                                                             │
│  Height            ┌0.14┐ inches                            │
│                                                             │
│  Shared Contents   [Yes][No]                                │
│                                                             │
│  Placement         [At Anchor]                              │
│                                                             │
│                    [Top][Bottom]                            │
│                                                             │
│                    [Following Anchor][Following Text]       │
│                                                             │
│  Vertical Alignment  ┌0.03┐ inches                          │
│                                                             │
└─────────────────────────────────────────────────────────────┘
```

(a)

```
┌─────────────────────────────────────────────────────────────┐
│ Frame properties:                                           │
├─────────────────────────────────────────────────────────────┤
│                                                             │
│  Name              ┌──────Top──────┐                        │
│                                                             │
│  Width             ┌6.50┐ inches              ↑             │
│                                                             │
│  Height            ┌3.25┐ inches                            │
│                                                             │
│  Shared Contents   [Yes][No]                                │
│                                                             │
│  Placement         [At Anchor]                              │
│                                                             │
│                    [Top][Bottom]                            │
│                                                             │
│                    [Following Anchor][Following Text]       │
│                                                             │
│  Horizontal Alignment  [Left][Center][Right][Other]         │
│                                                             │
│  Same Column as Anchor  [Yes][No]                           │
│                                                             │
└─────────────────────────────────────────────────────────────┘
```

(b)

Figure 10-5. Frame property sheets with default settings

Frame Size

The height of a frame can range from zero to a maximum of the text area height (page height minus top and bottom margins). You cannot extend a frame (or anything else in a component) into the top or bottom margins because the header/footer frames already occupy those margins. If you want a frame to extend out to the top and bottom limits of the printable page, you can change the top and bottom margins to zero. However, because of the limit imposed by the printing devices used with this software, nothing will print within approximately

one-quarter inch of the page edge. Therefore, you could set the top and bottom margins to .25 inches as long as you did not put anything in headers or footers.

Frame width can range from zero to the total width of the page, including its left and right margins. Common widths used for frames include the column width, the component width, and the text area width.

When a change you make to frame width would cause the frame to cross the column-width boundary in either direction or would make the frame width greater than the total page width, a stickup menu appears (Figure 10-6) *unless the value you entered is exactly equal to the page or text area width, the column width, or the component width.* This menu allows you to confirm the value you just entered or to choose one of three other widths: page width (which in this case refers to the width of the text area), column width, or component width. In some cases two or even three of these values will be the same. You can use this stickup to create frames with one of these widths by intentionally entering a width larger than the page width.
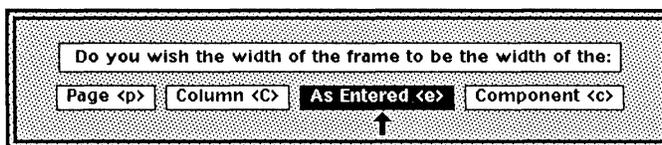
Figure 10-6. Frame Width stickup menu

Shared Contents

The *Shared Contents* property allows you to share the contents of frames with the same name within a single document. This property can apply to all instances of a particular frame or to only some instances of a master definition.

The following summarizes the major aspects of the *Shared Contents* feature.

- *Shared Contents* enables you to share both text and graphics among frames of the same name.
- You can turn on *Shared Contents* before or after you create or paste contents in your frame.
- Whenever you change the contents of a shared contents frame, this change will be reflected in all other frames of the same name and in the master, *if* the *Shared Contents* setting is turned on. The change is not visible in related frames until you close the frame in which you made the change.
- No change will occur in an instance of a related frame if *Shared Contents* is turned off for that instance.

Turning on Shared Contents for a Master Definition

If you set the *Shared Contents* property to **Yes** (the default is **No**) on a Frame property sheet and execute **Global Apply,** you will update the frame's master definition so that all frames of that name within a single document will share their contents.

For example, if you have a document that repeatedly requires the use of a thick rule ━━━ , you can create a master definition for a frame named *rule* with the *Shared Contents* variable set to **Yes**. Then you can create a line of the desired thickness in the open frame. When you create another instance of a *rule* frame, the frame will be displayed with a rule identical to the one in the first frame. Changes to the contents of any one of the frames will be reflected in all others of the same name (except where *Shared Contents* is turned off for individual instances of a frame of that name). If you update a shared content frame and its master definition has *Shared Contents* set to **Yes**, then the master will be updated when an instance of the frame has its contents changed.

To turn on Shared Contents for all instances of a frame, including the master:

✔ Open the property sheet of a frame whose contents you want to be shared with other frames of the same name.

✔ Turn on the **Yes** box next to *Shared Contents* and **Global Apply.**
You will see the stickup represented in Figure 10-7.

> **Warning: The contents of all 'rule' frames — including the master — will be replaced by the contents of this frame. Please Confirm.**
>
> Confirm Cancel
> ↑
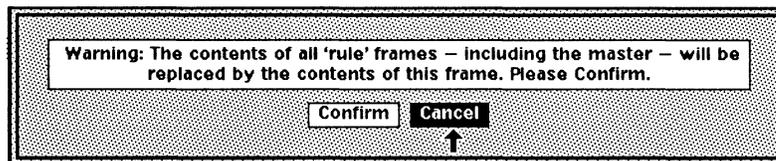
Figure 10-7. Update Master and Instances stickup

✔ Execute **Confirm** and **Close** the property sheet.
The master definition now has Shared Contents turned on.

If your document contained any additional instances of frames of that name, those frames and the master now have the same contents as this frame.

✔ **Create** another frame of the same name as the first.
A new frame appears with contents identical to those of the master.

✔ You can make a change to the contents of one of the instances and see that all existing instances as well as any new instance(s) you create reflect this change.

There may be occasions when you want to have *Shared Contents* turned *on* for a master definition, but turned *off* for all instances of the frame. For example, if you want to create a standard form to be filled in with information that varies from case to case, you first set *Shared Contents* to **Yes** for the master definition. Then you create the standard contents of your form in an instance of the frame. When you create other instances of the form, they will all share the standard contents. If you open the property sheet for each instance and turn off Shared Contents, you can enter the unique information in each instance of the standard form.

If, at any point after you have turned off *Shared Contents* in an individual instance of a frame whose master definition has *Shared Contents* turned on, you decide to turn *Shared Contents* back on, you will see a stickup warning you that the contents of that frame will be set to those of the master definition (Figure 10-8).

Warning: The contents of this frame will be replaced by the
contents of the master. Please Confirm.

Confirm Cancel
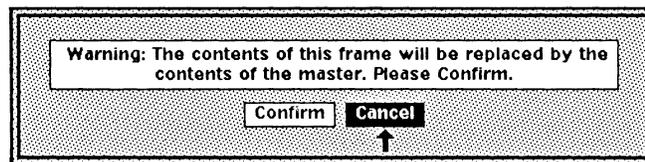
Figure 10-8. Replace by Master stickup

Shared Contents and Individual Instances of a Frame

The default for *Shared Contents* is **No**. If you open the property sheet of a frame, change the *Shared Contents* property to **Yes**, and execute **Apply**, only the contents of frames of the same name with *Shared Contents* turned on will be affected; their contents will be replaced when you execute confirm on the stickup that appears (Figure 10-9).

Warning: The contents of all other 'At Anchor' frames with shared—contents settings
of Yes will be replaced by the contents of this frame. The shared—contents
setting of the master remains No. Please Confirm.

Confirm Cancel
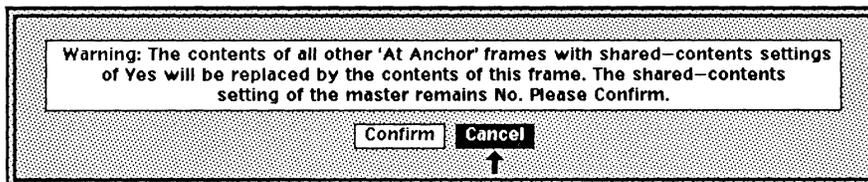
Figure 10-9. Shared Contents stickup

Note that it is the frame whose property sheet you have opened that retains its contents and these contents are the ones that are propagated to related frames with *Shared Contents* turned on. Before you execute **Confirm** on the stickup that appears, be sure you understand which frames' contents you are going to lose.

You might find that you want to share the contents of a frame with individual instances of that frame. For example, you might have three frames of the same name with *Shared Contents* turned off, and you want the contents of the first frame shared with the second, but not with the third frame. You would need to

perform a *name change* on the third frame, then turn on *Shared Contents* for the first frame, and **Global Apply** to update the master definition. The second frame, which has the same name as the first, would now have *Shared Contents* turned on also, and would share the contents of the master.

Another way to achieve the same results, without performing a name change, is to turn on *Shared Contents* for the second frame, then turn on *Shared Contents* for the first frame. The contents of the second frame are updated to those of the first, since the second frame had *Shared Contents* turned on. The third frame is unaffected because it has *Shared Contents* turned off.

Applying the *Shared Contents* property to a single instance as in the case above has numerous implications. Remember that if *Shared Contents* is turned off for the master definition of a frame, when you create *new* instances of the frame, their contents will not be shared. By the same token, if *Shared Contents* is turned on for the master, but you turn it off for an instance of the frame, that instance will not be affected by changes to frames of the same name with Shared Contents turned on.

When you are creating frames, decide their purpose(s) in advance, so that you can assign names and group them according to whether or not you want their contents shared. When in doubt, perform a name change to create a new master definition.

Pasting Shared Content Frames Between Documents

When you paste any frame, with or without *Shared Contents* turned on, into a document with no master definition of that name, the software will use the master of the frame you pasted to create a master in the document into which you are pasting. However, the results are different when you paste a shared content frame into a document that already contains a master definition for that name.

If the document into which you are pasting a shared content frame contains a master definition for that name with *Shared Contents* turned off, the frame you are pasting will appear as an instance (with *Shared Contents* turned on) of that master definition.

If the master in the document into which you are pasting has *Shared Contents* turned on, a warning stickup will appear stating that the contents of the frame you are pasting will be overwritten by the contents of the master in the document into which you are pasting.

> **Warning:** The intricacies of the shared contents feature can be subtle, so it is best to practice with the feature before using it in an important document. Whenever you use the shared contents feature in an important document, be sure to pay close attention to the stickups that appear.

Autonumbering in Microdocuments in Shared Content Frames

Autonumbers in microdocuments in shared content frames are identical in all instances of the frame with *Shared Contents* turned on. If the autonumber stream begins outside the frame, the numbers in all related microdocuments will be determined by the numbers in the first instance of the frame.

Search and Replace and Spell Checking in Shared Content Frames

When you use the *search and replace* and *spelling checker* features in microdocuments within shared contents frames, the software will locate only the first instance of the search object or incorrect word *in the document*. So, for example, if you are using the spelling checker and you begin the search in the middle of a document, past the first instance, you will have to wrap to the beginning of the document so that the first occurrence of the misspelled word can be flagged.

Frame Placement

All text-anchored frames have one of five placements:

- at anchor
- following anchor
- following text
- top
- bottom

The default *Footnote* frame is a special kind of *Bottom* frame and will be discussed with *Bottom* frames.

At Anchor Frames

When you create an at anchor frame, space is opened up for the frame at the position of the editing caret.

When an at anchor frame is created in the middle of text, such as in Figure 10-10, the frame moves along *with* the text that surrounds it.

Example 1

(b). Define a 2x2 real matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ in such a way that $[x,y] \rightarrow [x,y]A$ is the rotation described above.

Example 2

(b). Refer to the previous section, and define a real matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ in such a way that $[x,y] \rightarrow [x,y]A$ is the rotation described.

Example 3

(b). Define a real matrix $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ in such a way that $[x,y] \rightarrow [x,y]A$ is the rotation described above.

Example 4

(b). Define a 2x2 real matrix $A =$ in such a way that $[x,y] \rightarrow [x,y]A$ is the rotation described above.

Figure 10-10. Behavior of at anchor frames

In Figure 10-10, the height and width of the frame have been changed to accommodate the brackets. Example 1 shows the original lines of text. The line spacing has been adjusted by the software to accommodate the frame.

Examples 2 and 3 in Figure 10-10 show how an at anchor frame behaves like a character. If you add text before it, the frame moves forward to make room for the inserted text (Example 2). Similarly, if you delete text before an at anchor frame, the frame moves back to fill up the gap (Example 3).

If you delete the frame itself (Example 4 in Figure 10-10), the software automatically readjusts the line spacing. Deleting only the frame leaves a double space between = and *in*.

You should always put a space between an at anchor frame and the text on either side of it, so that the frame will not be considered part of a word for line breaks and hyphenation. However, if the frame is followed by a punctuation mark, you should not add a space between the frame and the punctuation mark.

There are times when you will want to put an illustration in a large at anchor frame in a component by itself because you want it to stay in a particular relationship to the text and you do not want text to flow around it.

Page Placement Frames

Default page placement frames share several properties in single-column documents.

- The dimensions on all default page placement frames *except* Footnote frames are the same: *Width 6.50, Height 3.25.*

- Footnote frames have a *Width* of *6.50 and Height of .20, by default.*

- *Horizontal Alignment* is **Center** for all the frames, except Footnote frames. Footnote frames have a horizontal alignment of **Left**.

The properties of frames created in multicolumn documents are discussed later in this chapter.

Following Anchor Placement

Ideally, when you create a frame with following anchor placement, a space is opened up right below the line with the anchor in it. If the anchor is too close to the bottom of the page, the frame is put at the top of the next page.

Figure 10-11a shows how two consecutive pages look on the screen when there is room for the frame right after its anchor. Figure 10-11b shows how two consecutive pages look when there is not enough room for the frame on the same page with its anchor.

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck *(Figure 1)*▣. This is an interest-

ing, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

page 6

It became clear fairly early that more than one word was needed to describe the objects people used in their daily lives. When and how *widget* came to be the name of a particular object is the subject for another day.

Frame immediately following anchor

(a)

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck. This is an interesting, though inaccurate, explanation. Paul Reynolds, the president of **Widget International,** has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

It became clear fairly early that more than one word was needed to describe the objects people used in their daily lives. *(Figure 1)* ▣ shows the confusion. When and how *widget* came to be the name of a particular object is the subject for another day. For the moment, it

page 6

is enough to know that our ancestors liked the word so well that they passed it down to us unchanged from its humble but important beginnings.

Frame on page following anchor

(b)

Figure 10-11. Following anchor frames

Following Text Placement

The best way to understand following text placement is to imagine a one-page document with nothing in it except a following text frame and its anchor. The anchor is the first thing in the document, and the frame itself is just below it. If text is added above the frame, the frame is pushed down the page until no more text is added or until there is no more room for text. When there is as much text as will fit on the page, the software leaves the frame where it is, which may leave some extra vertical space below the frame. Therefore, a full page of text with a following text frame as the last item might not have a justified bottom margin.

A following text frame is the preferred frame type when you want a diagram on the last page of a document *immediately* after all the text and when you want to put a diagram at the bottom of a page that has footnotes on it. In the second case, creating a following text frame assures you that the diagram will be above

the footnotes, no matter which anchor is placed first on the page. In all other instances, when you want a frame at the bottom of a page, create a bottom frame.

Figure 10-12a shows an example of a frame with following text frame on the last page of a document, and Figure 10-12b shows an example of a following text frame on a page with a footnote. In Figure 10-12, the extra space is between the following text frame and the footnote frame. Because the footnote placement frame is a bottom frame, page 6 is vertically justified.

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck. This is an interesting, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

As people became more sophisticated, they found it clumsy and rude to have to point at the things they needed. What, they sometimes wondered, was language good for if it couldn't be more specific.

page 6

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck *(Figure 1).* This is an interesting, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

page 6

It became clear fairly early that more than one word was needed to describe the objects people used in their daily lives. *Figure 1* shows the confusion. When and how *widget* came to be the name of a particular object is the subject for another day.

Frame on last page of document

(a)

It became clear fairly early that more than one word was needed to describe the objects people used in their daily lives. When and how *widget* came to be the name of a particular object is the subject for another day.

Frame on page with footnote

(b)

Figure 10-12. Following text frames

Top Placement

As the name implies, a frame with top placement will appear at the top of a page. If there is room for it on the page with its anchor, it will be at the top of that page (Figure 10-13a). If there is no room for it on that page, it will be at the top of the next page (Figure 10-13b).

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck *(Figure 1)*.This is an interesting, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

page 6

It became clear fairly early that more than one word was needed to describe the objects people used in their daily lives. When and how *widget* came to be the name of a particular object is the subject for another day.

Frame on page with its anchor
(a)

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck . This is an interesting, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone:*widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

It became clear fairly early that more than one word was needed to describe the objects people used in their daily lives. *Figure 1* shows the confusion. When and how *widget* came to be the name of a particular object is the subject for another day. For the moment, it

page 6

is enough to know that our ancestors liked the word so well that they passed it down to us unchanged from its humble but important beginnings.

Frame on page after its anchor
(b)

Figure 10-13. Top frames

Bottom Placement

A bottom frame is always placed after its anchor. It is placed at the bottom of the same page as its anchor if there is room for it. If it will not fit on the same page, it is placed at the bottom of the following page if there is text on that page or at the top of that page if there is no text. In Figure 10-14a the bottom frame is on the same page with its anchor; in Figure 10-14b it is on the next page.

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck *(Figure 1)* ▣. This is an interesting, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

It became clear fairly early that more than one word was needed to describe

page 6

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck . This is an interesting, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

It became clear fairly early that more than one word was needed to describe the objects people used in their daily lives. *Figure 1* ▣ shows the confusion. When and how *widget* came to be the name of a particular object is the subject for another day. For the moment, it is

page 6

---

the objects people used in their daily lives. When and how *widget* came to be the name of a particular object is the subject for another day.

page 7

---

enough to know that our ancestors liked the word so well that they passed it down to us unchanged from its humble but important beginnings.

page 7

With anchor and frame on the same page
(a)

With anchor and frame on different pages
(b)

Figure 10-14. Bottom frames

When you create a footnote frame, the software creates a bottom frame with the standard default width but with a default height .20 inch — much smaller than the height of a regular bottom frame. You can change the height of a footnote frame to meet your own specifications. The frame properties are set so that the frame will always appear on the same page as its anchor.

Frame Placement Summary

If a frame cannot be placed on the same page as its anchor, it will always be on a page *following* its anchor.

Frames of the same placement are always placed in the order in which their anchors occur. If different type frames are anchored on a page and there is room on the page for all of them, dissimilar frames will be arranged on the page according to their placement type (see Figure 10-12b for an example). If there is not enough room on the page, the frame whose anchor appears last on the page will move to the following page.

In one circumstance the rule of precedence may present an aesthetic problem. When you have a frame anchor—such as the anchor for a following text frame — so close to the bottom of a page that there is not enough room for the frame on the page with its anchor, the frame will be placed on the next page.

Then, if you put a footnote anchor after the following text anchor but on the same page, the line with the footnote anchor in it and all the rest of the lines after it will move to the next page because a footnote and its reference must be on the same page. Usually this will leave a gap at the bottom of the previous page. To avoid this, put the following text anchor after the footnote anchor.

Frame Alignment

At anchor frames have a vertical alignment property. All other text-anchored frames have a horizontal alignment property.

Vertical Alignment of At Anchor Frames

The default *Vertical Alignment* setting for an at anchor frame aligns the bottom of the frame to the deepest descender in the default component font and maintains the line spacing of the component, as shown in Figure 10-15a. If you enter a larger value, the frame will drop below the line of text and change the line spacing between the line with the frame in it and the following line (Figure 10-15b). If you enter a smaller value (for example, a negative value), the frame will rise above the line and change the line spacing between the line with the frame in it and the previous line (Figure 10-15c).

Notice that in all three cases, the vertical alignment is the distance from the *baseline* of the line of text to the *bottom* of the frame. It is positive when the baseline of the text in the component is above the bottom of the frame and negative when it is below.

You can vary the vertical alignment of at anchor frames in relation to the line they are on.

(a)
Vertical Alignment
set at the
default of .03

(b)
Vertical Alignment
is set at .5

(c)
Vertical Alignment
is set at −.5
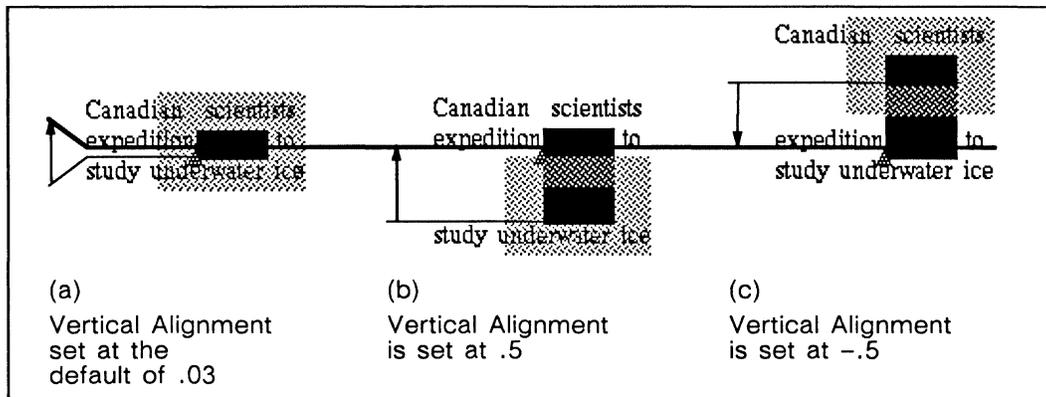
Figure 10-15. Vertical alignment changes to at anchor frame

Horizontal Alignment

There is no horizontal setting for at anchor frames. For all other default text-anchored frames, with the exception of frames with footnote placement, the horizontal alignment setting is **Center**. The default horizontal alignment setting for footnote frames is **Left**.

You can vary the *horizontal* alignment of page placement frames on the page. Figure 10-16 shows the different alignments available. Figure 10-16d shows an **Other** setting, which is the only one you have to enter a value for. The **Other** setting permits you to choose the distance (in inches) from the left margin of the page (or the left margin of the *column* in multicolumn documents) to the left edge of the frame.

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck *(Figure 1).* ▣This is an interesting, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

page 6

(a) Left

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck *(Figure 1).* ▣This is an interesting, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

page 6

(b) Center

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck *(Figure 1).* ▣This is an interesting, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

page 6

(c) Right

We are often asked for some background on the origin of the word *widget.* We have found that many people associate it with *widgeons,* an interesting kind of wild duck *(Figure 1).* ▣This is an interesting, though inaccurate, explanation. The president of **Widget International,** Paul Reynolds, has offered his own explanation that should satisfy everyone: *widget* is a word that has existed since the beginning of human language. Before there were individual names for things, everything was a *widget.* People would ask, for example, "Would you please hand me that widget, so that I can fix this widget."

page 6

(d) Other (indented 1.35 inches from left-hand margin of component

Figure 10-16. Frame horizontal alignment

To set an Other (horizontal) Alignment:

▤▢▢ Use the **left** button to turn on **Other**.

The numerical field appears when you choose Other. Figure 10-17 shows the Horizontal Alignment *for Figure 10-16b when the setting is switched from* **Center** *to* **Other** *for that page and frame size. The numerical field shows the distance (in inches) between the left edge of the frame and the left margin of the page when you choose switched from* **Center** *to* **Other** *alignment.*

Horizontal Alignment	[Left] [Center] [Right] **Other** [2] ins.
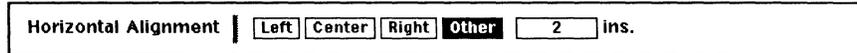
Figure 10-17. Horizontal alignment on Frame property sheet

▢ Type a number in the *ins.* box, confirm, and **Apply**.

Figure 10-18 shows the Horizontal Alignment *for Figure 10-16d.*

Horizontal Alignment	[Left] [Center] [Right] **Other** [1.35] ins.
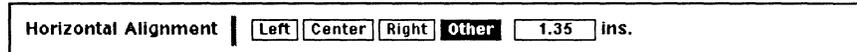
Figure 10-18. Horizontal alignment with new value entered for Other

You often will have a specific value in mind for the horizontal alignment. However, in other cases you may have only a general idea of how you would like a frame aligned. In these cases, you first could select a fixed Horizontal Alignment setting (left, right, or center) that is close to the frame position you want and then fine tune that alignment using the **Other** setting.

Same Column as Anchor

Same Column as Anchor setting applies to page-placement frames only. You have already seen the use of this property to assure the placement of frames with footnote placement with their anchors. In a single-column document, *same column* simply means the *same page*. For instance, if you want to make sure that a figure is always on the same page as its reference, you can set this property to **Yes**. If you set Same Column as Anchor to **No**, the frame will still be placed in the same column if possible, but no special adjustments will be made to accommodate it.

Anchor Visibility

You can choose whether or not you want the anchors in a document to be visible on the screen. It is useful to have them visible at least some of the time because you can use an anchor to select its frame. Also, when a frame and its visible anchor are selected, you can easily see the relationship between the frame and the reference to it in text.

Anchors are place markers, not characters, so they do not print and do not really take up room in text. If you have several visible anchors on the same line, characters and anchors may look on the screen as though they are overlapping or crashing into each other. However, the text will *print* correctly. If the frame anchors are visible on the screen and you want to see how your document will look when it is printed, execute ⌷Misc →⌷ ⌷Show →⌷ ⌷Frame Anchors →⌷ on the Text Location popup, which toggles the visibility of these anchors.

When frame anchors are in their invisible state, it is possible to select a range of text without knowing that an anchor is included. The software protects you from inadvertently cutting invisible frame anchors (and their frames) in two ways.

- You cannot cut an invisible frame anchor using keyboard commands. If you try to do so, the following message appears in the status line at the top of the document: *The region contains frames. Please use the cut operation.*
- If you attempt to cut an invisible frame anchor using the **Cut** command, a stickup menu appears. On the stickup, you can confirm or cancel the **Cut** command.

Frames in Multicolumn Documents

There are some differences between the behavior of frames in single-column and multicolumn documents.

Soft Frame Straddles

Regardless of the *Straddle* setting of the component it is anchored in, a frame straddles when its width exceeds the column width. This does not change its properties (frames have no straddle property) but only its behavior under these conditions. If you increase the column width to at least frame width or decrease the frame width to column width or less, the frame no longer straddles.

How Straddles Affect Frame Placement

Straddles are *impermeable* to frames created in non-straddle columns if those frames are no wider than the column. For example, if you create a footnote frame in a column that is separated from the bottom of the page by a straddle (Figure 10-19a), the frame appears at the bottom of the column and above the

straddle, not at the bottom of the page. If you anchor the frame in the column *below* the lowest straddle (Figure 10-19b), the frame appears at the bottom of the page.

Straddles do not prevent frames anchored in other straddles from reaching the top or bottom of the page. As Figure 10-19c and Figure 10-19d show, you can anchor a footnote frame in either of the straddles on the page and that frame will appear at the bottom of the page. In addition, if a frame anchored in a non-straddle column is wider than that column, a straddle component does not prevent it from reaching the top or bottom of the page.

The Origin of the Widget

by Robert J. Persimmon, Widget International

We at Widget International are often asked for some background on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, yet anyone looking the word up in a dictionary knows that it doesn't appear in any of the standard dic-tionaries. Even in *The Oxford English Dictionary*, the most complete dictionary of them all, there is no entry between *widgeon* and *widow*.

We have done some research of our own and come up with a few answers to the public's questions, and we would like to share these with you.

Widgets and Widgeons

We have found that many people associate *widgets* with *widgeons*, an interesting kind of wild duck. This is an ingenious and also appealing, though inaccurate, explanation that may be accounted for by their latching on to the closest word they can find in their diction-ary that is in the least bit like the word *widget*. Actually, *widgeon* and *widget* may have gone through similar

(a) anchor in column above straddle

The Origin of the Widget

by Robert J. Persimmon, Widget International

We at Widget International are often asked for some background on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, yet anyone looking the word up in a diction-ary knows that it doesn't appear in any of the standard dic-tionaries. Even in *The Oxford English Dic-tionary*, the most complete dictionary of them all, there is no entry between *widgeon* and *widow*

Widgets and Widgeons

We have found that many people associate *widgets* with *widgeons*, an interest-ing kind of wild duck. This is an ingenious and also appealing, though inaccurate, ex-planation that may be accounted for by their latching on to the closest word they can find in their diction-ary that is in the least bit like the word *wid-get*. Actually, *widgeon* and *widget* may have gone through similar transformations over

(b) anchor in column below straddle

The Origin of the Widget

by Robert J. Persimmon, Widget International

We at Widget International are often asked for some back-ground on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, yet anyone looking the word up in a diction-ary knows that it doesn't appear in any of the standard dic-tionaries. Even in *The Oxford English Dic-tionary*, the most complete dictionary of them all, there is no entry between *wid-geon* and *widow*.

We have done some research of our own and come up with a few answers to the public's questions, and we would like to share these with you.

Widgets and Widgeons

We have found that many people as-sociate *widgets* with *widgeons*, an interest-ing kind of wild duck. This is an ingenious and also appealing, though inaccurate, ex-planation that may be accounted for by their latching on to the closest word they can find in their diction-ary that is in the least bit like the word *wid-get*. Actually, *widgeon*

(c) anchor in straddle above straddle

The Origin of the Widget

by Robert J. Persimmon, Widget International

We at Widget Inter-national are often asked for some back-ground on the origin of the word *widget*. Everyone here knows what we mean when we say *widget*, yet anyone looking the word up in a diction-ary knows that it doesn't appear in any of the standard dic-tionaries. Even in *The Oxford English Dic-tionary*, the most complete dictionary of them all, there is no entry between *wid-geon* and *widow*.

We have done some research of our own and come up with a few answers to the public's questions, and we would like to share these with you.

Widgets and Widgeons

We have found that many people as-sociate *widgets* with *widgeons*, an interest-ing kind of wild duck. This is an ingenious and also appealing, though inaccurate, ex-planation that may be accounted for by their latching on to the closest word they can find in their diction-ary that is in the least bit like the word *wid-get*. Actually, *widgeon*

(d) anchor in straddle below straddle

Figure 10-19. Bottom frame placement in multicolumn documents

Chapter 11

Text in Frames

You can create three types of text objects in frames: *microdocuments, text strings,* and *graphic fonts.* All three are diagramming objects, but the methods for creating and editing each type are quite different.

A **microdocument** is a document you create within a frame; each microdocument has its own set of page properties. When you edit a microdocument, you access the same text and component editor available in the main document.

Since microdocuments and main documents share the same editor, you can transfer text and components between the two. For example, you can **Cut** text from outside a frame and **Paste** into a frame to create a new microdocument. Or you can **Copy** components from a microdocument and paste them into the main document. In addition, you can manipulate the microdocument like other diagramming objects. This combination of text processing and diagramming within a frame gives you increased flexibility in page layout.

Text strings are discussed in the chapter *Text as a Diagramming Object.* For information on graphic fonts, see the chapter *The Graphics Cabinet.*

Identifying Text Objects

To identify which type of object a given text object is, select the object. If it is a *graphic font,* the object will blink and you will see dots representing control points surrounding the selected words or characters. If the object is a *text string,* the text will blink. If it is a newly created *microdocument,* the text and a gray bounding box will blink alternately. Figure 11-1 shows all three types of objects in animation state; that is, after executing the **Move** command.
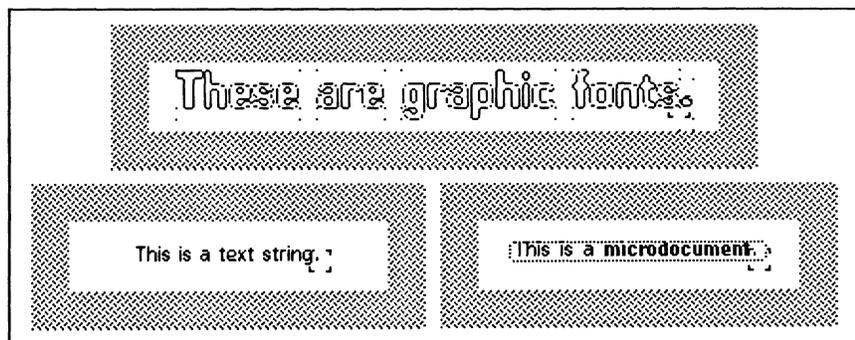
Figure 11-1. Identifying text objects

Microdocuments

In this section, we discuss the following topics:

- the editing environment
- microdocument fundamentals
- how to create microdocuments
- how to edit microdocuments
- how to manipulate microdocuments as diagramming objects

Microdocuments enable you to access text processing features while working in frames. The editing environment in a microdocument is essentially the same as the editing environment in the main document. Microdocuments are different in a few ways:

- each microdocument is currently limited to a single page
- the microdocument page height changes dynamically
- microdocuments cannot contain frames

Understanding the Editing Environment

The Document Window

Within each document window there can be only one **active document**, which can be a microdocument or a main document. A microdocument becomes active when you edit it; the main document is active when all of the microdocuments it contains are closed. The active document takes control of some parts of the document window.

Parts of the Window Controlled by the Active Document

Figure 11-2 shows how the active document affects the document window. Both the component caret and text caret appear in the active document. As illustrated, the active document determines:

- the component names displayed in the component bar
- the current font and attributes displayed in the document header boxes
- the messages displayed in the status line

For example, since the main document is active in Figure 11-2a, the component bar displays only the main document components. In Figure 11-2b a microdocument is active, therefore the component bar displays only the components in that microdocument.

(a)
Main document is active.

(b)
Microdocument is active.
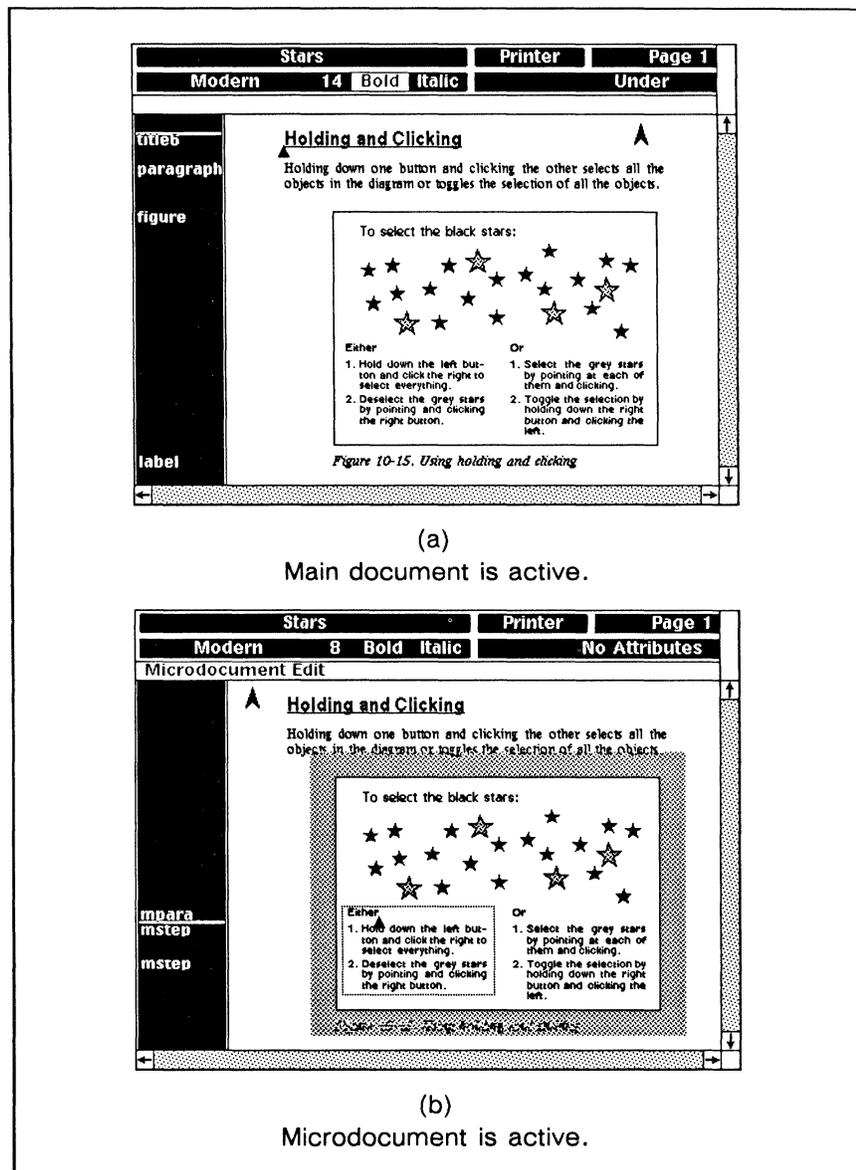
Figure 11-2. Effects of the active document on the document window

Parts of the Window that are Constant

Certain parts of the document window are constant whether the active document is a main document or a microdocument. The following commands affect the entire document or document window:

- the pulldown commands from the top row of document header boxes *except* **Properties** on the Page pulldown menu
- all commands on the Window popup menu

For example:

- **Save** on the Document Name pulldown menu saves all changes you have made to the *entire* document since the last save.
- all of the paging commands on the Page pulldown menu control which page of the *main* document displays

If you use the scroll bar to scroll an open frame containing an open microdocument completely out of the document window, the software automatically closes the microdocument and frame.

The Frame

You can create any number of microdocuments in any new or existing frame.

Just as the document window determines how much of your main document is visible, the frame controls what portion of your microdocument is visible. Consider the following similarities between the document window and the frame:

- the document window may be too small to display an entire page; the frame may conceal some of your microdocument
- as you type or move the text caret in the document, the document scrolls when the text caret reaches the window; as you type or move the text caret in the microdocument, the microdocument scrolls when the text caret reaches the frame border

Concept of Global vs. Local

Within the document window there are two categories of commands and resources.

- **Global** commands and resources extend across the boundaries of the main document and all its microdocuments.
- **Local** commands and resources affect only the active document.

Global Resources and Commands

Microdocuments and the main document share several global resources:

- component master definitions
- autonumber streams
- autoreference tokens
- index tokens

Microdocuments and the main document access the same component master definitions. Each time you use **Global Apply** or **Unify** to make changes to a compo-

nent, those changes are applied to all of the components with the same name in both the main document and microdocuments. There are just a few exceptions pertaining to components in microdocuments.

- **Begin New Page** on the Component Page property sheet is not effective in microdocuments. The software disregards that setting in microdocuments because microdocuments are limited to a single page.

- Changes made to the master for *caption*, the default microdocument component, are not applied to *caption*s in variable-width microdocuments. For details, see the section *Types of Microdocuments*.

- If you edit a microdocument in a header/footer frame and try to globally apply or unify component property changes, you may see the message: *Cannot unify or apply the properties globally.* This message appears only when an instance of that component exists in the main document. The software prevents such changes because they can cause the document page breaks to change, which can force the elimination of the header/footer frame currently being edited.

Microdocuments and the main document access the same autonumber streams. When you create an autonumber token from *streamA* in the main document and then create another token from *streamA* in a microdocument, the two tokens belong to the same stream and will be numbered consecutively.

You can use autoreferences globally. For example, you can select an autonumber token that you have created in the main document, execute the **Copy Ref.** command to create an autoreference token, and then paste the reference token in a microdocument. The autoreference token in the microdocument will refer to the autonumber token in the main document.

You can create index tokens in both the main document and microdocuments. The software uses all of these tokens to compile a single index for the entire document.

For information on tokens in shared-content frames, see the chapter *Text-Anchored Frames*.

Most of the commands on the Text and Component popup menus are local commands. Below we list the *global* commands available on these menus. If a command is not listed as global, you can assume that it has local impact.

- **Masters** command for purging components, frames, and autonumber streams
- **Tabs** for determining the display characteristics of tabs

- **Show** for toggling the display of components, tabs, hard re-turns, index tokens, frame anchors, text anchors and the gray bounding box of empty microdocuments
- **Find** for searching for text, a frame, or an index, autonumber, or reference token
- **Replace** for replacing one string of text with another
- **Spelling** for searching for and correcting spelling errors

For example, the [Show →] [Index Tokens] command toggles the display of index tokens in both the main document and in microdocuments. Similarly, when you search for text or for a spelling error, the software searches through the main document and microdocuments.

Document-wide Find, Replace, and Spelling operations do not include microdocuments in header/footer frames.

Local Resources and Commands

Currently you cannot create frames within microdocuments. Consequently, microdocuments cannot contain images or diagramming objects.

The first command on the Text Location menu in a microdocument is **Close** (Figure 11-3). This command closes the open microdocument, leaving the frame open. When the microdocument is closed, you can select and manipulate the microdocument using diagramming commands.

```
Close
Create  →
Paste    □
Fonts   →
Center
Misc    →
```
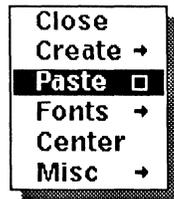
Figure 11-3. Microdocument Text Location popup menu

Keyboard Commands

All of the keystrokes available in the text of the main document are available within microdocuments. The domain of a keyboard command is consistent with that of its corresponding menu command.

Microdocument Fundamentals

All newly created microdocuments share certain properties, such as the *gray bounding box* and the default component named *caption*. But many of the properties of a microdocument depend on which type of microdocument you create.

The Bounding Box

When you edit a microdocument, it will be surrounded by a gray bounding box that has a unique dotted pattern ⬚. This box:

- indicates the page dimensions of the microdocument
- contains white fill during the microdocument edit; the fill blocks the display of anything that appears in back of the microdocument and makes the microdocument text easier to see and edit
- disappears when you close a microdocument
- does not print

Since microdocuments are diagramming objects, you can give them diagramming **Width**; that is, you can change the bounding box from nonprinting gray (width setting of *None*) to a printing box made of black or white solid or dashed lines that can have different thicknesses. By setting the width, you determine the appearance of the bounding box for the closed microdocument; the gray box will always appear when you edit the microdocument.

When a microdocument is closed and unselected:

- if the width of the box is *None* and the microdocument is empty, you will see the gray box
- if the microdocument has a black or white width, a box with the specified width displays regardless of the contents of the microdocument

You will see a box that reflects the current width setting (gray, black, or white):

- when you select a closed microdocument (the box blinks)
- when you execute an animation command after selecting a closed microdocument

Types of Microdocuments

The method you use to create a microdocument determines the type of microdocument you create.

The simplest way to create a new **variable-width** microdocument is to open a frame and begin typing. The microdocument bounding box and page width grow as you type each character.

To create a new **fixed-width** microdocument, you can open a frame and execute the `Create →` `Misc →` `Text` command. A microdocument with a page width of three inches will appear. You can immediately size or move this microdocument, or edit it by executing `Props →` `Edit` .

The names of the two types of microdocuments refer to their page width. The width of a variable-width microdocument changes as you add text to it; the width of a fixed-width microdocument is fixed so that text wraps around when it reaches the right page margin.

A variable-width microdocument can be no longer than one line; a fixed-width microdocument can be one or more lines. When a variable width becomes two lines long, the software converts it to fixed width.

The Default Component: Caption

Every microdocument you create using the two basic methods contains the default component named *caption*. The software creates the master definition for *caption* the first time you create a microdocument in a document. The properties of this master are nearly identical to the properties of the default *paragraph* master. *Caption* is different in that:

- the *Line Spacing* is 1.1
- the *Font* and *Alignment* vary depending on the method you use to create the microdocument

The *Font* of the *caption* master is set in each document when you create the first microdocument. The current font displayed in the Font box becomes the default font. By default, the *Alignment* is left and right justified. You can change the master definition for *caption*; however, such changes will apply only to those instances of *caption* in fixed-width microdocuments.

Instances of the *caption* component in variable-width microdocuments will always receive the original *caption* properties (except the *Font* and *Alignment* settings) and will not be affected by changes to the master. The software preserves the properties of all instances of *caption* in variable-width microdocuments in order to maintain the microdocument alignment, which you can specify for each variable-width microdocument you create. Furthermore, if you open the property sheet for a *caption* in a variable-width microdocument, change the properties, and apply the changes, the software automatically converts the microdocument to fixed-width.

The *Font* of a variable-width microdocument is the current font displayed in the Font box. You can change the default font by setting the current font before you begin typing. For information on the *Alignment* of variable-width microdocuments, see the section *Variable-Width Microdocument*.

When to Use Each Type

You can use variable-width microdocuments in any case where you formerly would have used *text strings*. For example, you can use them to create captions for illustrations. Or you can replace the text anchors in your default document with empty variable-width microdocuments.

Fixed-width microdocuments are useful when you want to put multiple lines of text into a frame. Using fixed-width microdocuments you can easily fit copy into a specified area. If you are not satisfied with a layout you have created using microdocuments, you can move and size the microdocuments to explore layout changes.

Creating Microdocuments

Variable-Width Microdocuments

There are two ways to create a variable-width microdocument.

- typing in an open frame
- pasting a variable-width microdocument that you cut, copied or duplicated from an existing microdocument (represented by a diagramming icon on the clipboard)

When you create a microdocument by typing, with each character you type the bounding box grows horizontally and the microdocument page width changes dynamically. As long as the microdocument remains a single line and you continue to type, the microdocument page width continues to change. Even after you close and reopen a variable-width microdocument, its width remains variable.

Alignment

By default, when you open a frame and start typing you create a left-justified microdocument. Using keyboard commands, you can specify the horizontal alignment of the variable-width microdocument you are about to create. Just execute one of the alignment commands and begin typing. Use:

- **Ctrl-l** to create a left-justified microdocument
- **Ctrl-c** to create a centered microdocument
- **Ctrl-r** to create a right-justified microdocument

By using these commands, you are setting the component alignment for the *caption* created in that microdocument. For example, if you press **Ctrl-c**, the newly created microdocument will contain a *caption* with Centered alignment.

Regardless of the alignment you select, text grows from the position of the diagramming cursor. Figure 11-4 shows three microdocuments created when the diagramming cursor was positioned at the vertical line. **Ctrl-l** causes the text to grow

from left to right; **Ctrl-c** makes it grow in both directions; with **Ctrl-r**, text grows from right to left.
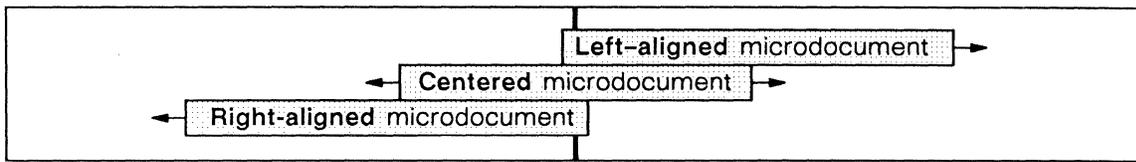
Figure 11-4. Microdocument alignment

You can also change the alignment of an existing variable-width microdocument by selecting the microdocument and executing:

- a keyboard alignment command
- an **Align** command

In some cases, the **Size Horizontal, Size Diagonal, Size Reflect Horizontal** and **Size Reflect Diagonal** commands will change the alignment of existing variable-width microdocuments. For details, see the chapter *Text as a Diagramming Object*.

Every time you open a frame the default alignment for text objects is reset to left-justified. However, as you create text objects in a frame the default alignment changes every time you:

- use a keyboard command or an **Align** command to specify an alignment that differs from the current default
- select a text string or variable-width microdocument that has a different alignment than the current default

For example, if you enter **Ctrl-c** and create a centered microdocument, the next text object you create will be centered by default. Similarly, if you select a right-justified microdocument, the next text object you create will be right-justified.

When you align graphic fonts, you do not affect the alignment default for other text objects.

How to Create a Variable-Width Microdocument

Before you create a new microdocument, make sure that all objects are deselected and that no microdocument is open.

To create a new variable-width microdocument by typing:

⤶ Create a frame and open it.

↖ Point the diagramming cursor where you want to position the baseline of the first line of text (Figure 11-5).

Figure 11-5. Position of the cursor before creating a microdocument

⊡ To specify alignment, type **Ctrl-c** for centered text, or **Ctrl-r** for right-justified text.

When you open a frame, the default alignment is left-justified.

⊡ Start typing.

The software creates and opens a variable-width microdocument. As Figure 11-6 shows, the bounding box grows horizontally to accommodate the new text.

This is a variable-width microdocument.

The microdocument expands horizontally as you type.

This is a variable-width microdocument. It grows horizontally as I type.
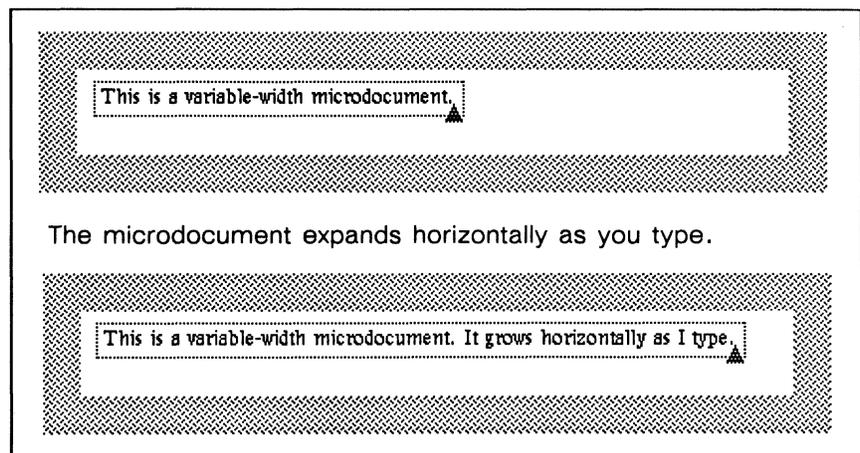
Figure 11-6. Creating a microdocument by typing in a frame

To create an empty variable-width microdocument, create the microdocument by typing any character and then delete the character.

Converting a Microdocument from Variable-Width to Fixed-Width

For the text in your microdocument to wrap around, the microdocument must have a fixed page width. Any action that causes a variable-width microdocument to become longer than one line converts the microdocument to fixed-width. These actions include:

- pressing the RETURN key
- creating a second component using the **Create** command or LINEFEED key
- typing until the microdocument is over 24 inches in width
- pasting into a microdocument so that microdocument width exceeds 24 inches
- applying changes to the Component property sheet or Page property sheet

Once you convert a microdocument to fixed-width, you cannot convert it back to variable-width.

To convert a microdocument from variable-width to fixed-width:

✔ When the first line of the microdocument has reached the desired page width, press the RETURN key.

or ✔ Position the text caret at the location in text marking the desired page width and press the RETURN key.

Use this step if the microdocument is wider than the desired page width.

✔ Press the DEL key to remove the return character.

Figure 11-7 shows the microdocument in Figure 11-6 after it has been converted to fixed-width. Note that the microdocument has grown vertically to accommodate the new text.

This is a variable-width microdocument. It grows horizontally as I type. Press RETURN and the microdocument becomes fixed-width and grows vertically.
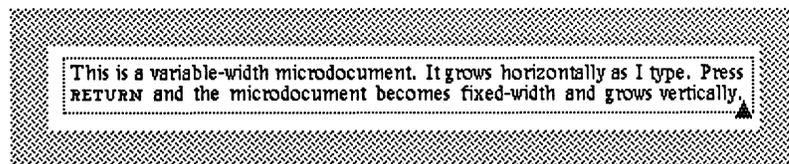
Figure 11-7. A microdocument converted to fixed-width

Any text you type subsequently will wrap to fit within the microdocument page margins. The top edge of the bounding box will remain stationary while the bottom edge drops when the text caret reaches the right page boundary. When you delete a line or more of text, the bounding box shrinks upward.

Fixed-Width Microdocuments

A fixed-width microdocument has a fixed page width and cannot be converted into a variable-width microdocument. All changes made to the *caption* master will apply to all instances of *caption* in fixed-width microdocuments.

How to Create a Fixed-Width Microdocument

You can create a fixed-width microdocument by:

- executing [Create →] [Misc →] [Text] on the Diagramming Nothing Selected popup menu
- pasting text, components, or a document into an open frame
- pasting a microdocument that you cut, copied or duplicated from an existing fixed-width microdocument
- converting a variable-width microdocument

When you use any of these methods except the last one, the software creates a microdocument that is closed but in a selected state so that you can immediately manipulate it using **Size**, **Move**, or **Props Font**. You can also edit it by executing **Props Edit**.

To create an empty fixed-width microdocument:

✔ Create a frame and open it.

◻▮◻ Execute [Create →] [Misc →] [Text] on the Diagramming Nothing Selected popup menu.

As Figure 11-8 shows, a bounding box will appear indicating the position and size of the microdocument. The microdocument will be in selected state, enabling you to immediately size, move, fill or edit it.
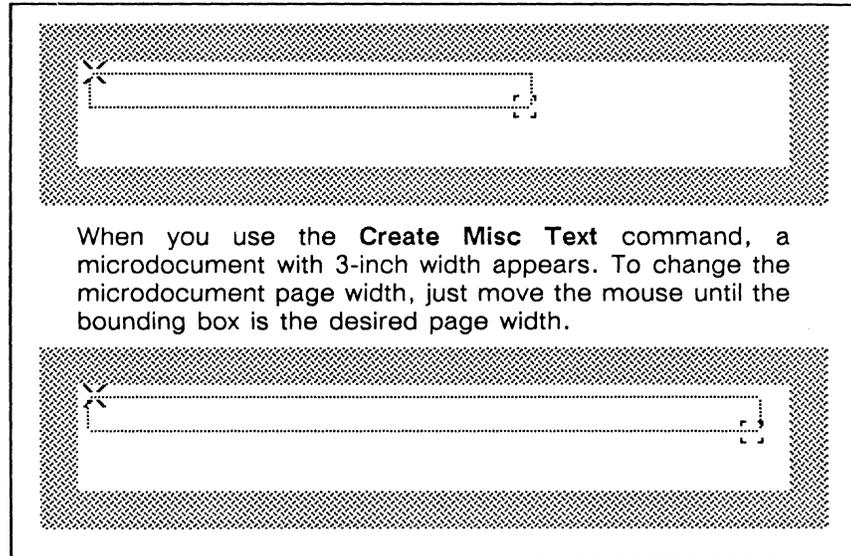
Figure 11-8. Creating an empty microdocument

⬛⬛ To open the microdocument, execute the `Props ➜` `Edit` command on the Diagramming Object Selected popup menu.

You will see the message: Microdocument Edit *in the document status line.*

As you type, the text you create will wrap around when it reaches the right page boundary and hyphenate as appropriate. The bounding box will grow vertically to accommodate the new text.

Cut, Copy, and Paste

You can use **Cut, Copy,** and **Paste** to move and copy text and components from the main document to a microdocument and, conversely, from a microdocument to the main document. These commands work in an open microdocument just as they do in a document. In addition, you can paste text directly into an open frame; this will create a new microdocument.

Cutting and Copying

When you cut or copy a *closed* microdocument, a *graphic icon* appears on the clipboard. When you cut or copy text or components from an *open* microdocument, a *document icon* appears on the clipboard.

Pasting into Frames

You can paste a microdocument (graphic icon) into a frame just like any other diagramming object.

If you paste text or components into an open frame, the pasted material will form a new fixed-width microdocument.

Pasting into Microdocuments

You can paste into a microdocument anything that appears on the clipboard as a *document* icon; that is, text or components that you have cut or copied from a document or microdocument.

Note that microdocuments cannot contain frames. If you paste text or a component containing frame(s) into an open microdocument, a stickup will appear requiring you to confirm or cancel the operation. The stickup warns that if you confirm, the frame(s) will be removed and replaced with the @ symbol (Figure 11-9). By confirming, you lose the contents of the frame.

To complete the paste operation, the frame(s) in the text will be removed and replaced with the '@' character. Please confirm.

Confirm Cancel
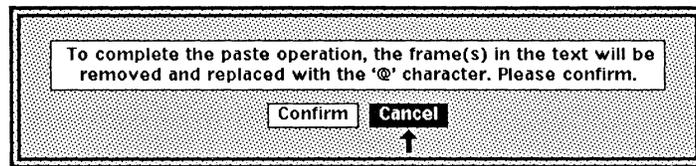
Figure 11-9. Pasting a Frame into a Microdocument stickup

If you try to paste a diagramming object or image directly into an open microdocument, you will see the message: *Either there is nothing to paste or the object cannot be pasted.* The graphic icon will remain on the clipboard.

Since a closed microdocument is a diagramming object, you cannot paste it into an open microdocument. To combine two microdocuments, open the first microdocument and cut its contents; open the second microdocument and paste.

Microdocuments cannot contain frames. If you paste a frame into a microdocument, you will lose the contents of the frame.

Pasting into the Main Document

When you paste a microdocument as a graphic object into a document, the software creates a frame named *Auto* that is automatically sized to the microdocument and places the microdocument in the frame.

You can paste text or components from an open microdocument directly into the main document. Text and components from microdocuments are pasted into the main document in the same way they are pasted from the main document. That is, if you paste text into the component bar, it retains component information; if you paste two or more components as text, you will receive a stickup warning you that you will lose component information.

Editing Microdocuments

This section discusses several topics related to editing microdocuments including:

- how to open and close microdocuments
- the scrolling behavior of microdocuments
- the Microdocument Page property sheets

Opening Microdocuments

There are two methods for opening a microdocument:

- selecting the microdocument and executing **Props Edit**
- pointing the diagramming cursor where you want to add a word, selecting the microdocument, and typing

When you open a microdocument in a frame in which the *grid* is on, the grid disappears during the microdocument edit. When you close the microdocument, the grid will reappear.

To open a microdocument:

🡒 Open the frame containing the microdocument.

🖉 Point the diagramming cursor at text within the microdocument where you want to position the text caret.

🡒 Select the microdocument.
The microdocument will blink.

🡒 Execute the ⌈Props→⌉ ⌈Edit⌉ command on the Diagramming Object Selected popup menu.
The text caret will appear at the diagramming cursor position.

To open and edit a microdocument using the point-and-click method:

🡒 Open the frame containing the microdocument.

🖉 Point the diagramming cursor at the position in text where you want to position the text caret.

▐▢▢ Click the **left** mouse button.
The microdocument will blink.

🡒 Type the word(s) you want to add.
When you point, click and begin typing, the software opens the microdocument and adds the new text at the diagramming cursor location.

Scrolling Behavior

While you are editing a microdocument, the text caret may reach the frame border (Figure 11-10a). When it does, the software scrolls the microdocument and the contents of the frame either horizontally or vertically so that the entire microdocument is visible (Figure 11-10b). When you close the microdocument, the software scrolls the microdocument and frame contents back to their original position. Part of the microdocument will be hidden (Figure 11-10c). If you want to display the entire microdocument once it is closed, enlarge the frame or resize the microdocument.

(a)
The text caret reaches the frame border.

(b)
Since the caret hits the right and bottom borders, the microdocument scrolls *horizontally* and *vertically*.

(c)
The closed microdocument scrolls back
to its original position.
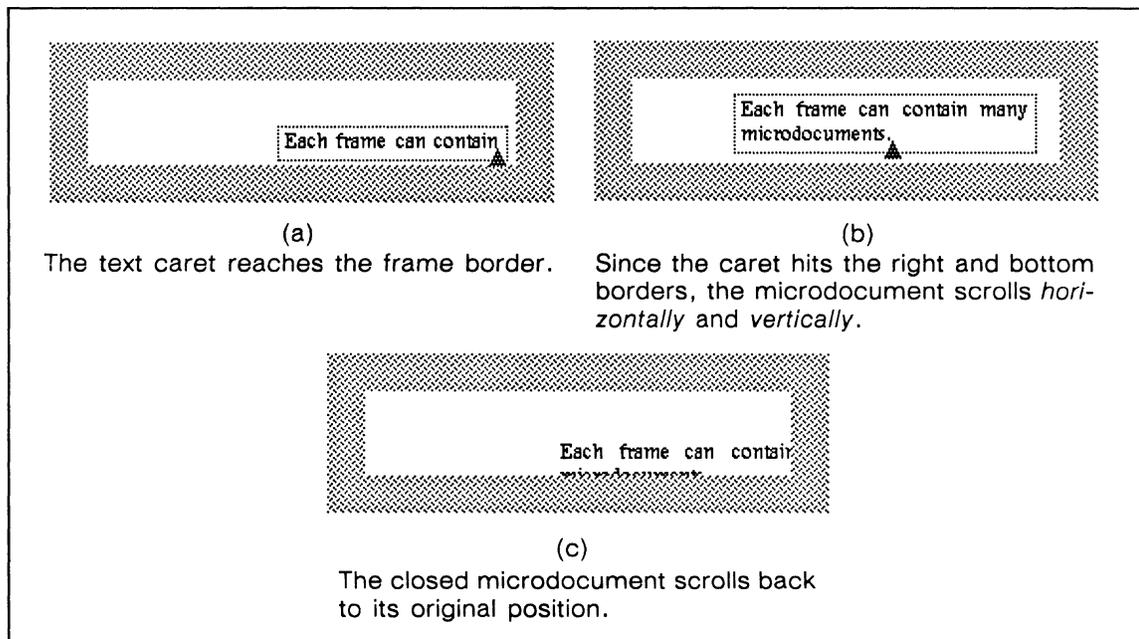
Figure 11-10. Microdocument scroll during edit

Any action that causes the text caret in a microdocument to reach the frame border will cause the microdocument to scroll. Additional methods of scrolling an open microdocument are:

- using the arrow keys or keyboard commands (except for the page scrolling commands) to move the text caret to the frame border
- executing the **Center** command on the Text Location menu
- pointing the text caret outside the frame border at an area where you know microdocument text is hidden by the frame and clicking the left button

When a microdocument is open, you will see only the components names for components whose text is visible in the frame.

Closing Microdocuments

When you close a microdocument, the gray bounding box disappears unless the microdocument is empty.

To close a microdocument:

       ✔ Execute the **Close** command on the Text Selected popup menu.

  or ✔ Point the mouse cursor outside the bounding box of the open microdocument and click the **left** button.

> *The microdocument will close. Depending on where you point the cursor, other things may happen. If you point at another diagramming object, it will be selected. If you point the cursor outside the frame, the frame will close.*

Microdocument Page Property Sheets

Each microdocument has its own set of page properties. When you open a microdocument and select the **Properties** command on the Page pulldown menu, you will see one of the two Microdocument Page property sheets, *Normal* or *Custom* (Figure 11-11). The properties available on these sheets are a subset of the properties on the Document Page property sheets and perform the same functions but apply only to the *open* microdocument.

The Normal Sheet

The Custom Sheet
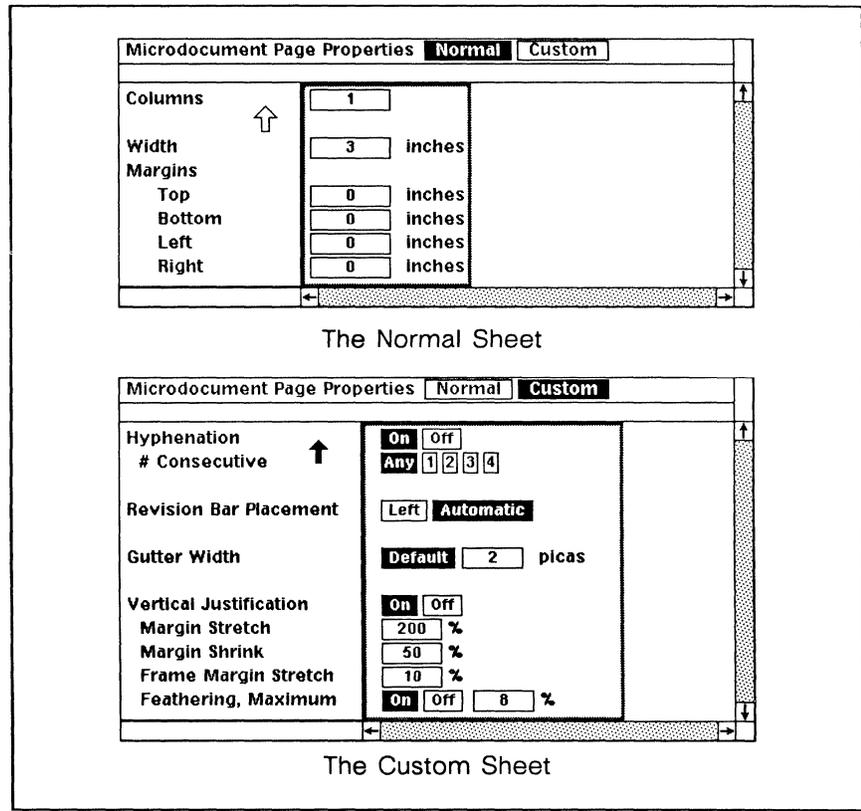
Figure 11-11. Microdocument Page property sheets

The Normal Sheet

On the Normal sheet, use:

- *Columns* to transform your microdocument from single column into multicolumn
- *Width* to change the size of the microdocument
- *Margins* to change the microdocument page margins

When you change the value of *Width* or *Left* or *Right* page margin, the number you enter must allow space for the text as well as the page and component margins. If the number you enter is too small, the software will not let you **Apply** and will display the message: *Width / left margin / right margin of the page is inappropriate.*

The value in the *Width* box changes when you use the diagramming **Size** command to resize a microdocument.

The Custom Sheet

On the Custom sheet, use:

- *Hyphenation* to control the amount of hyphenation in your microdocument

- *Revision Bar Placement* to control the placement of revision bars

- *Gutter Width* and *Vertical Justification* to specify properties of a *multicolumn* microdocument

The default revision bar setting is *Left*; that is, all revision bars appear in the left margin of a single-column microdocument and to the left of each column in multicolumn microdocuments. Select the *Automatic* box to see the following results:

- Two-column microdocuments will have revision bars on the outside of each column (i.e., to the right of the right *column* and to the left of the left *column*).

- Microdocuments with more than two columns will have revision bars on the left of all columns but the rightmost column, where they will be on the right.

The *Automatic* setting does not affect single-column microdocuments.

Revision bars may overlay the text within the microdocument. To correct this, adjust the appropriate margin (left or right) or the gutter width, depending on placement of the revision bar. The margin or gutter where the revision bar appears must be wider than the revision bar, which is 3/32 (or .09) inches wide.

Note that revision bars will not automatically line up with revision bars in other microdocuments or outside the frame.

Sizing Microdocuments

You can directly control only the *width* of a microdocument. The height of the microdocument depends on several factors. One way to change the dimensions of a microdocument is to change the value of *Width* on the Microdocument Page property sheet. As you change this value, the software increases or decreases the height to accommodate the text and margins in the microdocument (Figure 11-12).
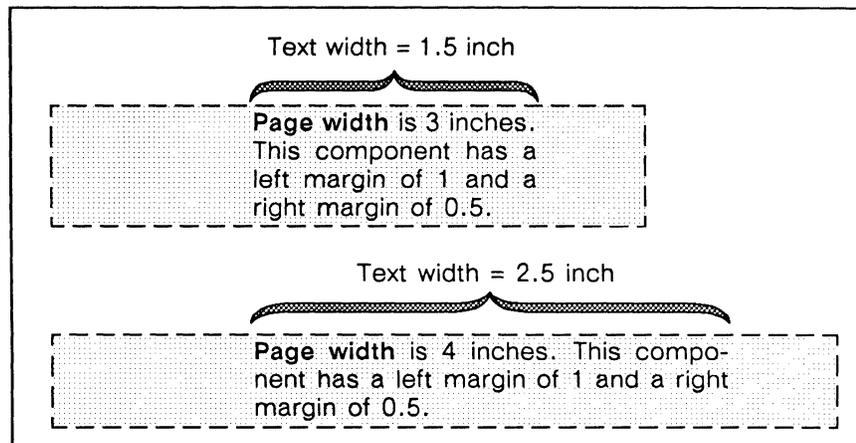
Figure 11-12. Sizing microdocuments

To change the size of an open microdocument:

☑ Make sure the microdocument is open.

The text caret should be visible within the gray bounding box.

☑ Execute the **Properties** command on the Page pulldown menu.

The Microdocument Page property sheet will appear.

☑ Make sure the Normal sheet is open.

☐ Type the desired page width into the *Width* box and **Apply**.

The dimensions of the bounding box will change and the text will rehyphenate.

☑ Execute the **Close** command.

Maximum Microdocument Page Size

All microdocuments are limited to a single, independent page; text will not flow from one microdocument to another. The maximum size of a microdocument page is 24 inches x 24 inches (width by height). This limit enables you to display 24 horizontal and vertical inches of text in each microdocument, although you cannot print or view at once an entire microdocument this size.

If you paste many pages of text or components into a microdocument, text and components that do not fit within the first 24 vertical inches will spill into an overflow area for that microdocument. You cannot *directly* access what appears in this overflow area. However, you can shift text from the overflow area to the first page by increasing the microdocument page width, if possible, or cutting the components that appear on the first page of the microdocument.

If you create a 24 square-inch microdocument, there are two methods for viewing the entire microdocument in segments. Either move the text caret within the microdocument so that the microdocument scrolls vertically or horizontally, or in-

crease both the document page dimensions and frame dimensions to 24 square inches, align the top and left sides of the microdocument to the top and left sides of the frame, and scroll the document page.

Manipulating Microdocuments as Diagramming Objects

When a microdocument is not being edited, you can:

- select one or more microdocuments and execute most diagramming commands, including **Size**, **Move**, **Align**, **Dup**, **Cut** and **Props Font**
- give the microdocument **Width** and **Fill**
- see the bounding box of an empty microdocument, even when it is not selected

Not all diagramming commands affect a single microdocument. You can cut, copy, duplicate, or paste a microdocument, and you can size it and move it around, but you cannot rotate it.

You can rotate a group that contains microdocuments. You can also rotate two or more microdocuments. Such a command does not affect the microdocuments themselves, but only their relation to each other.

You can also use **Align** to line up two or more microdocuments.

For details on using diagramming commands to manipulate microdocuments, see the chapter *Text as a Diagramming Object*.

Cutting Microdocuments

At some point you may decide to cut a microdocument. To **Cut** a microdocument, it must be closed. Select and cut it just as you would any other diagramming object.

You cannot remove a microdocument by cutting its contents. If you select and cut all of the components or text in a microdocument, the bounding box containing an empty component will remain.

Width and Fill

You can fill and give width to a microdocument just as you would any diagramming object (Figure 11-13). When you open the microdocument, the width continues to display while the fill is temporarily replaced with white.
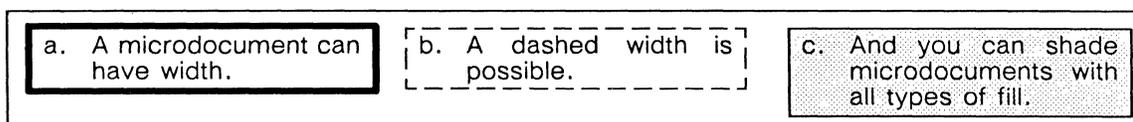
Figure 11-13. Microdocument width and fill

Microdocument Font

You can also change the font of an *existing* microdocument by selecting the microdocument as a diagramming object and executing a command on the Props Font submenu. Changes made using this submenu will change the font setting on the Component property sheets of all components in the microdocument.

Page Layout with Microdocuments

Microdocuments enable you to explore complex page layout. This section presents samples of three types of documents that require a unique layout — *tables*, *newsletters*, and *forms* — and provides some practical information and guidelines on using microdocuments.

The Table

Figure 11-14 shows a simple table created with many boxes and microdocuments.

Places to Visit in New York City		
Name	Description	Practical Information
American Museum of Natural History	The collections fill 11.5 acres (Michelin) or 25 acres (MTA) of exhibition halls on four floors. There is a full-sized replica of a blue whale and the world's largest star sapphire (stolen in 1964).	Hours 10-5, ex. Sun 11-5, Wed. to 9 Location Central Park West & 79th St. Subway AA to 81st St.
Bronx Zoo	About a mile long and .6 mile wide, this is the largest zoo in America. A lion eats 18 pounds of meat a day while a sea lion eats 15 pounds of fish.	Hours 10-4:30 winter; to 5:00 summer Location 185th St. & Southern Blvd., Bronx Subway To East Tremont Ave.
Brooklyn Museum	Five floors of galleries contain American and ancient art. There are American period rooms and architectural ornaments saved from wreckers, such as a classical figure from Pennsylvania Station.	Hours Wed-Sat, 10-5, Sun 12-6 Location Eastern Parkway & Washington Ave., Brooklyn. Subway 2,3 to Eastern Parkway
New York Historical Society	All the original paintings for Audobon's *Birds of America* are here, as are exhibits of American decorative arts, New York history, Hudson River school paintings, carriages, and glass paperweights.	Hours Tues-Fri & Sun, 1-5; Sat 10-5 Location Central Park West & 77th St. Subway AA to 81st St.

Figure 11-14. Sample table created with microdocuments

Figure 11-15 shows the boxes and microdocuments that compose this table. Note that there are three columns of microdocuments in the body of this table which correspond to the three table headings.

Figure 11-15. Microdocuments in the table

Every box in this table contains a microdocument, except for the pairs of boxes in the *Practical Information* column, which share a microdocument (Figure 11-16).

(a) Microdocument in a box   (b) Microdocument straddling two boxes

Figure 11-16. Microdocuments in relation to the boxes

Four different component masters were used to create the table: one for the headings and a different one for each of the columns. Each of the entries (i.e., *Hours*, *Location*, etc.) in the *Practical Information* column is a separate component. A left tab was used to align the information in the right-most box.

Suppose that you created this sample table and then decided to add information about the *admission* fee and *telephone* number for each place. To do this, you edit each microdocument under *Practical Information*, add two components to each (one for admission and one for telephone), and type the additional information. Then you discover that because the revised microdocuments have grown in length, they exceed the size of their surrounding boxes (Figure 11-17a).

To understand how to correct this problem, you can copy one **cell** from the table (i.e., a microdocument and the surrounding box or boxes), paste it in another frame, and examine its sizing behavior. If you position the diagramming cursor at the bottom right-hand corner of the cell, execute **Select All** and then **Size Vertical**, the selected objects remain anchored at the top while the box grows downward as you move the mouse downward (Figure 11-17b).

Hours	10-4:30 winter; to 5:00 summer
Location	185th St. & Southern Blvd., Bronx
Subway	To East Tremont Ave.
Admission	$1.00, free 2-5 T, W, Th
~~Telephone~~	~~212-933-1759~~

Hours	10-4:30 winter; to 5:00 summer
Location	185th St. & Southern Blvd., Bronx
Subway	To East Tremont Ave.
Admission	$1.00, free 2-5 T, W, Th
Telephone	212-933-1759

(a) Microdocument exceeding the box size

(b) Vertically sizing the contents of a cell

Figure 11-17. Sizing behavior of microdocuments and diagramming objects

Whether you are sizing many objects or just a few, the sizing behavior is consistent. Using this technique, you can size the entire table. The upper left-hand corner of the microdocuments and boxes will be anchored while you control sizing from the lower right-hand corner. Figure 11-18 shows the table after it has been sized vertically.

You can also use this technique to size the table horizontally. The **Size Horizontal** command causes the page width of microdocuments to expand and contract. In response to the changing microdocument width, the text in multi-line components may shift among lines and rehyphenate.

Places to Visit in New York City		
Name	Description	Practical Information
American Museum of Natural History	The collections fill 11.5 acres (Michelin) or 25 acres (MTA) of exhibition halls on four floors. There is a full-sized replica of a blue whale and the world's largest star sapphire (stolen in 1964).	Hours 10-5, ex. Sun 11-5, Wed. to 9 Location Central Park West & 79th St. Subway AA to 81st St. Admission Donation: $1.00 asked Telephone 212-873-4225
Bronx Zoo	About a mile long and .6 mile wide, this is the largest zoo in America. A lion eats 18 pounds of meat a day while a sea lion eats 15 pounds of fish.	Hours 10-4:30 winter; to 5:00 summer Location 185th St. & Southern Blvd., Bronx Subway To East Tremont Ave. Admission $1.00, free 2-5 T, W, Th Telephone 212-933-1759
Brooklyn Museum	Five floors of galleries contain American and ancient art. There are American period rooms and architectural ornaments saved from wreckers, such as a classical figure from Pennsylvania Station.	Hours Wed-Sat, 10-5, Sun 12-5 Location Eastern Parkway & Washington Ave., Brooklyn Subway 2,3 to Eastern Parkway Admission Free Telephone 212-638-5000
New York Historical Society	All the original paintings for Audobon's *Birds of America* are here, as are exhibits of American decorative arts, New York history, Hudson River school paintings, carriages, and glass paperweights.	Hours Tues-Fri & Sun, 1-5; Sat 10-5 Location Central Park West & 77th St. Subway AA to 81st St. Admission Free Telephone 212-873-3400

Figure 11-18. Table that has been sized vertically

The Newsletter

The newsletter shown in Figure 11-19a was created using microdocuments. At first glance you may think this layout was achieved using the multicolumn feature. Notice, however, that the two columns are not the same width.

Once you decide on the layout of your newsletter, you can create a template. Figure 11-19b shows the first page of the template used to create the sample. The template contains boxes which are printing-locked and position-locked; they act as a layout guide.

Figure 11-19c shows the template after microdocuments and diagrams have been added. As you can see, the microdocument text has been aligned to the locked boxes. Figure 11-19d shows all of the microdocuments selected, enabling you to see the dimensions of each microdocument as indicated by the bounding boxes.

After you have created a newsletter from your template, you can further enhance the template by positioning an empty fixed-width microdocument in each column. To do this, copy one microdocument from each column and paste into the template. Then align the text in these microdocuments to the locked boxes. Edit the microdocuments, select all of the components in each, and cut them. The next time you use the template, you can begin creating components and typing in these empty microdocuments.

(a) Sample newsletter

(b) Template used to create sample

(c) Template after microdocuments
and diagram have been inserted

(d) Newsletter with all microdocuments
selected so that bounding boxes show

Figure 11-19. Sample newsletter and template

Additional Samples

Figure 11-20a shows a sample form and Figure 11-20b shows another sample newsletter, both created with microdocuments and diagramming objects. You can find a copy of the newsletter in the *Text* folder of the *Samples* drawer in your *System* cabinet.

(a) Sample form

(b) Sample newsletter available in *Samples* drawer

Figure 11-20. Additional samples

Chapter 12

Header and Footer Frames

Header and footer frames are special kinds of shared content frames. Header and footer frames occupy the top and bottom margins of every page in a document. In these frames you can enter information such as the chapter name or number, page number, or document revision dates.

Header/footer frames share their contents throughout a document according to the layout chosen on the Page property sheet. These frames can be reduced to zero height but they cannot be cut from the page. You cannot change the properties of header/footer frames as you can with other types of frames. (See the chapter *Text-Anchored Frames*.)

Figure 12-1 shows open header and footer frames without any content.
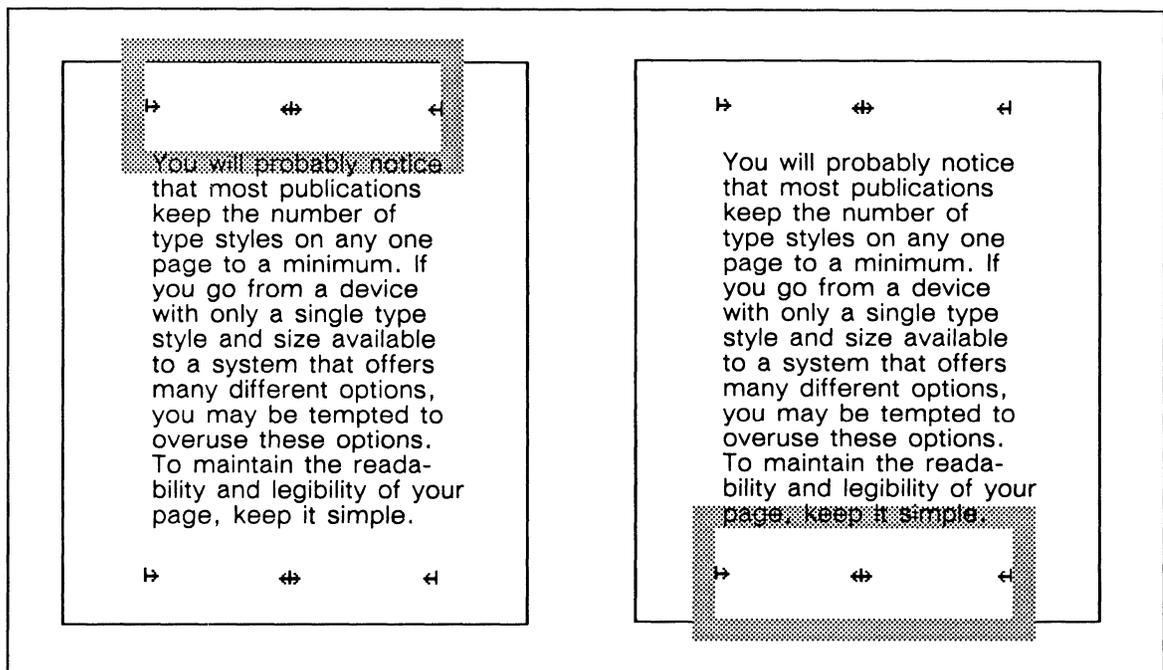
Figure 12-1. Open header (left) and footer (right) frames

Since headers and footers are located in frames, you can create and manipulate diagramming objects in them, enter multi-line, multi-font text, and create and edit microdocuments in a multicolumn format, if you choose. You can also paste text from your document and text and graphics from other frames into header and footer frames.

Figure 12-2 shows an example of some of the varied contents you can produce in header/footer frames.
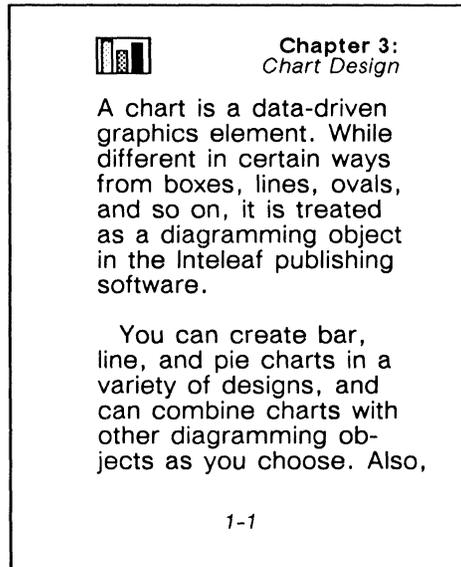
Figure 12-2. Header/footer frames containing graphics, multi-line, multi-font text, and page number

The major features of header/footer frames are:

- Header frames always fill the top margin of every page; footer frames always fill the bottom margin of every page.
- You can create a variety of text and graphic objects and position them anywhere inside header and footer frames.
- You can use tokens in frames so that page numbers or the document name automatically appear on each page.
- You can choose a layout in which unique headers and footers appear on the first page of a document.
- You can choose a double-sided page layout in which left and right pages have different headers and footers.
- Changes made to the contents of a header or footer frame are displayed in related frames as soon as you close the header or footer in which you made the change.

You can use the complete range of capabilities provided by Interleaf's diagramming features in header/footer frames. These commands allow you to:

- create boxes, lines, ovals, charts, and other basic objects
- rotate any object except text strings, microdocuments or charts
- size any object except text strings
- change the width of lines or the fill patterns in closed objects

- move, duplicate, group, or align any object in a frame
- define and show a grid that can be used to precisely align objects in a frame

For complete information on the use of these and related features, see the *Diagramming* chapters of the *Reference Manual,* Volume 2.

Creating Headers and Footers

Before you can create or change anything in a header or footer frame, you must open the frame.

Entering Text in Header and Footer Frames

There are two methods you can use to enter text in your header and footer frames: you can create text strings or you can create microdocuments. You can even use a combination of text strings and microdocuments in a single header/footer frame.

There are advantages to each method. When you create text strings, you can have them automatically positioned by using the text anchors that are in every header and footer frame, and you can have your document name automatically appear on each page of the document. On the other hand, it is easier to perform extensive editing and to create multi-line, multi-font text in microdocuments.

Using the Default Text Anchors to Create Text Strings

Header and footer frames contain three text anchors (Figure 12-1). The text anchors are there for your convenience — you are not required to use them. At the left margin of the frame there is a flush left text anchor, in the horizontal center there is a center anchor, and at the right margin there is a right text anchor. By default, the left text anchor ⊢ enters text flush left, the center text anchor ⊣⊢ enters text centered, and the right text anchor ⊣ enters text flush right.

If you want your text string positioned by an anchor, be sure to select the text anchor you want to use before you start to type a text string; the anchor will blink when selected. While a text anchor is selected, text you type at the keyboard is entered at that position.

To create text in headers or footers using the text anchors provided:

    ✔ Open the frame.

    ✔ Select the anchor for the text you want to enter first.

▣ Type the text you want to display in that position. If you want to use multi-line text, press **RETURN** at the point you want to end your first line, then continue typing on the second.

Any text you type after you press **RETURN** *constitutes a separate text string.*

↙ If you want to insert additional text in another location in that frame, select another text anchor in that frame and type the text you want to display. Repeat for the third text anchor if desired.

The text you enter will appear in the identical position in every header and/or footer in a single-sided document with no first page differentiation. For double-sided documents or headers and/or footers with a unique first page, see the Page Layout *sections, later in this chapter.*

Creating Text Strings without Using the Default Anchors

To enter a text string without using one of the three default text anchors to position it, you begin by executing one of three Control sequences: **CTRL l**, **CTRL r**, or **CTRL c**, followed by **CTRL o**. If you enter **CTRL l**, **CTRL o**, your text will grow from left to right. **CTRL r**, **CTRL o** makes your text grow from right to left; and **CTRL c**, **CTRL o** causes text to be centered at the position of the mouse cursor. In each case, the text grows from the position of the mouse cursor. Once you have typed your text, you can use the **Move** commands on the Diagramming Object Selected menu to position the text strings anywhere you like inside the frame.

If you simply begin typing in an open header or footer frame, the text you enter will be a microdocument.

See the chapter *Text in Frames* for more information on text anchors and for details on creating microdocuments.

Deleting Text Strings without Deleting the Text Anchors

Often you may want to delete all or part of a text string but leave its anchor in place. If you select a text object and use the **Cut** command, the anchor will be deleted along with the text. You can use the keyboard to delete text without deleting the anchors.

To delete a text object without deleting its anchor:

▤ Select the text object.

▣ Hold down the **CTRL** key and press **d**.

The contents of the text object are deleted, but the anchor remains in place and is still selected so you can immediately enter new text in that location.

Use the **DEL** key to delete text in a header or footer when you want to delete only part of the the text string.

Text Anchor Visibility

A text anchor indicates the position and alignment of a text object that does not contain characters. While a text object contains characters, its anchor is never visible.

Text anchors are either all visible or all invisible in *closed* frames throughout a document. If you do not want to see the text anchors in header/footer frames, or if they are currently invisible and you do want to see them, you can use the **Text Anchors** command on the Misc Show submenu of the Text Location popup menu. This command toggles the visibility of text anchors in all frames — both header/footer and text-anchored. When you open a frame, its text anchors become visible no matter what the Show selection is.

Editing Text in Header/Footer Frames

You can make a wide range of changes inside header/footer frames.

Changing Header/Footer Fonts

When you change the font of any text element (an anchor, a text string, or a microdocument) in a header/footer frame, the change affects all instances of *that element* in all related frames in the document. It does not affect any other text elements in the frame. For example, if you change the center footer font of the even-numbered pages of a document, the font change will be reflected in the center footer of every page of the document. The font of all other text elements in the footer will remain unchanged.

A text string is a text anchor plus contents. Each text anchor in a header/footer frame has a font property associated with it. (The default font property of the text anchors in a new document is 10-point Classic roman.) If you select an anchor and begin typing, the text string will be in this font unless you change it.

To change the font of a text anchor:

- ✔ Open the frame.
- ✔ Select the text anchor whose font you want to change.
- ✔ Select the font you want by executing the command(s) on any of the ⌈ **Props** → ⌋ ⌈ **Font** → ⌋ submenus.
- ✔ Select the anchor again and begin typing.
  The text you type in will appear in the font you have chosen.

Because an existing text string is associated with a specific anchor (which is sometimes invisible), you use essentially the same procedure to change the font of a text string as you use to change the font of a text anchor. Just select the text string and execute the [Props →] [Font →] submenu command(s) of your choice.

When you type a text string that *does not take its properties from a selected text anchor*, the font you get is the one that is displayed in the Font box of the document header. The information in the Font box reflects one of the following:

- the position of the text caret in the document text,
- the font of the most recently selected text string in the header/footer frame,
- the font you determined by making a change in the Font box

Always check the Font box before you enter a new text string.

To change the font before you enter a text string:

- ✔ Open the header/footer frame.
- ✔ Change the font in the Font box to the font you want to use.
- ✔ Move the mouse cursor back into the frame and type one of the following: CTRL l, CTRL r, or CTRL c, followed by CTRL o.
- ✔ Type the text you want.

Use this method when you are entering text that will not take its properties from a text anchor.

Changing Text Alignment

With keyboard commands, you can change the alignment of an existing text string or microdocument, or change the alignment prior to creating a new one. An existing string or microdocument must be selected before the command is executed.

If you want to change the alignment of an existing variable-width microdocument, select it, and enter CTRL l, CTRL c, or CTRL r for left, center or right alignment, respectively, and then type your text. By selecting a text string and using any of the alignment commands, you can change the alignment of a text string.

These changes are to the alignment of the *text string or microdocument itself*, not to its alignment to the frame. For example, if you make the alignment flush right, you fix the right edge of the text to its current position so that typing new characters will cause the text to grow to the left.

Editing Microdocuments in Header/Footer Frames

When you are editing a microdocument within a header or footer frame, you can only use **Global Apply** or **Unify** on a microdocument Component property sheet if *all components of that name exist only in microdocuments*. If you try to globally apply or unify changes to a microdocument Component property sheet for a component name that resides both in microdocuments and in the main document, you will get a message, *Cannot unify or apply the properties globally*.

However, you can perform the reverse operation: if you have a component name that resides in both microdocuments and in the main document, you can globally apply or unify a change to the Component property sheet in the *main* document (when the microdocument is closed) and have the change affect the component in microdocuments.)

Horizontal Reflection

The [Size →] [Reflect →] commands on the Diagramming Object Selected menu can be used to reflect the position of both diagramming and text objects in a header/footer frame. For reflecting text strings and microdocuments in header/footer frames, you will use the **Horizontal** command almost exclusively.

Horizontal reflection works only on selected objects in the frame. When you execute this command with two objects selected, the objects exchange positions. Characters in text strings and microdocuments maintain the proper order so that they read correctly. Figure 12-3 shows the results of a horizontal reflection in a header/footer where two objects (the cube and the text string or microdocument) were selected.
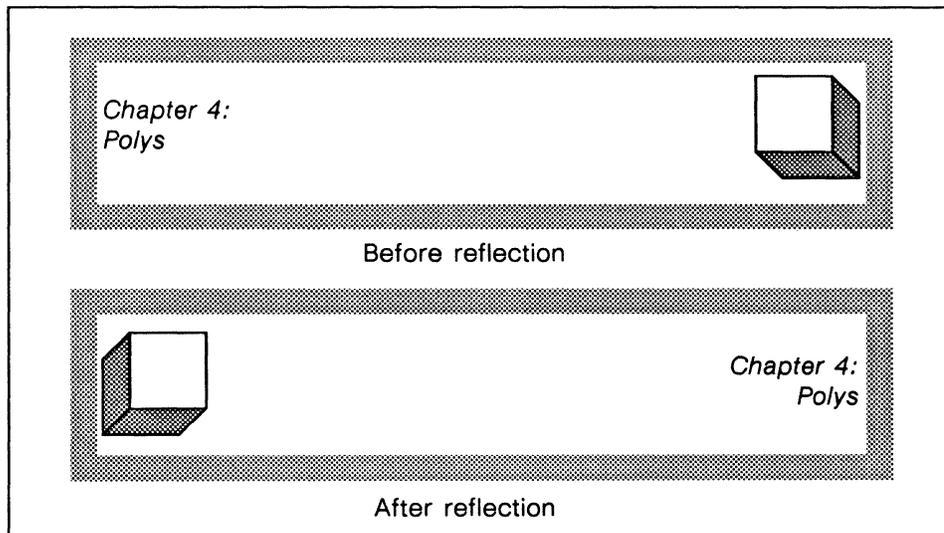
Figure 12-3. Horizontal reflection in header/footer frame

Because the software protects the readability of text objects, if you horizontally reflect a single text object, no change will occur in the location or appearance of

that object. If you select three objects in the default text anchor positions and a text object is at the center, that center object will undergo no change.

If you horizontally reflect a single graphic object, that object will be reflected around *its* left/right midpoint (Figure 12-4).

Before reflection   After reflection
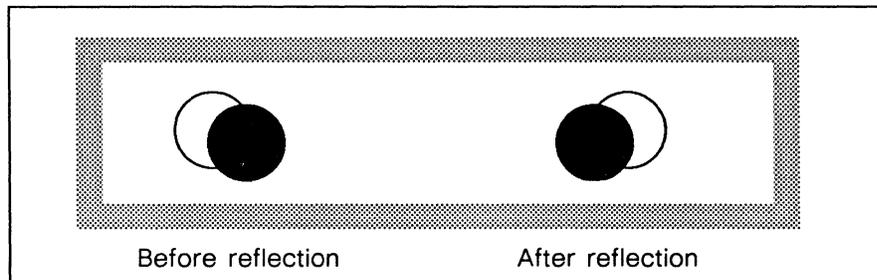
Figure 12-4. Single object reflected horizontally

To horizontally reflect the content of a frame:

    ✔ Select all of the objects you want to reflect.

    ✔ Execute the [Size →] [Reflect →] [Horizontal] command on the Diagramming Object Selected popup.

Horizontal reflection of headers and footers on alternating pages happens automatically when you change from a single-sided to a double-sided page layout.

Closing Header/Footer Frames

Before you can enter another header or footer frame, the frame you were working in must be closed. Since opening a new frame closes previously open frames, you often will not need a specific step in order to close the frame. However, when you must close a frame, you have several options to choose from.

To close a header or footer frame:

    ✔ Click the **left** or **right** mouse button while the cursor is outside the frame,

   or ✔ Scroll the frame out of the document window,

   or ✔ Execute the **Close** command on the Diagramming Nothing Selected menu.

Automatic Page Numbers and Document Name

Among the most common items of information included in headers or footers are the page number and the document name. When you use text strings in a header

or footer frame, the page number and the document name can be represented by tokens. To get automatic page numbers in microdocuments, you use the [Create →] [Ref. →] command on the Microdocument Text Location Selected menu. There is no automatic document name capability for microdocuments.

Tokens in Frames

A **token** is a character that is expanded into a text string. By entering a token that represents a *variable* instead of entering the *current value* of that variable, you can be sure that the correct value appears and prints in the frame, even when changes have been made. It is the expanded text string, not the token, that is always displayed and printed on the page.

Before entering a token, be sure to either select an anchor or enter one of the Control sequences described in the section, *Creating Text Strings.*

Page Number Token

ESC # When typed in any frame, this sequence enters a token for the number of the page the frame is on. In a header/footer frame, this sequence causes the page number to be displayed on each page, the first page only, or alternate pages, depending upon the page layout.

Document Name Token

ESC @ When typed in any frame, this sequence enters a token for the name of the document the frame is in. In a header/ footer frame, this sequence causes the document name to be displayed on each page, the first page only, or alternate pages, depending upon the page layout.

Automatic Page Numbering in Microdocuments

You can use the autoreferencing capability in microdocuments to create automatic page numbering.

To create automatic page numbering in microdocuments:

↙ Open a microdocument.

↙ Hold down the middle button to see the Microdocument Text Location menu and execute the [Create →] [Ref. →] command.

The Reference property sheet will open.

✔ Close the Reference property sheet without entering a tag.

By closing the property sheet without entering a reference tag, a self-reference is created for each related frame. Each self-reference translates into a page number.

All the page layout variables on the document Page property sheet apply to automatic page numbering in microdocuments created in header and footer frames. For example, if you use autoreferencing in the microdocuments in your headers and footers, and you turn on *Diff 1st Header* or *Diff 1st Footer*, the automatic page numbers will not appear in these locations. See the section, *Page Layout* later in this chapter.

If you want your page numbers to read Page 1, Page 2, and so on, for example, you can create a microdocument consisting of the word Page with a space after it, and then execute the [Create →] [Ref. →] command as outlined in the steps above.

Page Numbers in Headers/Footer Frames

To display the page number in the way that you want, at least one, and sometimes several steps are needed.

- If you want the page number to display anywhere other than the center footer in a default document, select and delete the existing page number token, and then insert the page number token at the position of your choice.

- If you want the starting page number to be anything other than 1, you must change the *Starting Page #* setting on the Page property sheet. If the document is one of several in a book, and is not the first document in the book, you may want to have the *Inherit* property turned on so that the document page numbering begins where the previous document numbers left off.

- If you want the starting page number to be shown in roman numerals rather than arabic, you must change the *Page # Style* setting on the Page property sheet.

If you are numbering the pages of documents in a book, be sure to refer to the section *Headers and Footers in a Book* in the chapter, *Large Documents in Books*.

Other choices common to all headers and footers, including page layout and font, must also be made for page numbers.

Determining Page Numbering and Style of Page Numbers

When you have a long document, you may want to break it into sections and then reassemble the sections after you have edited and printed the shorter documents.

To maintain consecutive page numbering under these circumstances, you may need to adjust the page number for each section of the large document. You can do this in the *Starting Page #* box on the Page property sheet.

You may enter any positive arabic numeral including *0* in the *Starting Page #* box. Using *0* and turning off the page number display for the first page is especially convenient when you want an unnumbered cover page as the first page of a document. Then the second page of the document will be labeled *page 1*.

Starting Page # is linked to both *Page # Style* and to the page number token that is used in header and footer frames. Any header/footer frame in which you type ESC # will display the number of the page on which it appears. If you were to change the *Starting Page #* on the Page property sheet (Figure 12-5) from *1* to *8*, for example, and the page number token were to be included in the footer frame, the numeral 1 in the footer would be replaced by the appropriate Arabic numeral 8. Then, if you changed the *Page # Style* from *Arabic* to *Lowercase Roman*, the first numbered page would have –*viii*– as its center footer.

Figure 12-5. Starting Page # variables on the Page property sheet

Page Layout

Single-Sided Layout

The default *Page Layout* on the Document property sheet is **Single Sided**. If your final copy is to be printed on just one side of the paper, you will probably choose **Single Sided** because it puts the headers and footers in the same place on every page (Figure 12-6a). This makes finding the page number or heading you want easier than it would be if the headers and footers alternated from right to left.
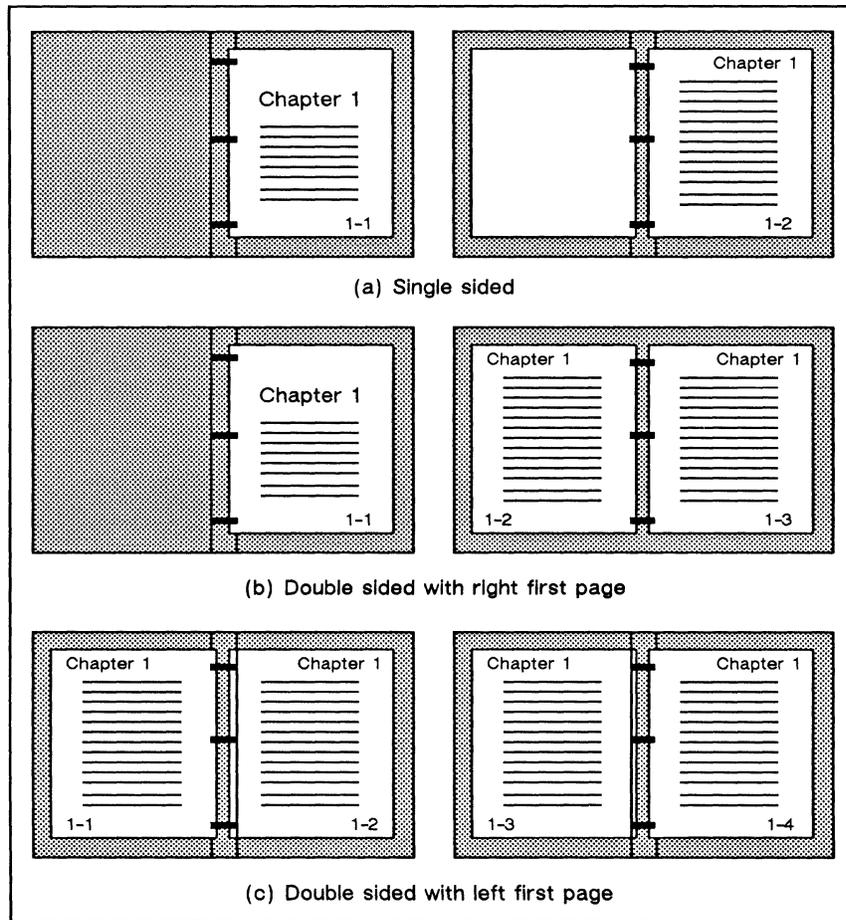
Figure 12-6. Page layout for printing

Double-Sided Layout

You can create documents in a double-sided format by turning on either **Right 1st Page** or **Left 1st Page**, where "first page" refers to the odd numbered pages. Turn on **Right 1st Page** if you want to have all your odd pages on the right; turn on **Left 1st Page** if you want to have all your odd pages on the left. (You will almost always want your first page to be on the right, since this is the convention for books in English.) This feature ensures that no matter how many pages each individual document has, the first page of the next document in the book will appear on the correct side of the page.

When you use page number and document name tokens in a double-sided document, remember to insert the tokens on one odd page and on one even page (commonly on the first and second pages) of your document.

When you change to a double-sided format from single-sided, the headers on even and odd numbered pages are automatically reflected; that is, objects on the left side of the page on a left-hand page move to the right side on a right-hand page, and vice versa. Figure 12-6b shows what happens to the header and footer when the layout is changed from **Single Sided** to **Right 1st Page**. Note that the

objects themselves maintain their internal orientation (i.e., the words read correctly).

You can see how this works in a finished document by looking at where the headers and footers on the page you are reading appear and then looking at where the headers and footers on the facing page appear. This chapter was printed using the **Right 1st Page** option. Figure 12-6c shows an example of facing pages when **Left 1st Page** is turned on.

Diff 1st Page

The default for *Diff 1st Header* and *Diff 1st Footer* is **No**. When you change from **No** to **Yes** on either of these variables, you will get empty text anchors in the header and/or footer on the first page of the document while headers and/or footers on other pages will be unaffected. You are free to insert information of your own choosing.

Diff 1st Header and *Diff 1st Footer* are independent. You can have a unique first-page header, for instance, while maintaining the document-wide footer on the first page.

Note that if you change from double-sided to single-sided, or if you change from **Yes** to **No** in the *Diff 1st Header* and/or *Diff 1st Footer,* the software maintains the discarded information until you save the document. If you return to the previous layout before you have saved the document, the software restores the previous headers and footers.

Content Changes in Different Layouts

You can edit a header/footer on any page of a document and the change will be picked up in all related frames.

If you have a single-sided layout without any first-page differentiation, header/footer changes you make on any page of the document will appear on all pages of the document.

If you have a double-sided document without any first-page differentiation, header/footer changes made to any "left-hand" page will appear on all other "left-hand" pages. Header/footer changes made to any "right-hand" page will be appear on all other "right-hand" pages.

If you have a document with any first-page differentiation, and have selected **Diff 1st Header** or **Diff 1st Footer,** changes you make on any page other than the first will appear on all appropriate pages of the document (according to the single- or double-sided rules), except the first page. Changes you make on the first page will affect that page only.

Variations on The Basic Header/Footer Creation Procedure

The cases described below assume that you have already followed the basic procedure for creating simple single-sided layout. They can be combined to achieve more complex layouts.

Double-Sided with Reflected Headers & Footers

> ↙ Change *Page Layout* to **Right 1st Page** or **Left 1st Page** on the Page property sheet.
>
> *By entering the header/footer content while still in single-sided layout and then changing the layout, you take advantage of the automatic reflection feature.*

Double-Sided, Different Content in Left and Right Page Headers & Footers

> ↙ Change *Page Layout* to **Right 1st Page** or **Left 1st Page** on the Page property sheet.
>
> ↙ Open the frame or frames on any alternate page (e.g., if you want to change all left hand pages, choose any left hand page). If you want to delete the content of a text anchor, select the anchor and press CTRL and **d**, and then type the text you want to appear there.
>
> *You must have at least two pages in a document before you can create all headers and footers for this layout.*

Blank 1st Page Headers/Footers

> ↙ Change *Page Layout* to **Diff 1st Header** and/or **Diff 1st Footer** on the Page property sheet.

New Content in 1st Page Headers/Footers

> ↙ Change *Page Layout* to **Diff 1st Header** and/or **Diff 1st Footer** on the Page property sheet.
>
> ↙ Go to the first page of the document and open the frame you want to add material to. Select a text anchor if you want to use one, and type the appropriate text. Repeat for any additional text anchors and the other frame, if needed.
>
> *You must have at least two pages in a document before you can create all headers and footers for this layout.*

Adjusting Header and Footer Frame Size

You can control the size of your headers and footers by altering the size of your page margins or by using the *Bleed* variable on the document Page property sheet.

Frame Width Bleed

Although when created they have the left and right margins of the page, header/footer frames can be made to **bleed** (extend) into the horizontal margins. (Remember, however, that it is best to keep your page margins at least .25 inches because most printers will not print copy that is within .25 inches of the page edges.)

Figure 12-7 shows the same header/footer frames with **Yes** and **No** settings for the *Bleed* variable. Notice the change in both the frame size and the position of the text anchors.

(a) No Bleed   (b) Bleed
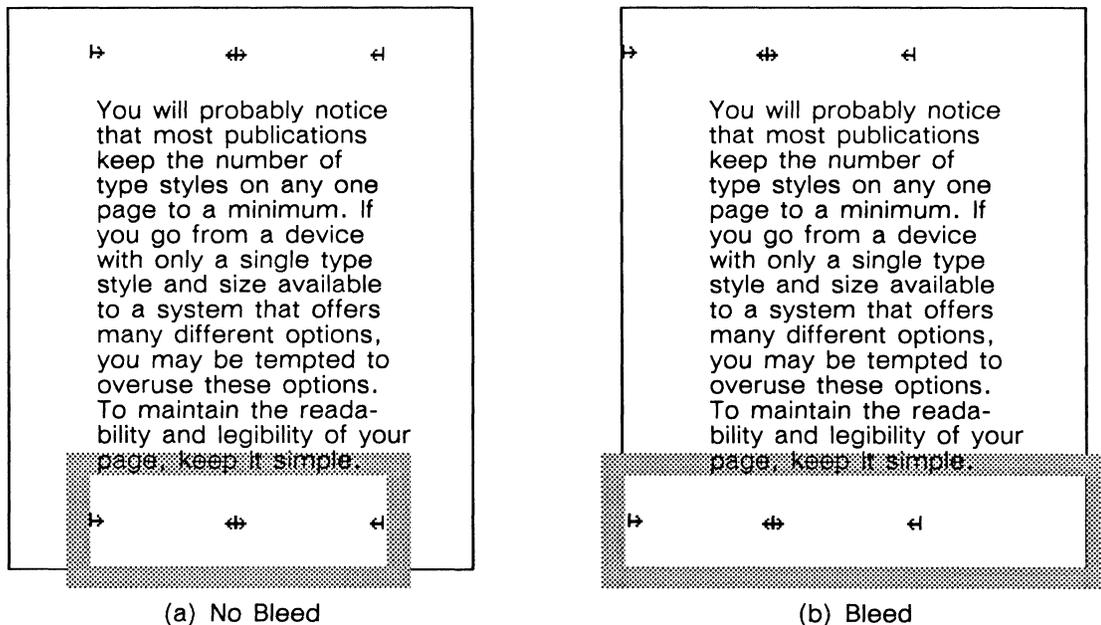
Figure 12-7. Margin Bleed setting for header/footer frames

Adjusting Headers and Footers to New Frame Width

When you adjust page margins on the Page property sheet, the width and height of header/footers frames are automatically adjusted, but the *contents* of the header or footer maintain the same relationship to the left border of the frame. Likewise, when you select **Yes** for *Bleed,* the header/footer frame bleeds into the left/right page margins and its contents move to the left (Figure 12-7b).

If you want to center the contents in the frame, ⌷ Size → ⌷ ⌷ to Frame → ⌷ ⌷ Horizontal ⌷ on the Diagramming Object Selected menu is particularly useful for rescaling the contents after a margin change, or for creating a rule scaled to the size of the frame.

To redistribute headers/footers with multiple elements, select all of the elements before you execute the command. Also, if any of the elements are diagramming

objects, you may want to select them and lock their size (using the **Lock->Size** command on the Props submenu) before you select and size the header or footer to the frame.

| Misc → | | Align → | | to Frame | leads to a submenu of commands especially useful for positioning headers and footers in their frames. With these commands, you can align objects with the top, bottom, left, or right edge of the frame, or with its vertical and/or horizontal center.

Note: If you want to align the entire contents of a header or footer to the frame, be sure to group the contents before executing the **Align** command.

Remember that you can use the **Undo** command if any alignment or sizing operations produces results you are not satisfied with.

Chapter 13

Document Templates

A template is a document that contains a set of masters — components, frames, autonumber streams, and page properties — that you can use to produce a specific document format. Templates are stored in some accessible directory, and whenever you want to create a document in a particular format, you can make a copy of the template.

This chapter discusses the following subjects:

- System supplied templates, accessed through the Desktop Create menu.
- Creating your own templates.
- Adding your templates to the Desktop Create menu, through the *Create* cabinet.

Prerequisites

The information in this chapter builds on most of the preceding chapters. In addition to basic text processing, we assume you are familiar with autonumbering, component and frame masters, links to documents and directories, and accessing other users' desktops.

System Supplied Templates

Templates for some basic text and graphic documents are provided with the Interleaf publishing software. You can create copies of them using the Desktop Create submenu (Figure 13-1).
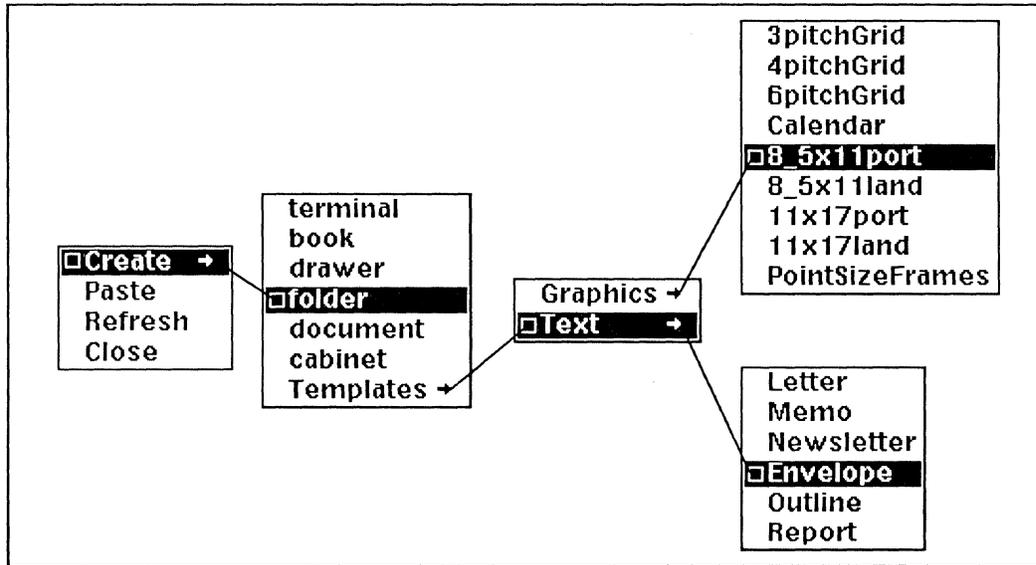
Figure 13-1. Creating a system supplied template

For example, to make a copy of the Memo template, execute the [Create ➔] [Templates ➔] [Text ➔] [Memo] command.

While the purpose of some of these templates will be obvious, others are designed to be used in particular ways. The following sections describe each template and its special properties.

Graphics Templates

The templates on the Graphics submenu contain frames.

The *3-*, *4-*, and *6pitchGrid* documents all contain frames 7.5″ by 9.5″. The frames contain ruled lines, with 3, 4 and 6 lines per inch respectively. The grid spacing in each document has been set so that there are the same number of major grid units per inch as there are lines, with one minor grid unit per major unit. You might use one of these templates as the starting point for developing a form. Figure 13-2a shows *3pitchGrid* open.
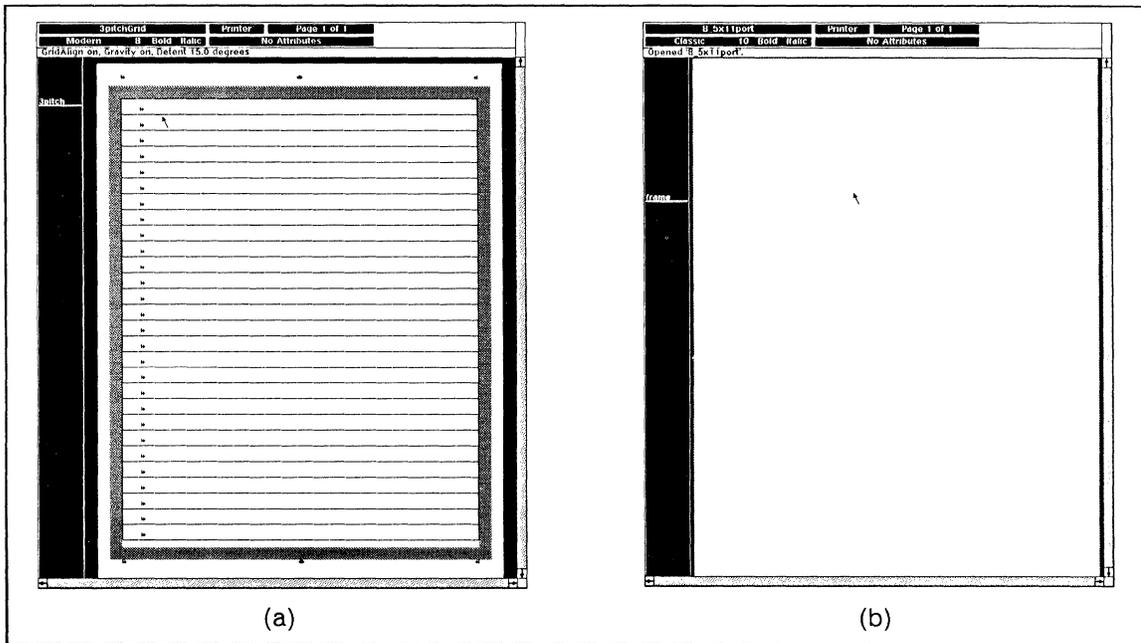
Figure 13-2. The "3pitchGrid" and "8.5x11port" templates

The *8.5x11port*, *8.5x11land*, *11x17port* and *11x17land* documents each contain a frame with the dimensions and orientation specified in the document name (*land* for landscape, *port* for portrait). The page margins of the documents are set to 0. Figure 13-2b shows *8.5x11port* open.

The Calendar Template

The calendar template allows you to easily create a calendar page for any month, in any year. Figure 13-3 shows the calendar template.
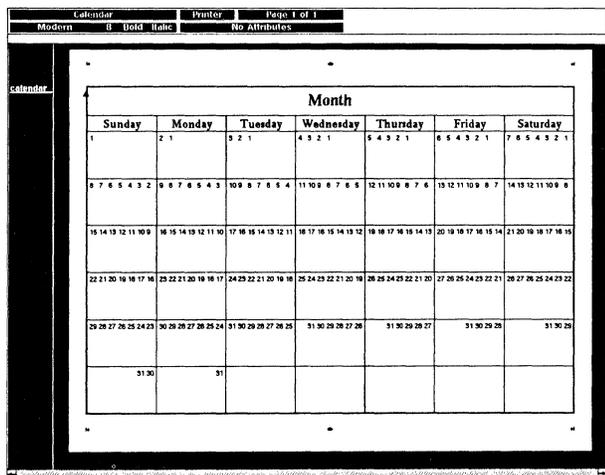
Figure 13-3. Calendar template

The numbers are grouped into seven sets. You select and cut all but the one that gives you the correct numbering, then move the numbers into the position you prefer. Each square on the calendar template is made from a separate box so that you can fill them with patterns to indicate deadlines, or vacation time. Figure 13-4 shows a calendar for the month of October, 1986.
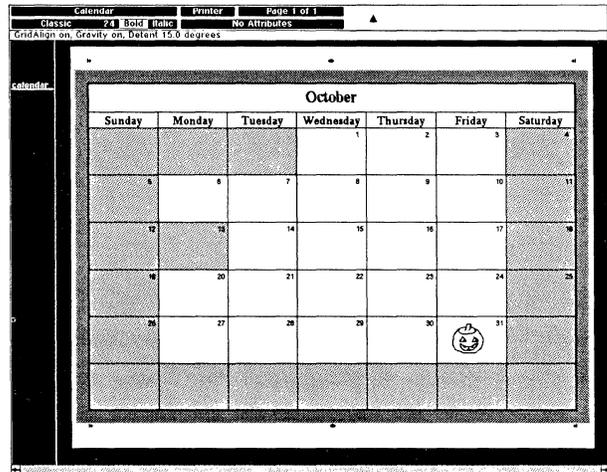
Figure 13-4. An edited copy of the calendar template

The PointSizeFrames Template

This template, unlike the others, is intended to be pasted into another document. It contains in-line frames for each point size so that you can insert graphics into lines of text without disturbing the line spacing. If you paste it into a document, masters for the in-line frames will be added to the Create Frame menu for that document.

To add in-line frame masters to a document:

✔ Execute the [Create →] [Templates →] [Graphics →]
[PointSizeFrames] command.

✔ **Cut** the document and **Paste** it anywhere in your main document.

A single component containing in-line frames for each point size will be pasted into your document. The name of each frame will be added to the Create Frame submenu (Figure 13-5).

```
┌─────────────────────────┐
│■□06pt                   │
│  08pt                   │
│  09pt                   │
│  10pt                   │
│  11pt                   │
│  12pt                   │
│  14pt                   │
│  18pt                   │
┌──────┬──────┐ 24pt      │
│Create│Frame │ 36pt      │
│ Paste│Index │ At Anchor │
│ Fonts│Auto #│ Bottom    │
│Center│ Ref. │ Following Anchor│
│ Misc →      │ Following Text  │
└──────┴──────┘ Footnote  │
              │  Top      │
              └───────────┘
```
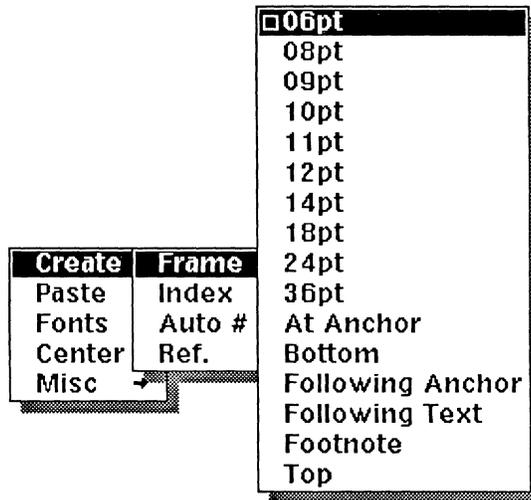
Figure 13-5. In-line frame masters on the Create Frame menu

✔ If you pasted the *PointSizeFrames* document as a component,
Cut the *ptszframe* component.

✔ Execute the **Create Frame 12pt** command to create an in-line frame in 12 point text.

If you are going to use only a few point sizes, and you do not want ten master frames added to the Create Frame submenu, use the following procedure.

To add masters for a few sizes of in-line frames:

✔ Execute the [Create →] [Templates →] [Graphics →] [PointSizeFrames] command.

✔ **Open** the *PointSizeFrames* document.

✔ Select the point sizes you want and **Cut** or **Copy** them.

✔ **Paste** the frames anywhere in your main document and **Cut** them.

The names of the frames you pasted and cut will be added to the Create Frame submenu.

Text Templates

The text templates contain the components you need to create several simple document layouts – a memo, a letter, a report, and so on.

Letter

The *Letter* template has a large header frame in which you can insert your corporate logo. It has components for the date, the address, the salutation, and so on. "Dear" and "Sincerely yours" are already entered in the appropriate components.

Memo

Like the *Letter* template, the *Memo* template has a large header frame for your company's logo. The first four components have the words, "To:", "From:", "Subject:", and "Date:" in bold, followed by roman tabs. There are also components for the body of the memo.

The Newsletter Template

The large frame in the Newsletter template contains vertical rules to help create a three-column layout with microdocuments and graphics. The rules are printing locked, so you can leave them in place and they will not appear in the hard copy.

Figure 13-6a shows the template and Figure 13-6b shows a sample newsletter.

(a)   (b)
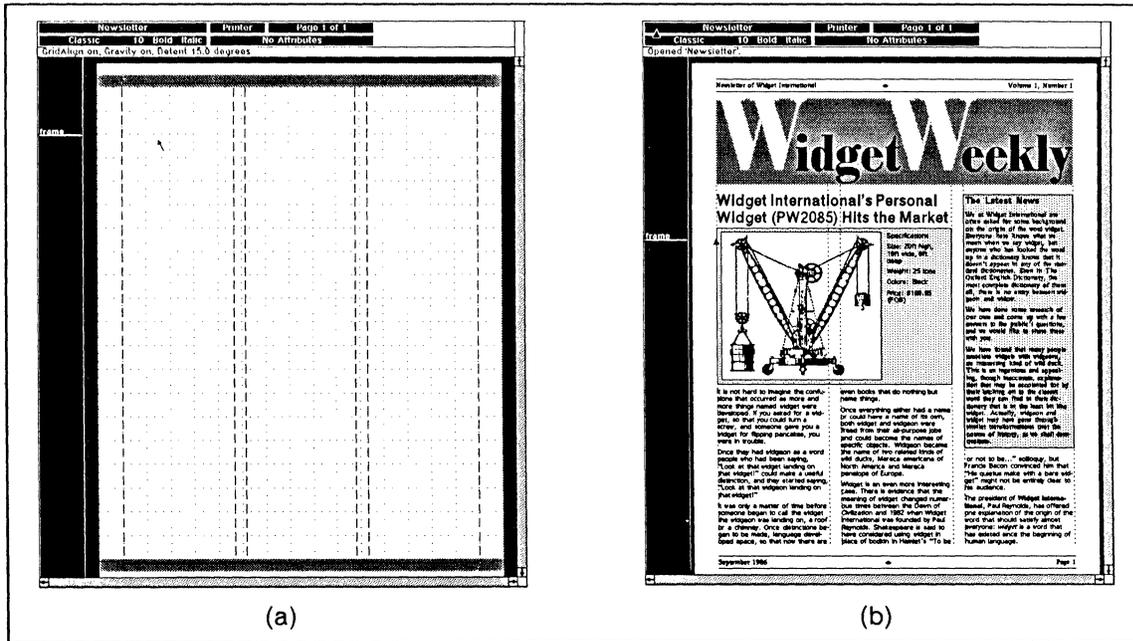
Figure 13-6. The Newsletter Template and a sample newsletter

Envelope

The *Envelope* template has *Address*, *Sender*, and *To* components, and a microdocument in the footer with the word "Attention:" entered.

Outline

The *Outline* template contains four components. The first three components are associated with the three levels of the Outline autonumber stream. The last component contains a bullet.

Report

The *Report* template contains four levels of headings and a paragraph component. The headings are *title*, *subtitle*, *section* and *subsect*.

Creating a Template

The system supplied templates are only a starting point. For most of your work, you will want to design your own layouts. You will find information about design and layout in the chapters *Page Design* and *Typography and Typefaces*.

Once you have an appropriate design, this section will help you turn it into a template for future documents.

To begin with, you will need a document that contains at least one example of each component, frame, and autonumber stream that should be in the template. The easiest way to do this is to follow this procedure:

To create an example document for a template:

✓ **Copy** the document in which you developed the design and **Open** the copy.

✓ Select all the components in the design document and **Cut** them.

✓ **Create** one example of each component that belongs in the template.

✓ **Create** one example of each frame that belongs in the template.

✓ **Create** one example of each autonumber stream that belongs in the template.

The next step is to eliminate the masters you created in the design process that you did not use and therefore will not want to include in your template.

To purge unused masters from the example document:

✔ Execute the [Misc →] [Masters →] [Purge Components] command.

✔ Execute the [Misc →] [Masters →] [Purge Frames] command.

✔ Execute the [Misc →] [Masters →] [Purge Streams] command.

All component, frame, and stream masters without an instance will be purged.

If you are creating a simple template for short documents in which all the components will always be used, you will probably want to leave the example components in the template — particularly if there is text always associated with the component.

If you are creating a template for a more complicated document, or one in which not all components are always used, the example components may get in the way, and you will probably want to cut them. You may want to have another document that contains the example components, to use as a sample.

Using Templates

You can either create a copy of the template first and then enter the contents of the document, or create the contents and paste them into a copy of the template.

It saves time at the end of the production cycle if your documents were created in a template, and do not have to be formatted. However, there are several reasons why it may not be practical to design the template before the document has been written.

First, it is easier to design a template if you have a sample of the text you will be formatting to work with. Second, you may have word processing equipment that you want to continue using for text entry. And finally, even if your writers are working in the publishing software, it may not be possible to complete the design before they must begin writing.

These factors result in a design cycle: make a preliminary design, write a little, revise the design, reformat the document. To keep this cycle in hand, consider the following:

- Decide early what the names of widely used components, autonumber streams and frames will be. This will save time in reformatting the documents.
- Institute procedures for creating new masters and modifying existing ones. You will almost certainly find that you need a new heading, for example, during the writing process.

- Always paste text or components into the template. The page properties, the headers and footers, and the master components, frames, and autonumber streams will be taken from the document you paste into.

Using the Create Cabinet to Share Templates

To make an often used template readily available to you or to all users, you will use the *Create* cabinet.

The *Create* cabinet determines the commands that appear on your Desktop Create submenu. Documents and empty directory icons in the *Create* cabinet are represented as commands. Directory icons that have contents are shown as submenus; their contents are entries on the submenu. By changing the contents of the *Create* cabinet, you can change the commands, and even their positions, on the menu.
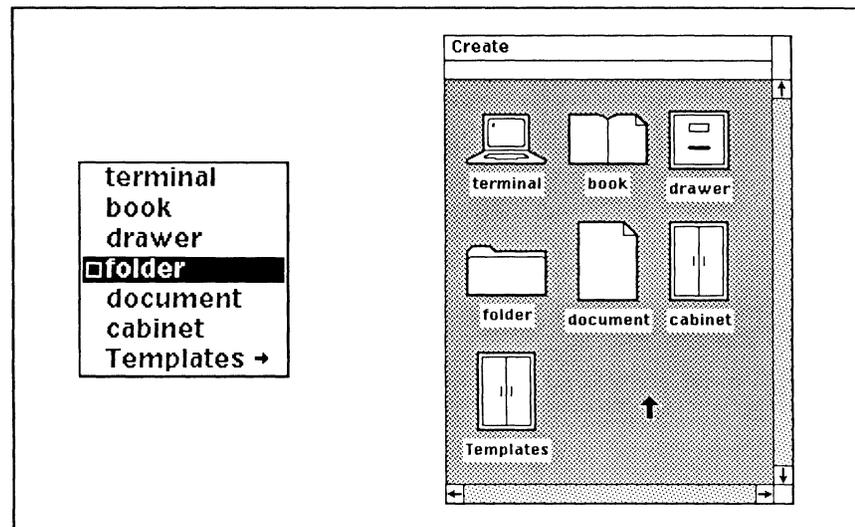
Figure 13-7. The Create submenu and the Create cabinet

The Master Create Cabinet

The publishing software's master copy of the *Create* cabinet is in the *System* cabinet on your desktop. If you explore this cabinet, you will find all the system supplied templates described in the first section of this chapter, in addition to the standard desktop icons — document, folder, drawer, etc.

Although the System Administrator can change the contents of the master copy of the *Create* cabinet, it isn't a good idea since the cabinet will be overwritten by future releases of the publishing software.

Instead, you can make a *Create* cabinet on your desktop, and customize it by adding your own templates, links to objects in the *System* cabinet, and links to objects on other users' desktops.

Customizing Your Create Cabinet

The master *Create* cabinet in the *System* cabinet controls the contents of the Create submenu *unless* you have a cabinet named *Create* on your desktop. To customize the Create submenu, you need only build a *Create* cabinet that contains the icons and templates you want. Then you will be able to create a copy of a template by executing a single command.

If you have more than one cabinet named *Create* on your desktop, the software will choose the cabinet that is closer to the upper left corner, moving from left to right and top to bottom.

The *Create* cabinet can be simple, containing only real icons, or it can be complex, with links to documents and folders on other users' desktops. The next sections describe four scenarios, and four *Create* cabinets that might be appropriate.

Adding Your Own Templates to the Create Cabinet

If you are the only user of the publishing software, or if your work is independent of other users at your company, you may want a simple *Create* cabinet, containing only the basic icons and your own templates. The following procedure will help you create such a cabinet.

To add your own templates to the Create Cabinet:

> ✓ **Create** a cabinet on your desktop, change its name to *Create*, and open it.
>
> ✓ **Copy** or **Cut** your own templates and paste them into your new *Create* cabinet.
>
> ✓ Open the *Create* cabinet in the *System* cabinet.
>
> *As soon as you have a Create cabinet on your desktop, it controls the popup menu. To get the basic icons, therefore, you will need to copy them from the System cabinet.*
>
> ✓ Select the objects you want to be able to create through the Desktop Create menu and execute the **Copy** command.
>
> ✓ **Paste** the icons into the *Create* cabinet on your desktop.
>
> *The Desktop Create submenu will be updated immediately to reflect the new objects you placed in the cabinet.*

Sharing a Create Cabinet with Your Department

If everyone in your department uses the same set of templates for their work, you may want to build a customized *Create* cabinet and have all users make a link to it.

To share a Create cabinet with your department:

✓ Make a customized *Create* cabinet containing the department's templates and the standard icons.

✓ Move the cabinet into the correct directory.

If you have a Desktops cabinet, you might want to put the Create cabinet on the documentation designer's desktop. Otherwise, you can put it in the directory that holds the Bulletin Board.

✓ On each user's desktop, make a link to the shared *Create* cabinet and **Paste** it.

A shared *Create* cabinet like this is very easy to maintain. Whenever you change the departmental *Create* cabinet, whether you have added a new template or modified an existing one, the changes are automatically reflected in the other users' *Create* cabinets. However, it is not very flexible. All users have the same templates in their *Create* cabinet.

Linking to Department or Company Cabinets

A more flexible solution may be to build customized cabinets for department or company templates. Then each user can make a link to those cabinets within his or her *Create* cabinet.

To include departmental templates in your Create cabinet:

✓ **Create** a cabinet, name it *Department* and open it.

✓ **Paste** the departmental templates into the *Department* cabinet.

✓ Move the cabinet to the appropriate directory.

The appropriate directory might be the System Administrator's desktop, if users at your site have a Desktops cabinet. Otherwise, it can be placed in the same directory that holds the Bulletin Board.

✓ Make a customized *Create* cabinet on your desktop, containing the standard icons and your own templates.

✓ **Copy** a **Link to** the *Department* cabinet.

If you place the Department cabinet in the directory with the Bulletin Board, you may need to create the link through the operating system. Ask your system administrator to set it up for you.

✓ **Paste** the link in your *Create* cabinet.

This approach requires a little more administration: if you change the name or the location of the cabinet that contains the department's or the company's templates, all users will need to update their link. However, the templates themselves

will be updated automatically for the users, and the users will be able to put templates they alone need on their own Create submenus.

Linking to Individual Documents

If you use only four of your company's 250 accounting forms, it may seem easier to create links directly to the documents in the *Accounting* cabinet. In most cases we recommend against this, because it is so difficult to maintain the links. If the name of one of the forms changes, your link will not be updated and you will not be able to create that form until you have remade the link.

If you do use links to specific documents, you will need to make arrangements with the person maintaining the real template to notify you if she changes it, so you can update your link.

A Sample Create Menu and Cabinet

Figure 13-8 shows an example of a Desktop Create menu that was built using icons, links to departmental and corporate cabinets, and links to individual documents.
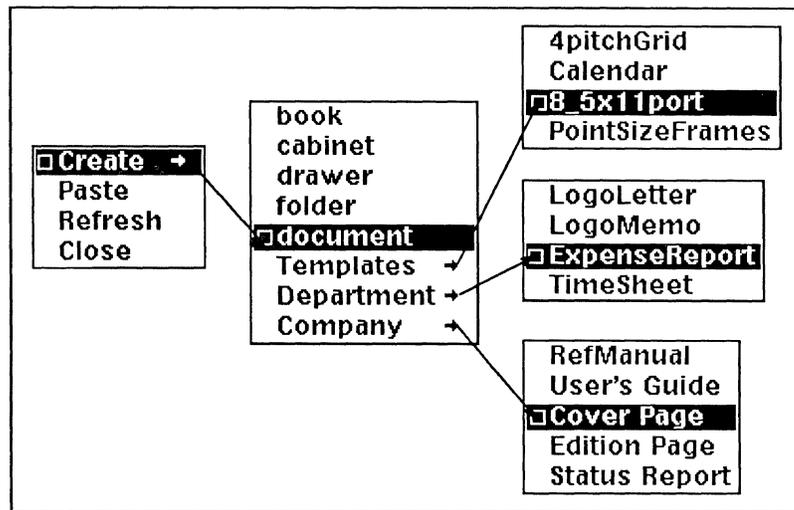
Figure 13-8. Customized Create submenu

The *Create* cabinet on which the menus in Figure 13-8 are based was created in the following way:

- The *Create* cabinet itself is not a link; it is a true icon.
- The book, cabinet, drawer, folder and document icons were copied from the master *Create* cabinet in the *System* cabinet.
- The *Templates* cabinet is not a link; it is a true icon. It contains links to four documents in the *Graphics* folder in the master *Templates* cabinet in the master *Create* cabinet.

- The *Department* cabinet is a link to a cabinet on the documentation designer's desktop.
- The *Company* cabinet is a link to a cabinet maintained by the System Administrator for the whole company.

Changing the Order of Commands

The order of the commands on the Create submenu is determined by the position of the corresponding icon in the *Create* cabinet (left to right, top to bottom). If the icon is in the upper left corner of the *Create* cabinet, the command will be at the top of the Create submenu.

It is easier if the template you use most often is the default, which is in the middle of the menu. The next most often used templates should be on either side of the default, and finally, the ones you use least should be at the top or bottom of the menu.

The Samples Drawer

The *System* cabinet on your desktop also contains the *Samples* drawer. In the *Samples* drawer, you will find examples of some of the ways you can use the Interleaf publishing software. Among them are examples of most templates. You will find it helpful to compare the templates in the *Create* cabinet with the samples created from them.

Chapter 14

Document Management

This chapter deals with document management decisions you can make inside an open document window. Here, we tell you how to

- save documents in different formats;
- view and save different files represented by one icon; and
- mark revisions in a document.

The tools for making these decisions are the pulldown menus available in the Name box and the Attributes box of a document header (Figure 14-1).
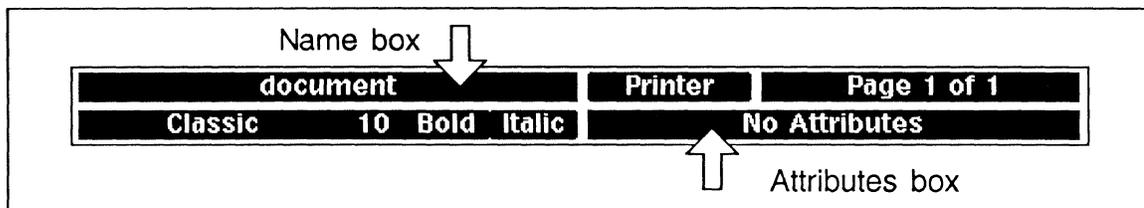
Figure 14-1. Document header

The *Name* pulldown menu offers you control over the way your document is saved, the version of it you view or edit, and the creation of new, renamed copies of an open document.

The *Attributes* pulldown menu gives you access to a series of text-marking attributes — revision bars, strikethrough, and underlining — that you can use to indicate changes between one version of a document and the next.

The three chapters following this one cover other document management issues.

Desktop Management explains how you can use directories to contain and organize your documents, and how access to those documents is controlled.

Large Documents in Books concentrates on the *book* directory, which you can use to break large documents into smaller sections while maintaining page and other numbering relationships among these smaller documents.

Indexes and Tables of Contents describes the automatic index and table of contents creation features of the publishing software.

Saving Documents

Document Format

On the Document Name pulldown menu, there is a Save submenu (Figure 14-2).
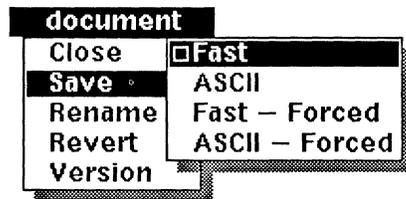
Figure 14-2. Save submenu of Document Name pulldown

There are four choices on the Save submenu, and they are paired.

When you use **Fast** or **Fast — Forced**, you get the same kind of file. Either command saves your document in a format that can be interpreted only by the publishing software.

Fast is the default you get when you release the middle mouse button on **Save** without sliding onto the submenu and choosing another option. It is also the type you get when you save a document using a stickup menu.

When you use **ASCII** or **ASCII — Forced**, you get another kind of file format that you can read using either the Interleaf publishing software or some other editor. You would most often use an **ASCII** save when you want to communicate your document to other text editors.

Some information will be lost when you save ASCII. For example, discretionary hyphens and the position of the cursors are lost in an ASCII save. Certain optional fonts are unreadable in ASCII form. If a frame is open when you **Save ASCII**, it will be closed when you reopen the document. If you do not want to use another editor with a document, you should always use one of the Save → | | Fast | commands.

As the name implies, a **Fast** save takes less time than an **ASCII** save. When a document has been saved **Fast**, it also opens more quickly than it does when it has been saved **ASCII**.

While a document is being saved, messages appear in the status line telling you what is currently being saved. If you are saving an ASCII-format version of the document, a message also tells you the ASCII version is being saved.

Forced Save Commands

If you execute the **Fast** or **ASCII** Save commands but have not made any changes in a document, the status line displays the message: *You have not modified* 'documentname' *since it was saved* and cancels the save. In most cases, this is a reasonable time-saving feature. However, we provide the **Forced** options in case you have a special reason to save an unaltered document.

One reason for executing a forced save is to save a new text caret position and thereby have a document open on that page. The **Forced** commands are also useful for integrating newly pasted documents into a book (see the chapter *Large Documents in Books*). You do not have to use the **Forced** commands to save an ASCII document as a Fast document, or vice versa.

Copying (Renaming) Open Documents

Usually, when you want a copy of a document, you will use the **Copy** command on the Desktop popup menu. However, there are circumstances when you will need to make a copy of an open document. The following are two examples of such circumstances:

- You are working in a document for which you do not have write permission, and you want to save your work. In this case, if you try to save the document, you will see the Lack of Permission stickup menu.

- You have done some work in a document, but you think you may not be using the best version of the document. You want to **Revert** to the last saved version of the document, but you also want to be able to save the work you have done.

To rename a document:

↙ Select **Rename** on the Document Name pulldown menu.

▦ **Type** the new name on the Rename stickup menu (Figure 14-3).


```
┌─────────────────────────────────────────────┐
│        Please type the new document name.    │
│  ████████████newname▂████████████████████    │
│              ┌─────┐┌──────┐                 │
│              │Enter││Cancel│                 │
│              └─────┘└──────┘                 │
│                  ↑                           │
└─────────────────────────────────────────────┘
```
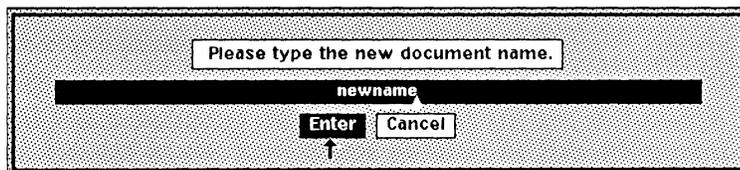
Figure 14-3. Rename stickup menu

↙ Select **Enter**.

or ▦ Press the RETURN key.

If you do not type a new name for the document, and simply select **Enter** on the stickup, the software will use the original document name and add the extension *.new*, so *pingo* would become *pingo.new* on the desktop.

You should avoid using **Rename** inside a book directory unless you understand its possible effects on page numbers, autonumbers, and references. See the chapter, *Large Documents in Books* for details.

Saving Renamed Documents

When you rename a document, the original document is closed automatically and a new icon is created to represent the new document. However, in a sense, this new document does not yet exist: **Cancel** is the only command on the new document's Revert submenu. You must save the document in order to give the icon contents.

Always save a renamed document immediately. You cannot close your desktop if you have any unsaved, renamed documents open; if you try to close a renamed document without saving it, you will see the stickup menu in Figure 14-4.
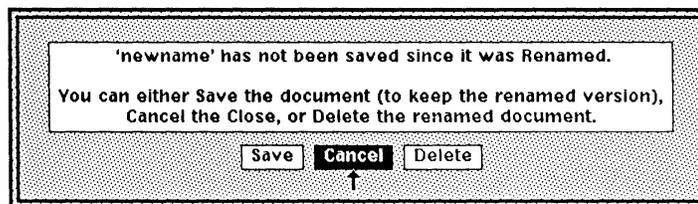
Figure 14-4. Close stickup for an unsaved, renamed document

If you have not saved a renamed document and select **Delete** on the Close Renamed Document stickup, you will lose any changes you have made in the renamed document. There is no way to retrieve this information.

Although you may not have been the owner of the original document, you almost always own a renamed document. The renamed document will have the same permissions as a newly created document on your desktop. (For a discussion of ownership and permissions, see the chapter *Desktop Management*.)

Document Versions

The Interleaf publishing software has four mechanisms for creating document files. *Document files* and *crash files* are created in response to actions by the user. *Backup files* and *checkpoint files* are created automatically by the software. All of these files can exist simultaneously for the same document.

On your desktop, one icon represents all existing versions of a document. In general, when you open a document you see the *document file*. However, all of the versions of a document are accessible to you through the **Revert** command on the Document Name pulldown menu.

The Document and Backup Files

When you create a new document on your desktop, the software creates a corresponding *document file*. Initially, this *document file* consists only of some structural information, such as the dimensions and margins of the page and the font of the default component.

When you save the document for the first time, the initial document file becomes the *backup file,* and the document you have just saved becomes the *document file*.

As you work on the document and save it, this process continues. If the Revert submenu shows that *document* and *backup files* exist, you know that the *document file* is always the copy of the most recent save you have made, and the *backup file* is always the copy of the second most recent save you have made.

The Checkpoint File

As you work in a document, a *checkpoint file* is created automatically by the software whenever you have typed a certain number of keystrokes. For text entry the number of keystrokes is between 3000 for short documents and 7000 for long documents. When you are working in diagramming, the mouse movements are also figured into this keystroke total.

The software recognizes a difference between straight text entry and editing, and it saves checkpoint versions of a document more often when you are editing than it does when you are entering text.

If you have not provided any input from the mouse or the keyboard for one hour, the software creates checkpoint files for every document open on the desktop. After the files are created, a screensaver pattern consisting of moving Interleaf logos will display on your screen. The desktop reappears in exactly the state you left it as soon as you use the keyboard or mouse again.

The *checkpoint file* safeguards against loss of work if something like a power failure should occur while you are editing a document. However, the frequency of checkpoint document creation is a compromise between providing maximum protection of your work and avoiding excessive interruption of your editing and

writing. Therefore, you should save your documents regularly, particularly when you are doing something that would be time-consuming or difficult to reconstruct.

If there is a *checkpoint file*, it is a more recent version of your document than the *document* or *backup files*.

When you save a document in the normal way, the *checkpoint file* is deleted because it is superseded by a more recent *document file*. The *checkpoint file* is not deleted when you create a *crash file* because there is a chance that saving a document in a *crash file* will produce a version that is damaged. If this were to happen, the *checkpoint file* would be the best version to use as a starting place for rebuilding the document.

The Crash File

There are two rare circumstances in which you are given the opportunity to save documents in *crash files:* if your computer crashes or if it runs out of memory. In either case, you will not be able to work in documents on your desktop until you have exited from the publishing software and then re-entered it.

In case of a crash, the Interrupt stickup menu appears automatically. On this stickup, there is a **File** option that gives you a way of saving documents. If you want to save the changes in any documents open on your desktop, be sure to select **File** before **Exit**. **Exit** takes you out of the publishing software immediately, without providing an opportunity to save your documents.

If you have more than one document open when you select **File**, you will be asked if you want to save a *crash file* for each document you have changed.

If your computer runs out of memory, a stickup menu will explain what is happening, and you will be given the opportunity to save your documents before the computer closes down.

Stickups when Opening a Document

Normally, when you open a document, the *document file* opens automatically. There are two exceptions to this rule:

- if something is wrong with the *document file* and it cannot be opened
- if there is a *crash file* of the document

If a *document file* stored in Fast form is damaged in some way, it cannot be opened. For example, if there is a power failure while the software is saving a document before closing it, the *document file* may be unreadable. If so, when you try to open it, a message will be displayed in the status line telling you what is wrong with the file. If there are any other versions of the document, you will also see a stickup menu listing the possible actions (Figure 14-5).
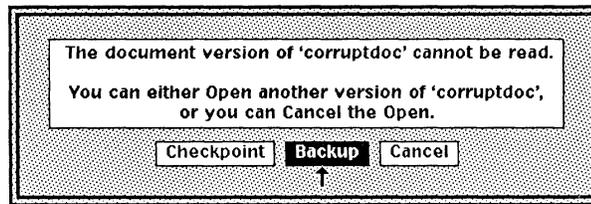
Figure 14-5. Stickup menu for damaged document

ASCII files generally open even when damaged.

When you open a document that has a *crash file* associated with it, a stickup menu will give you a choice of which version of the document to open. Figure 14-6 shows the most complete version of this stickup menu.
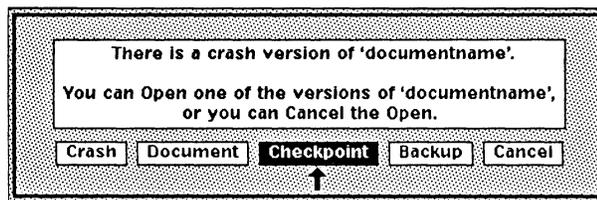
Figure 14-6. Crash version stickup menu

If there is a *crash file,* it is almost always the most recent version of the document and the version you want to use. There is one, very rare exception. If you open a crash file of a document and you do enough editing so that a checkpoint file is created, and then you close without saving the document, the checkpoint file will be more recent than the crash file.

When you open a crash file version of a document, and then you save the document in the normal way, the version labeled *crash file* is deleted. (It will, of course, become the *document file* if it is the version you save.)

If there is a *crash file,* it is almost invariably the most up-to-date version of the document.

If there is a *checkpoint file,* it is more up-to-date than the *document file.*

The *document file* is more up-to-date than the *backup file.*

The Revert Submenu

The Revert submenu grows or shrinks depending on which versions of a document exist at a particular time. There are several different sequences in which these changes may occur. Figure 14-7 illustrates one sequence.

When you open a newly created document, the **Revert** submenu looks like this:

```
document
Close
Save      →
Rename   □Cancel
Revert    to Document
Version
```
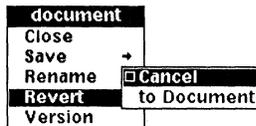
If you work for a while and then save your document, the **Revert** submenu will look like this:

```
document
Close
Save      →
Rename   □Cancel
Revert    to Document
Version   to Backup
```
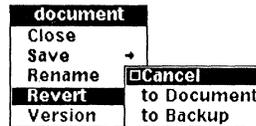
If you enter many characters and the software checkpoints the document, the **Revert** submenu will look like this:

```
document
Close
Save     □Cancel
Rename    to Document
Revert    to Checkpoint
Version   to Backup
```
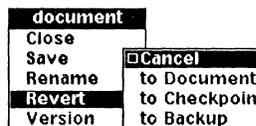
If your computer crashes and you file your document, the **Revert** submenu will look like this:

```
document
Close    □Cancel
Save      to Crash
Rename    to Document
Revert    to Checkpoint
Version   to Backup
```
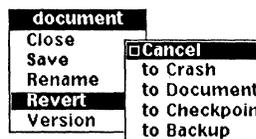
When you save your document the next time, the **Revert** submenu will look like this again because Crash and Checkpoint files are removed when you save your document:

```
document
Close
Save      →
Rename   □Cancel
Revert    to Document
Version   to Backup
```
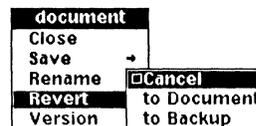
Figure 14-7. Revert submenu sequence

You can use the **Revert** command to browse through as many of the versions as you like, but it is important to remember that once you save your document, the version you save becomes the *document file*. The next time you look at the **Revert** submenu, you will see only *document* and *backup files* listed.

If you execute the **Close** command after reverting and you do not want to make the current version the document file, you can use the Close Document stickup to **Cancel** the closing or **Close** without saving — that is, without altering the versions of the document.

> If you **Save** after using **Revert,** the version of the document
> you have open when you **Save** becomes the *document file.*

Canceling a Revert Command

You can use either CTRL **g** or the Interrupt stickup menu to cancel a **Revert** com-
mand. When you cancel a **Revert** command, the document is automatically
closed.

Checking Software Version

The last command on the Name pulldown menu is **Version.** If you execute this
command, you will see a message in the status line that tells you what version of
the publishing software you are running.

Text Attributes for Revision Tracking

The publishing software provides three text-marking devices that you may find useful for indicating changes you make between one draft of a document and the next. These devices, which we call **Attributes,** can be used with every font, including **Symbols** and **Math** fonts. The Attributes include the following:

- You can use **Strikethrough** to indicate any text that you want deleted
- You may use the **Underline** feature for any purpose you like. One possible use is to indicate added text.
- You can use **Revision Bars** to mark lines you have changed.

You can turn one or more of these attributes on at the same time. The Attributes box shows which, if any, of the attributes are on, at the text caret position. Figure 14-8 shows how the Attributes box looks when all three attributes are on simultaneously.
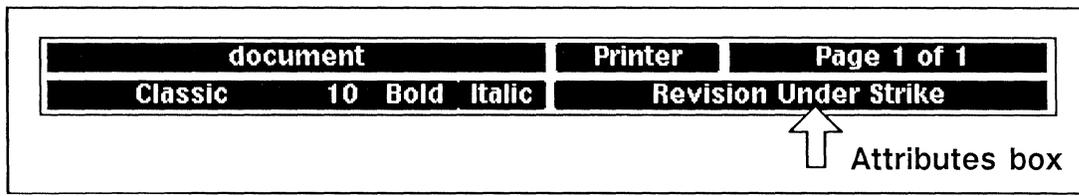
Figure 14-8. Attributes box

Access to Attributes

The interface for turning attributes on and off is essentially the same as that for turning fonts on and off.

You can use the `Fonts →` `Attributes →` commands on the Text Selected popup menu to turn an attribute on or off for selected lines of text. For Strikethrough and Underline only, you also have a **Toggle** command on this menu that allows you to turn an attribute on where it is off and simultaneously turn it off where it is on. Figure 14-9 shows the submenu of choices for underlining selected text.
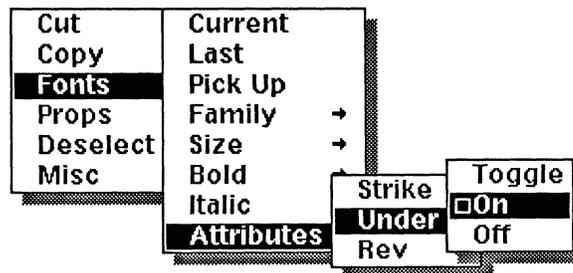
Figure 14-9. Attributes submenu for selected text

With the cursor in text but with no text selected you can use one of the `Fonts →` `Attributes →` commands on the Text Location popup menu to turn an attribute on, at the editing caret. Any new text you enter there will be marked with that attribute. Figure 14-10 shows one of the Attributes submenus for a text location.

Create	Last	
Paste	Pick Up	
Fonts	Family	→
Center	Size	→
Misc	Bold	
	Italic	
	Attributes	

Strike	Off
Under	□On
Rev	→
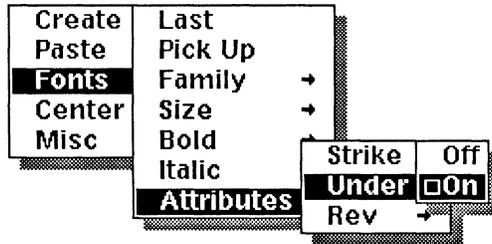
Figure 14-10. Attributes submenu for text location

You can use the Attributes pulldown menu to turn on an attribute at the editing caret and to mark new text you enter there with that attribute. If text is selected when you execute this command, it will be marked with the attribute you choose. Figure 14-11 shows the Attributes pulldown menu.

No Attributes	
Strike	Off
Under	□On
Rev	→
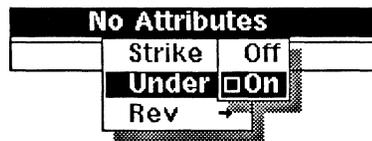
Figure 14-11. Attributes pulldown with Underline submenu

You can turn an attribute on or off for an entire component using one of the *Attributes* boxes in the Component Format property sheet (Figure 14-12). If you turn the property *on* and any characters in the component already have that attribute on, those characters will lose the attribute.

Component Properties: **Format** | Tab | Page | Custom

Name	para	
Margins		
Top	0	inches
Bottom	0.10	inches
Left	0	inches
Right	−0.10	inches
First Indent	0	inches
Line Spacing	1.15	lines
Alignment	**Flush Left** | Centered | Flush Right
Font	Classic | 12 | Bold | Italic
Attributes	Under | Strike | Revision
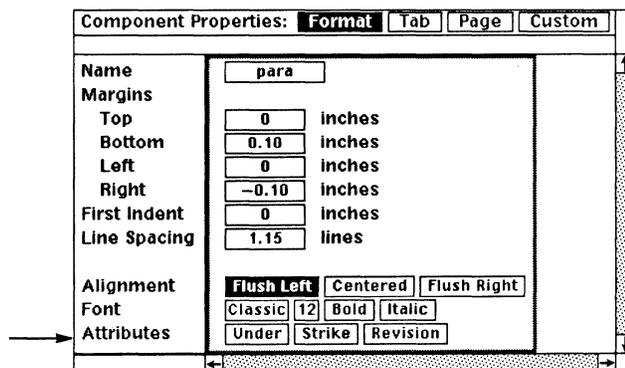
Figure 14-12. Attributes line, Component Format property sheet

Strikethrough and Underline

The weight of the underline is suitable to the size of the selected font. Figure 14-13 shows how it changes as the font size changes.

> This is 10-point type.
> ## This is 18-point type
> # This is 24-point type.
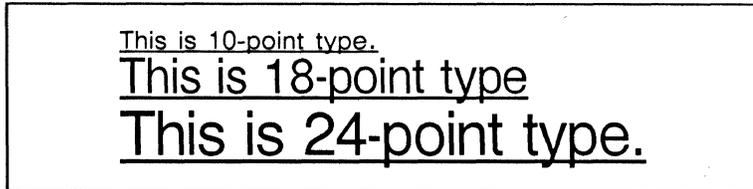
Figure 14-13. Underline weight for different fonts

When you choose [Under →] [On] on the Text Location Fonts submenu, everything you type including spaces is underlined. When you choose [Under →] [On] on the Text Selected Fonts menu, everything in the selected region is underlined. The text looks like this: underlining is *on* for this text.

Toggle is the default on the Fonts submenu of the Text Selected menu. If the text is underlined, **Toggle** removes the underlining; if the text is not underlined, **Toggle** underlines it:

> This text with some underlining *becomes* This text with some underlining.

The strikethrough commands work in exactly the same way as the underline commands.

You can also toggle strikethrough and underline using functions keys. See the chapter *The Keyboard* for the specific keys that accomplish these functions on your workstation.

Revision Bars

You can create **revision bars** to mark changes you make in a document. The revision bar is a vertical black line 3/32 of an inch wide, displayed in the margin next to a line of text, as in Figure 14-14.

> This document shall be produced using 10-point Classic type. It shall be 8.5 inches wide and 11 inches deep, and shall have top and bottom margins of 1 inch, a left margin of 1.1 inches, and a right margin of 1.5 inches.
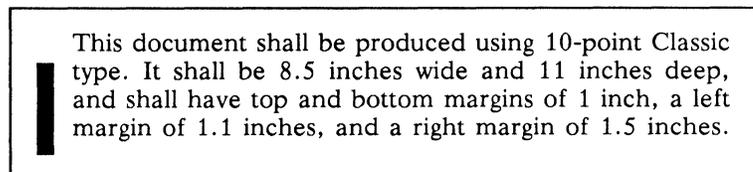
Figure 14-14. Revision bar (actual size)

You have complete control over the creation of revision bars — they do not automatically appear every time you make a change.

How to Turn On Revision Bars

Revision bars are turned on and off exactly like underline, strikethrough, and other text attributes. The major difference between revision bars and other attributes is that the bars are displayed on a line-by-line, rather than character-by-character, basis. If any character in a line has the *Revision* attribute turned on, a revision bar is displayed next to that line.

To turn revision bars off, select all the text in that line (or lines) and execute the **Off** command.

If you highlight a line and execute [Rev →] [Off] but the bar remains on, you may have to select the soft space at the end of the line by placing the caret at the end of the line and moving the mouse cursor to the beginning of the next line with the right button held down. Then execute the command again.

Where Revision Bars are Displayed

No matter how component margins in a document vary, all revision bars for a column (except bars in frames) will line up vertically. If the last line of one component and the first line of the next component both have revision bars on, the bar will display across the component margins.

If you want a revision bar to display in the margin next to a frame, use an At Anchor frame and put it in a component by itself. You can then turn the revision bar on using the Component property sheet or by selecting the frame and using one of the **Revision** commands for selected text.

By default, revision bars are displayed in the left margin of a document in single-column documents and to the left of each column in multicolumn documents. You can also change the *Revision Bar Placement* setting on the Custom sheet of the Page property sheet to *Automatic*. With this property on *Automatic*, you will see the following results:

- Single-sided, single-column layouts continue to have revision bars on the left.
- Double-sided, single-column layouts have revision bars on the outside of the page (i.e., on the right for the right-hand *page* and on the left for the left-hand *page*).
- Double-sided, two-column layouts have revision bars on the *outside* of each column (i.e., on the right for the right *column* and on the left for the left *column*).
- Double-sided layouts with more than two columns have revision bars on the left of all but the rightmost column.

Revision bars will not cross the halfway point of a gutter between columns. If there is not enough room to display them properly, they will overlap the text they are associated with.

Revision Bars in Microdocuments

You can turn revision bars on for microdocuments inside frames, including header/footer frames. These revision bars behave like any others, except that they do not necessarily line up with the bars outside of frames and they may overlay text. You can eliminate this overlap by increasing the page margins of the microdocument (not the main document).

Printing Attributes

You can use any of the attributes strictly as an on-screen aid or you can choose to have them print as well. You control attribute printing on the Printer property sheet (Figure 14-15).

```
Printer Properties

Header Page          Yes No
Double Sided         Yes No
Manual Sheet Feed    Yes No
Print Rev Bars       Yes No
Print Strikes        Yes No
Print Underlines     Yes No
Final Output Device  cx
Default Printer      nearest-cx
```
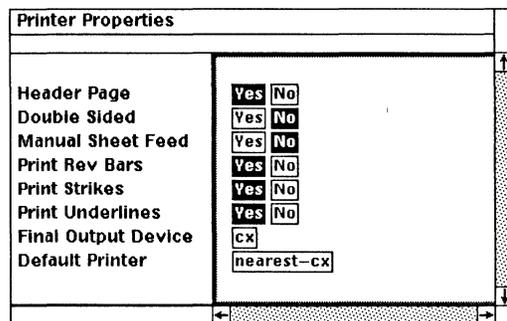
Figure 14-15. Printer property sheet with default settings

To toggle printing of an attribute:

✓ Point the mouse cursor into the Printer box of the Document Header.

✓ Execute the **Properties** command.
The Printer property sheet appears.

✓ In the *Print Rev Bars, Print Strikes,* or *Print Underlines* field, choose **Yes** or **No**.

✓ **Apply** the change and **Close** the property sheet.

Chapter 15

Desktop Management

In the previous chapter, we discussed the kinds of document management decisions you can make in an open document. This chapter concentrates on the ways you can store your documents in directories and how you manage document and directory icons on the desktop. The specific areas covered include

- **icon state** (how icon appearance varies and what the differences mean)
- **icons on the desktop and in windows**; ways to organize your documents in directories and the tools available to do that
- **links to documents and directories** on your desktop
- **permissions and locks**, two mechanisms that can protect your documents against access or alteration
- **links to other desktops** and the kind of document access they can provide

Icon States

There are five different icon states, each represented by a different appearance. Figure 15-1 shows the various states.
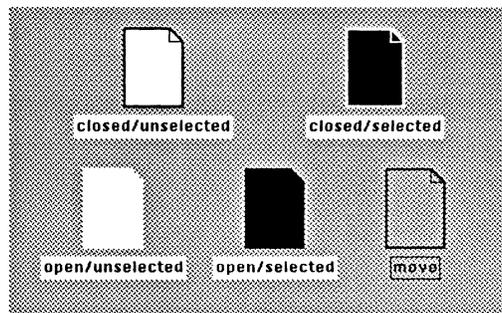
Figure 15-1. The five icon states

If an icon is displayed as a *white object* with a *black border*, it is unselected (no menus apply to it) and closed (its window is not displayed).

If an icon is displayed as a *black object* with a *white border*, it is selected (a specific menu applies to it) and closed.

If an icon is displayed as a *white object* with *no border*, it is unselected but its window is open and activity can take place within that window.

If an icon is displayed as a *black object* with *no border,* it is selected *and* its window is open. Even though the *contents* of the object are displayed in the window, the icon can still be *moved,* the icon's contents *prior to any changes made since the last save* can be *printed* or *copied,* and the property sheet for the icon can be opened and changed.

If you execute the **Open** command in this state, the window associated with the icon comes to the front of the other windows on the desktop. This can be a useful feature when a window is completely covered by other windows but its icon is not. You cannot **Cut** an open icon.

Finally, if an icon appears as a *border only,* it is being moved either during the **Move** command or after the **Paste** command. In both cases, the Icon Selected menu applies to the object. The *move* state is part of a broader *animation* state, which appears throughout the Interleaf publishing software when icons, windows, or graphic objects are being moved, resized, or pasted.

Icons on the Desktop and in Windows

The way you organize your desktop should be determined primarily by the way you will most easily find your documents. For most of your document organization, you will use commands on the Icon Selected popup menu and the Nothing Selected popup menus.

The Icon Selected Popup Menu

Each command on the Icon Selected popup menu (Figure 15-2) works the same on the desktop and in an open directory. Here, we are primarily concerned with the **Cut, Copy, Move,** and **Props** commands.

```
Print      →
Cut
Copy       →
Open       □
Deselect
Move
Props
```
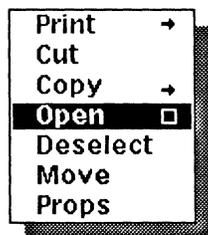
Figure 15-2. Icon Selected popup menu

You will see additional commands on this menu when the mouse cursor is inside a book window.

The Nothing Selected Popup Menus

Figure 15-3 shows the two Nothing Selected menus that refer to icons.

Desktop Nothing Selected Menu	...with Create Submenu	Directory Nothing Selected Menu
Create →	Create / terminal	Create →
Paste	Paste / book	Paste
Refresh □	Refresh / drawer	Close □
Close	Close / ◘folder	
	document	
	cabinet	
	Templates →	
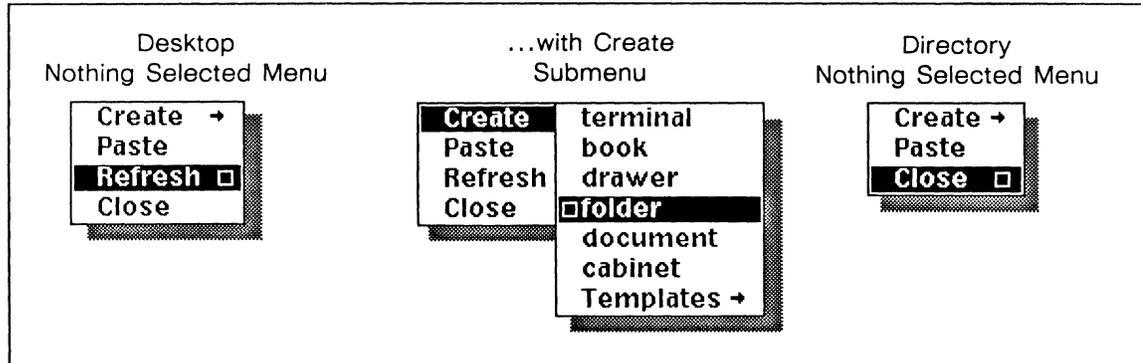
Figure 15-3. Nothing Selected popups and the Create submenu

The only difference between the desktop and directory window Nothing Selected menus is **Refresh**, which is not available on the Directory Nothing Selected menu; it is placed instead on a separate Window menu. The Create submenu is identical on the desktop and in a directory.

Creating Directories

You may already have directories in which you want your documents stored. If you do not, you have a choice of four kinds of directories — *folders, drawers, cabinets,* and *books* — on the Desktop and Directory Create submenus. Once you have created a directory, you can change its name on the Object property sheet, and you can even change the type of directory it is.

If you plan your directory structure in advance, you can minimize the amount of icon shuffling you will have to do later. However, you can always cut a directory and paste it (and all of its contents) into a new directory (including the desktop).

Moving Icons in and between Windows

Icons can be *moved* inside a directory window but not across window boundaries. To transfer an icon from one window to another, you must *cut* and *paste* it.

To put an icon into a directory:

✔ **Open** the directory.

✔ Select and **Cut** the icon from its current location.
If you want copies of the icon both in the current location and in the new one, use the **Copy** *command instead.*

✔ **Paste** the icon into the open directory window.

    ↙ When the outline of the icon appears in the directory window, move the mouse to place the icon where you want it in the directory, then click the **left** button to deselect the icon.

To move an icon inside a directory window:

    ↙ Select the icon and execute the **Move** command.

    ↙ With the icon in move state, move the mouse to place the icon where you want it in the directory, then click the **left** button to deselect the icon.

How Windows Localize Commands

Selection/deselection procedures and popup menu commands affect objects *only in the window the mouse cursor is in*. If the cursor is on the desktop, your actions will affect all objects on the desktop *that are not in another window*. If your cursor is in a window other than the desktop, your actions will affect *only those objects in that window,* not any of the other objects on the desktop or in other windows. Figure 15-4 gives an example of this distinction.
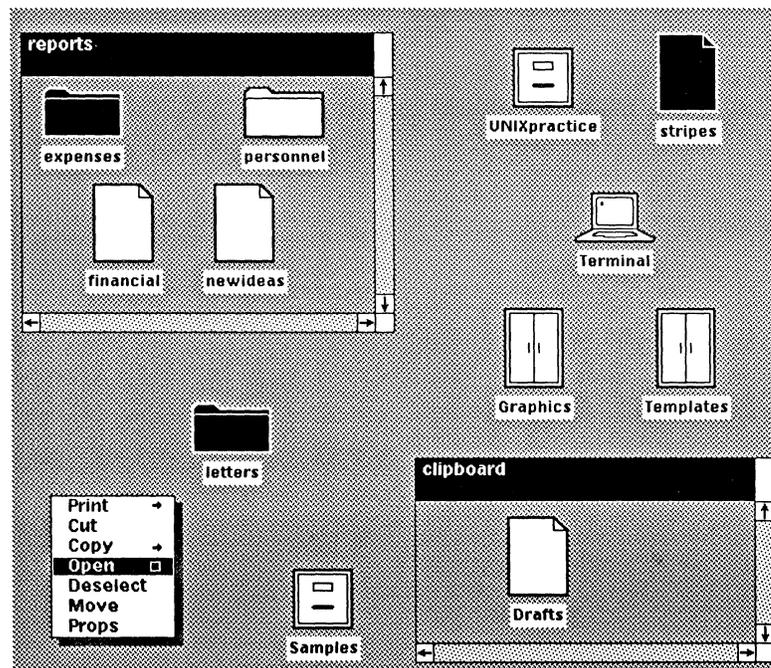
Figure 15-4. Selected objects in different windows

In the figure, if the **Open** command is executed when the mouse cursor is on the desktop itself, the *letters* folder and *stripes* document will be opened but the *expenses* icon in the inactive *reports* window will not.

Hidden Icons in Directory Windows

Icons in a directory or components in a document can be distributed over an area wider than the window they are displayed in. Since object icons are not connected the way text in a component is, you may lose track of an icon that is beyond the edge of the window. One of the most likely times for this to happen is when you reduce the size of a window.

If you cannot find an object that you do not remember cutting, resize the window in which you last saw it, temporarily making that window as big as the whole screen. If it is not visible then, it is not in that window.

Closed Directories with Open Contents

You do not have to keep a window open just because an object it contains is open. For example, if you open a folder and then open a document that resides in that folder, you can close the folder without affecting the open document. When you are finished reading or editing the document, you can close it without re-opening the folder. In many cases, closing a directory immediately after opening the object you need from it provides the most efficient use of your desktop.

The Object Property Sheet

The Object property sheet is a window in which you can see and change the key characteristics or *properties* of an object. Every icon on the desktop, with the exception of the clipboard, has an Object property sheet.

The properties displayed in the Object property sheet are:

- the object name
- *(for links only)* the pathname of the object to which this object is linked; links are described later in this chapter.
- the icon type (document, folder, etc.)
- the owner of that object
- the group that the owner belongs to
- which users have permission to 'read' (access) the object
- which users have permission to 'write' to (change) the object
- when the last change was made to the object
- when the object was last accessed

To open an Object property sheet:

↙ Select the object whose properties you wish to see or change and execute the **Props** command.

Figure 15-5 shows an Object property sheet with the most common *Read/Write permissions.*

Figure 15-5. Object property sheet

Owner, Group, and Others

The meaning of most of the terms on the Object property sheet should be self-evident. However, the terms we use to define the ownership and permissions may require some explanation.

Owner, group, and *others* are categories used to specify users and classes of users so that permission to read and change files can be assigned easily to the appropriate people.

Who owns an object on a particular desktop depends on how that object got to the desktop. The name listed in the *owner* field refers to a user's login name.

- All objects *created* on a desktop are owned by the user (login name) who created them.

- When an object is *copied* using ⌊ Copy → ⌋ ⌊ Normal ⌋, whether on the same desktop or from one desktop to another, the copy is owned by the user performing the copy operation.

- When an object is *copied* using ⌊ Copy → ⌋ ⌊ Link to ⌋, the owner of the copied object also owns the link. This command is described in detail later in this chapter.

- Two objects were provided on your original desktop — the *clipboard* and the *System* cabinet. You own the clipboard for your desktop, although you cannot view this ownership on a property sheet. By default, the *System* cabinet is owned by an account named *leafadm*.

A standard Interleaf installation provides two *groups* for users who have login names on the system.

- *leaf* includes one member — *leafadm* (the publishing software administrator).

- *user* includes all other users.

Your software administrator may set up different groups, but we do not provide instructions for setting up these groups.

Others consists of all users with login names outside of your group. So *leafadm* is *others* for members of the *user* group, and vice versa.

Default Properties of Newly Created Objects

When an object is created, the publishing software provides it with a series of default properties. These remain in place until changed by the owner.

The default *name* is the name of the object on the Create submenu — for example, the default name of a newly created document is *document*.

The default *owner* of the object is the login name of the person who created the object. The default *group* is the group shown on the property sheet of the directory in which the object is created; it cannot be changed unless the system administrator changes group membership. Normally the group will be *user*.

In almost all cases, the *owner* of a new object has both *Read* and *Write* permission, while the *group* and *others* will be permitted to read but not write to the object. However, if you or your software administrator has customized the Create menu, the permissions could be different. See the chapter *Document Templates* for more information on the Create cabinet.

Finally, the *time of last change* and *access* will be those on the system clock at the time the file was last written to and read, respectively.

Making Changes in Object Properties

On an Object property sheet, the owner can change these items:

- the *Name* of an object

- the *Read* and *Write* permissions

- the *Icon Type* (when more than one is shown)

- the *Pathname* (of a link)

You cannot change the *owner* of an object on its property sheet. Unless you belong to more than one group, you cannot change the *group* on the sheet.

Changing Icon Type

The Icon Type line in the Object property sheet displays one or more icon names, each in a box. The name of the current icon type is displayed in reverse video.

If the object is a directory, all of the directory types will be displayed. You can change the current icon type to any of the others shown by selecting the box for the icon you want. However, keep in mind that if you change a *book* to another type of directory, you may alter page and numbering information that is shared among documents.

You cannot change a *book* to another type of icon when any *document* in that book is open. You cannot change another directory icon to a book when *any* icon in the directory is open. A stickup menu appears when you attempt to change icon type under conditions that do not allow the change. See the chapter *Large Document in Books* for information about icons in a *book* directory.

Figure 15-6 shows the Object property sheets for a folder that is changed to a drawer. Notice that changing icon *type* does not change the icon *name*. You must make that change in the *Name* box.
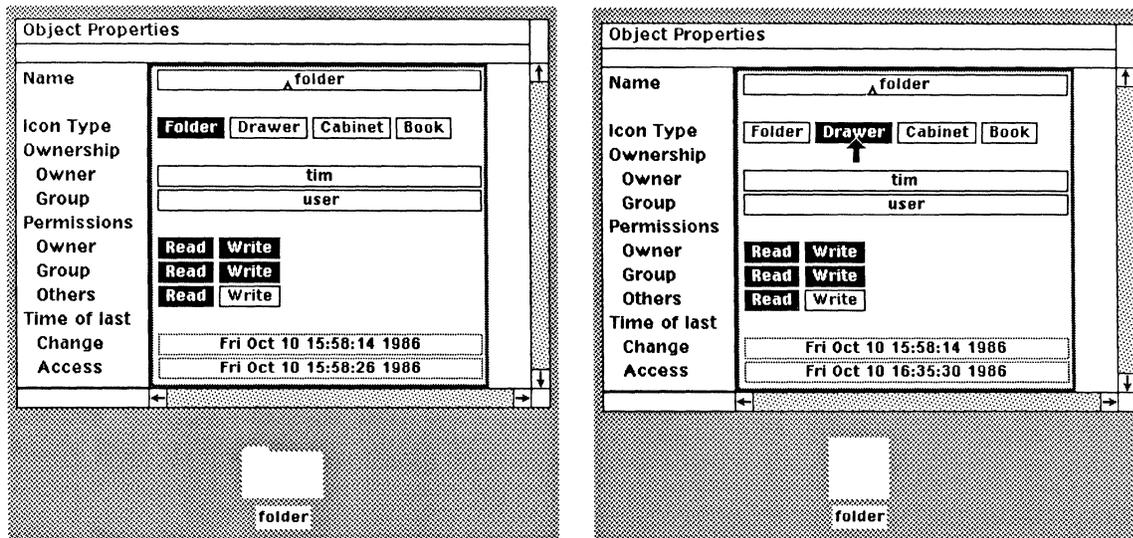
Figure 15-6. Changing icon type

Changing Permissions

Read and *Write* permissions are toggled on and off with a click of the left or right button while the cursor is in the appropriate box. A black box with white letters

means that the permission is available. All six of the permissions boxes can be turned on and off individually for any combination of permissions.

Change and Access Times

You cannot change the *Time of last change* or *Time of last access* directly on the property sheet. However, when you save an object or copy it using the **Normal** command, the new copy's time fields change to the time of copying. When you copy an object using **Link to**, the copy has the same times of change and access as the original.

In many network configurations, opening an icon may not change the time of last access on the Object property sheet.

Read and Write Permissions

For any object you own, the permissions you set for the object determine what you and others can do with that object *itself*, what you and others can do *inside* the object when it is open, and, occasionally, what you can do with a directory that contains that object.

Deciding on Adequate Permissions

Although exceptions exist, usually you will want to give yourself both *Read* and *Write* permission for the directories and documents you create.

When you give other users *Read* permission for a *directory*, you give them permission to open the directory on the desktop but not to change anything inside it.

When you give other users *Read* permission for a *document* that is in a *directory* where they also have *Read* permission, you give them permission to copy the document to their own desktops and to open the document.

When you give other users *Write* permission for a *directory*, you give them permission to put documents into the directory.

When you give other users *Write* permission for a *document*, you are giving them permission to overwrite your copy of a document with one of their own.

No matter how you set permissions, your system administrator can read any document.

Figure 15-7 shows how different permissions affect desktop capabilities. In it, we are changing permission of a *folder* containing *documents*; the folder is in a *drawer*. We assume Read/Write permission for the drawer and the documents.

You can execute the checked commands when a folder containing documents has the permissions shown at the top of each column.

On the folder icon itself — Folder permissions

	Read Write	**Read** Write	Read Write
Cut	✓	✓	
Copy	✓	✓	
Open	✓	✓	
Move	✓	✓	✓
Props	✓	✓	✓
Purge	✓		

Inside the open folder window — Folder permissions

	Read Write	**Read** Write	Read Write
Cut	✓		N/A
Copy	✓	✓	N/A
Open	✓	✓	N/A
Move	✓		N/A
Props	✓	✓	N/A
Create	✓		
Paste	✓		

On a drawer containing the folder — Folder permissions

	Read Write	**Read** Write	Read Write
Cut	✓	✓	✓
Copy	✓	✓	*
Open	✓	✓	✓
Move	✓	✓	✓
Props	✓	✓	✓
Purge	✓		

* If you copy a directory that contains objects for which you have no permission, the directory is copied but any no-permission objects inside it are not. Stickup menus will inform you of this situation.

Figure 15-7. Effects of read/write permissions

This example shows just some of the nearly infinite number of permission variations. For instance, if the folder in the example had no contents, some permissions would affect your access to commands differently. If you did not have

Read/Write permission for the drawer or the documents, the effect of the folder's permissions could be different.

Remember that no matter what the permissions are, only the owner can *change* permissions on the property sheet. If another user has *Write* permission, that user can change the document name.

If protection of your desktop objects is critical, you should experiment with permission combinations so that you understand precisely what will happen in your particular situation. The following section explores some of the issues you might want to think about.

Read-only Directories and Documents

The *System* cabinet you received as part of your Interleaf desktop is owned by *leafadm,* your publishing software administrator. Everyone with a desktop has *Read* permission for this icon, and so can make copies of it, but only *leafadm* can add to it or otherwise change its contents. Because of these restrictions, the *System* cabinet is a *Read-only* directory. In addition, all of its contents are protected by being individually designated *Read-only.*

Sometimes you will want to protect a directory or a document in this way from *yourself* as well as from others. For example, you might create a new *cabinet* to hold templates. You need permission to read and copy the contents of this cabinet, but you usually will not want to change the contents of the templates.

The greatest protection you can provide in this case is *Read-only* permission for the *cabinet,* the *drawers,* the *folders,* and each of the *documents.* This may be more protection than you want. If, for example, you want to be able to put new *drawers* in the *cabinet* while not changing the contents of existing *drawers,* you should give yourself *Read* and *Write* permission for the *cabinet,* but *Read-only* permission for objects *in* the cabinet. Everyone with access to your desktop can copy the contents of the drawer, but no one can put anything in it.

If you have a *write-protected* document in a *writeable* directory, you will still be able to **Cut** and **Purge** the directory. If you *write-protect* a directory with anything inside it, you will be able to **Cut** but not **Purge** the directory.

Write-only Directories and Documents

You can set permissions so that the *owner, group,* or *others* have *Write* permission but not *Read* permission. While this option has some value in special situations, like communications files that the owner wants others to be able to add to but not read, you are unlikely to find *Write-only* permission a useful choice while working on the desktop.

Desktop Links

Using the [Copy →] [Link to] command, you can create an icon that represents a link to another object on your desktop or any object inside any directory on your desktop. Links offer several advantages.

- You can simplify access to an object that is several directories deep by creating a link to it on your desktop.

- You can use links to incorporate standard documents into different *books*. This technique is discussed in the chapter *Large Documents in Books*.

- If your system administrator has set up a desktop link that allows users to share files throughout a network, you can link to objects on other users' desktops.

- Every desktop has access to the objects in the *System* cabinet through links, saving the storage space that would be required to put a copy on each desktop and making sure that updates to the cabinet are reflected on all desktops. You can further simplify access to these items by making a link on your desktop to the objects you most frequently use.

To create a linked object:

✔ Select the object *to which* you want to create a link.
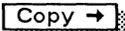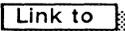
✔ Execute the [Copy →] [Link to] command.

   A link to the object is stored on the clipboard in the form of the icon you copied.

✔ Paste the linked object where you want it stored.

Editing Properties of Linked Objects

The icon for a linked object looks exactly like the icon for the object it is linked to. For example, if you create a link to a document named *sunrise,* the linked object also appears as a document named *sunrise.* When you open the linked object, you are, in effect, opening the object to which it is linked.

Four properties of a linked object may be changed on its Object property sheet:

- the name of the linked object (the name that appears on the desktop under the icon);

- the object's *target* — the actual document or directory to which the link points (initially, this target is the object you copied);

- the icon type of the linked object if it is a directory;

- the permissions for the linked object *and* its target.

The linked object's name is changed in the same manner as the name of any other object. The target of the link can be changed by editing the *Pathname* field, which appears on the Object property sheets of all linked objects (Figure 15-8). *Pathname* contains the full name of the target object as it is listed in the operating system.

```
Object Properties

Name           |            sunrise            |
Pathname       | /u/tim/desktop/new.fdr/sunrise.doc |

Icon Type      | Document |
Ownership
  Owner        |             tim               |
  Group        |             user              |
Permissions
  Owner        | Read | Write |
  Group        | Read | Write |
  Others       | Read | Write |
Time of last
  Change       |     Thu Jul  3 11:28:56 1986     |
  Access       |     Thu Jul  3 11:28:55 1986     |
```

Figure 15-8. Link Object property sheet

By editing the pathname of a linked object, you can make the link point to a different object, including an object on another desktop. You edit the pathname the same way you edit the object name. If the pathname is too long to fit in the field, it will scroll into view as necessary when you move the editing caret. However, the pathname cannot be longer than 256 characters.

Users without some degree of knowledge about the operating system should consult their system administrator before editing a pathname.

Link Permissions and Ownership

When you change the name, pathname, or icon type for a linked object, the change affects that object only. However, the *Ownership* and *Permissions* properties displayed on the link's Object property sheet are actually those of its target object. If you are able to change the permissions on the linked object's property sheet, any changes you make will be applied to the target object.

Keep in mind, however, that unlike a normal copy, a linked copy is not owned by the user making the copy. Therefore, you will not be able to change all (or in some cases any) of the properties of a linked object that points to an object you do not own.

Since opening a linked object gives you access to its target, the Change and Access times listed on the link's property sheet are the same as those for the target object.

Cutting a Link

Cutting a linked object does not affect the target object, nor does cutting a target object automatically cut linked objects that point to it. However, cutting the target object without also cutting (or editing the pathname of) the object linked to it leaves the links *pointing into space*, and therefore useless. When you try to open a linked object after its target has been cut, a stickup message informs you that you do not have permission to open the object. When you open that linked object's property sheet, the status line tells you that the link points to a nonexistent file.

If you paste the target object in the same directory it was cut from, the link will be re-established. However, if you paste the target object in a different directory, the link will remain broken.

Editing in Linked Objects

Whether you access a document directly or through a link, any changes you make and save are reflected in the stored document and visible the next time you open the document by either means. If you do not have permission to save changes, you will be given the opportunity to rename the edited document.

If you use the **Rename** command to copy either the linked object or its target, the original linked object remains linked to the original target. The renamed object does not become part of the link relationship, and you are the owner of it.

Document Locking

When you select and attempt to open some documents, the Document Locked stickup appears. This stickup indicates that a **lock**, which is a separate file related to the document, is in place for that document for one of two reasons:

- The document is already open, either through a link or at another workstation using the same desktop.
- The document file or its lock file was damaged in some way that generates a lock message.

Figure 15-9 shows the Document Locked stickup. The stickup provides you with information you need to determine whether you will honor or override a lock. In addition to the name of the document, this information includes the time and date when the document was opened, the name of the user that has the document open, and the workstation on which that user is working.

```
                          document is locked.
It was opened at Fri Oct 10 17:45:48 1986, by user "tim" on node Jasper.
              Do not override the lock without tim's permission.
                    Do you wish to override the lock?


                          [Yes]  No
                                  ↑
```
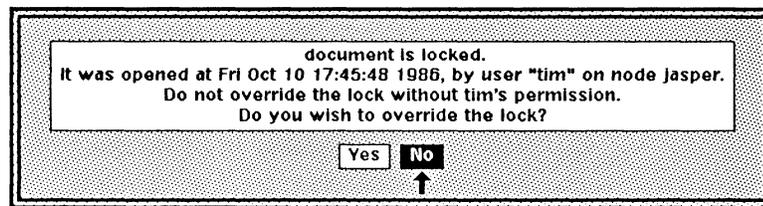
Figure 15-9. Document Locked stickup

Locks are designed to prevent two or more users from making simultaneous changes to the same document. When two users have the same document open, one user does not see the changes that the other is making. As a result, both could edit what they assume to be the *final* version of a document, only to learn that sections one was working on had been changed or even eliminated by the other. Whoever last saves the document overwrites all of the changes made by the other user.

Because of this potential for lost work, the default on the Document Locked stickup is **No**. As the stickup recommends, you should get the approval of the user who has the document open before overriding the lock.

If your login name and workstation are shown on the stickup, check to see that you do not have the document open in another window on your desktop. If you have determined that the document is not open on your desktop, the message may indicate a damaged lock or document.

By answering **No** to the Document Locked stickup, you cancel the **Open** command and deselect the document you attempted to open. If you answer **Yes**, the document opens (assuming you have *Read* permission on its property sheet).

Do not override a lock without determining the current
status of the locked document. Inappropriate overriding of
locks can result in the *permanent* loss of information.

Non-lockable Documents

When you open a document, a lock is created in the directory containing the
document. However, if you have *Write* permission for a document but not for the
directory containing it, you will not be able to lock the document. You can open
it, but you will be warned by a stickup menu that the document cannot be locked
against concurrent opening. You should be careful about changing the document
under these conditions, and should try to determine that no one else is editing it.

Access to Other Desktops

Your system administrator can set up links that let you examine and copy documents and directories on other users' desktops and let them do the same on yours. Although this can be done in many different ways, we recognize two standard desktop link configurations.

The Bulletin Board

The Bulletin Board offers a limited sharing of objects among the users on a network of workstations. Your system administrator creates a directory called *Bulletin Board* (or any other name), and then creates, or instructs you to create, a link from your desktop to that directory. The permissions on the *Bulletin Board* are set in such a way that all users can **Cut**, **Copy**, and **Paste** in it.

Once everyone has links to the *Bulletin Board*, the directory becomes a transfer point for objects. You can **Copy** or **Cut** objects from your desktop and **Paste** them into the Bulletin Board so that they are available to all of the other users on your network, who can then **Copy** or **Cut** them from the Bulletin Board and **Paste** them onto their own desktop.

In general, use of a *single* Bulletin Board limits *what* other users have access to but does not limit *who* has access to it. Your system administrator may set up more than one Bulletin Board so that all users have access to one set of shared objects but only selected users have access to another set of objects.

You may be able to paste a link into the Bulletin Board, giving other users access to an object that is still in its original location on your desktop. However, for this to work, your system administrator must have set up your workstation in a particular way so that the file systems used by all workstations on the network are available to it. In many cases, you will have to put the actual object, not a link to it, in the Bulletin Board to make it available to others.

The Full Desktop Link

The Full Desktop Link provides complete access to the desktops of other users on the network. If your system administrator has implemented this configuration, you will have a cabinet on your desktop called *Desktops* (or any other name). When you open that cabinet, it will display a directory window containing cabinets labeled with the name of each of the other users on the network.

When you open a cabinet with the login name of another user, that user's desktop will be displayed in the directory window, if the file system it is stored in is readable by your workstation. You can then **Copy**, **Open**, check **Props** of, or **Print** any of the objects you find there, assuming you have permission to perform these operations. If the user whose desktop you are examining has used the default permissions for objects, you will not be able to edit a document or change the properties of any object on that user's desktop.

In most cases, the permissions on another user's desktop will prevent you from cutting icons. However, for safety and courtesy, avoid the **Cut** command while working through a link to someone else's desktop. Your ability to **Paste** on someone else's desktop is determined by the same permission that would let you **Cut** from it. If you cannot paste a document on someone's desktop and you want them to have it, let them know where it can be found and have them **Copy** it from you. This has the added benefit of making them the owner of the copy.

In some cases you may be able to **Move** an object in the link window, but the object does not move on the owner's desktop.

If there is one particular icon on someone else's desktop that you have frequent need of, you can make a link to that icon and paste the linked object directly on your desktop for easier access.

Other Desktop Links

Two other common desktop links for limited sharing of objects are the Scanner directory and the DataComm directory. If your facility has optional scanner equipment, images scanned in by this equipment are stored in a Scanner directory on one desktop. The Scanner link provides other users with access to those scanned objects.

A DataComm link can provide access to any communications port.

Copying in the Operating System

Using a terminal window and operating system commands, you can copy objects from other desktops on the network. The procedure for copying in the operating system is described in the manual *System Administration*.

Chapter 16

Large Documents in Books

A **book** is a special directory that creates a relationship among the documents it contains. The following features are available in a book:

- Consecutive documents can share a **single page number series**, with each document (or selected documents) automatically inheriting its starting page number from the document preceding it.

- **Autonumber streams** available in one document are available in all documents. You can extend any numbered series across several documents.

- You can create **references** to numbers in an autonumber stream, even if the reference and the number are in different documents.

- A change anywhere in the book automatically updates affected page numbers, autonumbers, and autoreferences in documents throughout the book.

- When you do **concurrent editing** in two different documents located in the same book, changes in one place are immediately visible in the other.

- The **Next** and **Previous** page commands scroll from one document to another when necessary.

- You can **print the book** as if it were a single document.

- You can automatically **create an index** for one or more documents.

- You can automatically **create a table of contents** for one or more documents.

The indexing and table of contents capabilities of the book are described in the chapter *Indexes and Tables of Contents.*

Creating and Organizing a Book

You create a book the same way you create any other icon, using the Create submenu on the Nothing Selected menu. Like other icons, the book has an Object property sheet on which you can change its name. Figure 16-1 shows a book icon as it appears in the publishing software.
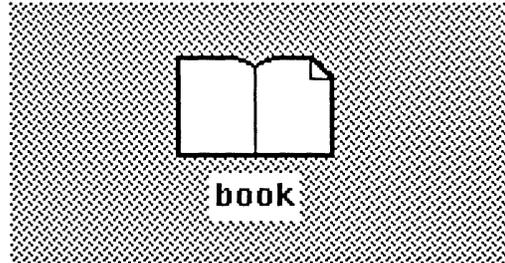
Figure 16-1. Book icon

Changing Icon Type

You can change a non-book directory to a book using the *Icon Type* line on a directory's Object property sheet. However, you can only do this when the directory window and everything in it are closed.

You can change a *book* to another directory type only when all *documents* in the book are closed. Open icons other than documents do not prevent you from changing a book to a folder or another directory. If you try to change icon type under the wrong conditions, a stickup menu notifies you that the change cannot be made.

Before you change the *Icon Type* to or from *book*, be sure that you understand how this change may alter page and autonumbers.

Book Membership

You can create or paste any icon, including a book, inside a book window. However, only *books* and *documents* really become part of the book; that is, they share page and autonumbering information, print when the book is printed, and contribute entries to an index or table of contents created in the book.

Any other icon can be *stored* in a book but does not become part of the book. You can use other icons to include material related to the book without affecting page and figure numbering. For example, if you have memos or comments about a reference manual, you can put those items in a folder and then put the folder

inside the book containing the chapters of the manual without affecting the documents that make up the manual.

Documents and books become part of a book that contains them. Other icons are merely stored in the book.

Icon Placement

Document Order

Figure 16-2 shows the order of documents in a book, which is the same as the reading order of text — start at the top, move from left to right, then go to the left end of the next row, and move from left to right again. You can change the printing and page numbering order of a document by changing the position of the document in the window. If you **Paste** a new document into a book or **Cut** a document from it, page and autonumbers in documents following the pasted or cut document are updated to reflect this change.
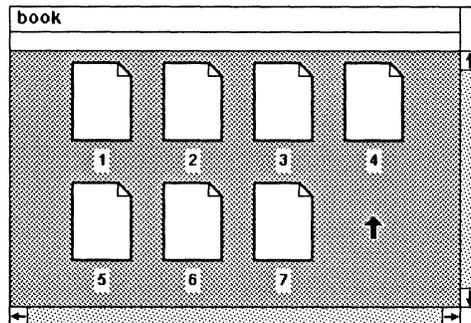
Figure 16-2. Document printing and numbering order in a book

Sub-books

You might want to create one book for each volume of a publication and then collect those volumes in one main book. We call these books-within-books **sub-books**. Sub-books form no barrier to consecutive page numbering and printing or shared autonumber streams, but they give you the option of printing, indexing, or making a table of contents for the sub-book only.

Figure 16-3 shows the book in Figure 16-2 with some of the documents included in a sub-book. The documents have the same relationship to each other as they did before the sub-book was created.

a. book with sub-book   b. contents of the sub-book
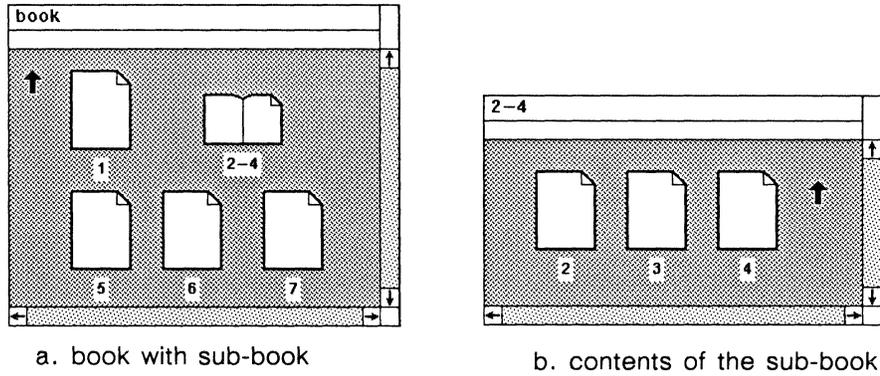
Figure 16-3. Sub-book in book

The Lineup Command

Sometimes the order of icons in a book is ambiguous. The **Lineup** command, which is available on the Icon Selected menu within a book, can help you determine document order. This command moves any selected icons to the nearest point on an invisible grid within the book window, making it easier to see if any icon is not in the order you intended. The *nearest* grid point is not always the *next unoccupied* point — icons are not necessarily adjacent to each other after **Lineup**.

Lineup affects all icons in the book (not just documents and books). Figure 16-4 shows the results of a **Lineup** command in one book.

a. before Lineup   b. after Lineup
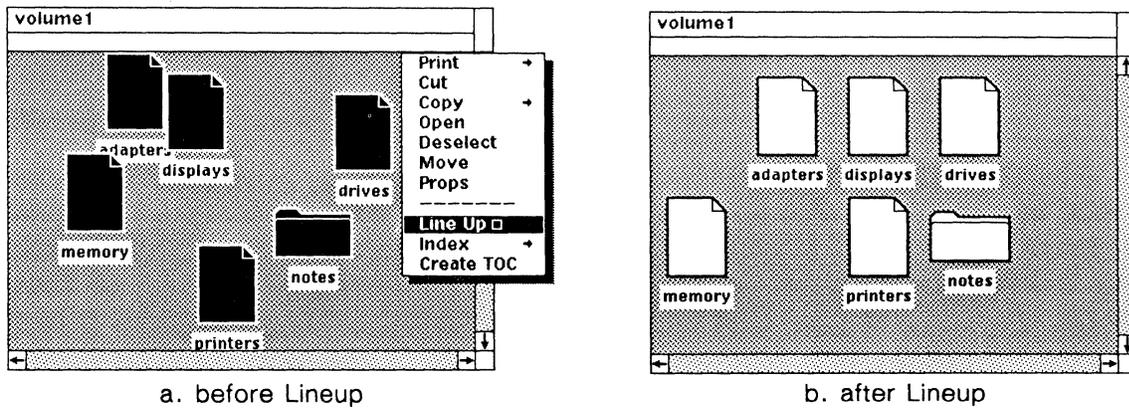
Figure 16-4. Effect of Lineup command

If two or more icons are so close that they would move to the same grid location or if they already overlap so that one is invisible, the **Lineup** command offsets the icons so that you can distinguish between them. In Figure 16-5a, the name and icon for *adapters* completely obscures the name and icon for *displays*. When all icons in the window are selected and the **Lineup** command executed, the second icon becomes visible (Figure 16-5b).

a. Before Lineup   b. After Lineup
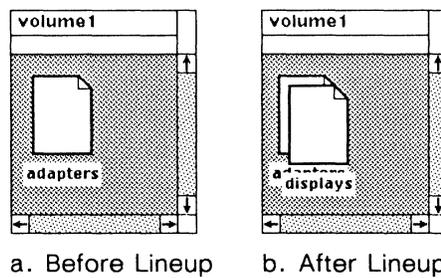
Figure 16-5. Lineup command and overlapping icons

Hidden Icons in Books

As with any directory window, you can change the size of the book window so that it does not show all of its contents. You can also move icons out of the viewing area of the window.

Because icon placement in a book affects page numbering, autonumbering, and printing order, an icon outside the viewing area of a book window can pose problems that do not occur in other windows. If you think that documents in a book are misnumbered, **Resize** the window to be sure that there are no icons hidden outside the old window boundaries.

Page Numbering

Two properties on the Page property sheet control the way a document's page numbering interacts with other documents in a book. These properties are the *Starting Page #* and the *Page # Style*.

By default, the *Starting Page #* line for newly created documents shows the **Inherit** box *on*. This enables the document to inherit its starting page number from the previous document in a book. You can turn this property on or off at any time to specify an inherited or a fixed starting page number for any document.

If a document in a book has a different *Page # Style* from the document preceding it — for example, **Arabic** numerals in one and **Lowercase Roman** numerals in the other — the second document will not inherit the first document's page numbering, even when **Inherit** is *on*. Check this property if page number inheritance does not seem to be working correctly.

To enable a document to inherit its starting page number:

    ✔ **Open** the Page property sheet for the document.

    ✔ Examine the *Starting page #* field. If **Inherit** is shown in reverse video, it is *on*, so you need not take additional steps.

    ✔ If **Inherit** is not *on* (in reverse video), point the cursor into the **Inherit** box and click the **left** button to turn it on.

↙ **Apply** the change.

↙ **Save** the document.

To specify a fixed starting page number for a document:

↙ **Open** the Page property sheet for the document.

☐ In the *Starting page #* field, type the desired page number in the Page Number box and press **RETURN**.

When you enter a number, the inherit property is turned off automatically, and the Inherit box changes from reverse to normal video.

↙ **Apply** the change.

↙ **Save** the document.

To specify a fixed starting page number of 1:

☐ In the *Starting page #* field, point the cursor into the **Inherit** box and click the **left** button to turn Inherit *off*. The default page number when you toggle **Inherit** is 1.

↙ **Apply** the change.

↙ **Save** the document.

Inherit only applies to page numbers. It has no effect on autonumber tokens in text.

Interrupting Page Numbering

Documents with **Inherit** on always inherit their starting page number from the document preceding them, regardless of whether that preceding document has **Inherit** *on*. If a document has Inherit *off* (or has a *Page # Style* that differs from the previous document), it becomes the new starting point for any succeeding documents that have **Inherit** *on*.

Figure 16-6 shows how **Inherit** can affect page numbering in a book. In the example the first document contains 5 pages, the second contains 10 pages, and the third contains any number of pages. When all documents have **Inherit** *on*, the third document inherits its starting page number from the second document, which inherits its starting page number from the first document. If the second document is changed to have a fixed starting page number of 1, the first document makes no contribution to the starting page numbers of the second or third.
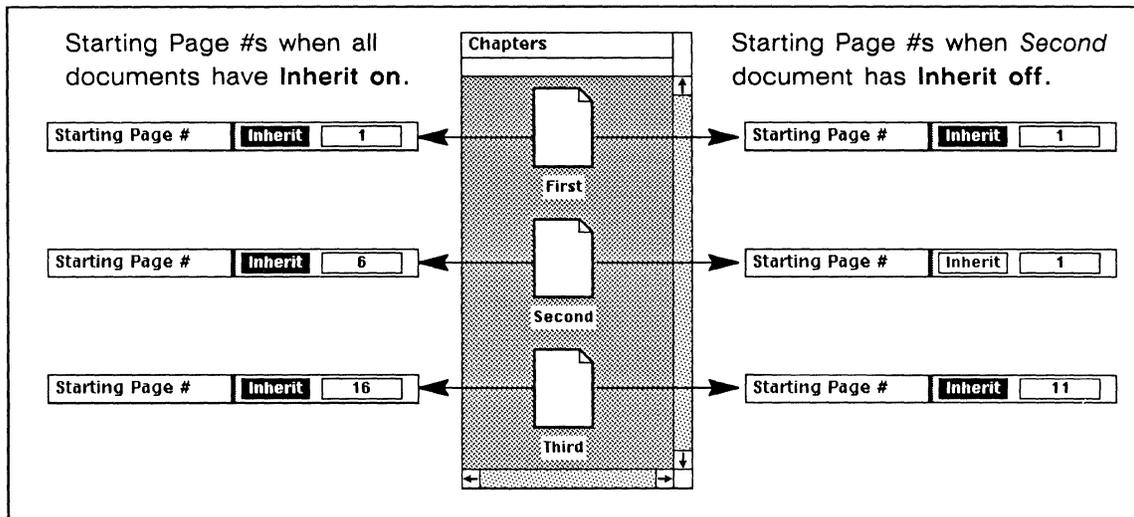
Figure 16-6. Effect of Inherit in a book

Page # Prefix

The *Page # Prefix* line on the Page property sheet allows you to insert a prefix — the chapter number or the word *Page*, for example — in one place and have it display and print anywhere the page number token is used in that document. In this chapter, *16-* is the prefix for each page number. Page numbers for entries in automatically created indexes, tables of contents, and page number references include the *Page # Prefix*.

Unlike the number itself, page number *prefix* cannot be inherited from another document. If you want all documents in a book to have one particular prefix, you must insert that prefix in each document's Page property sheet or use a template document with the prefix already in it.

The Page property sheet, on which you change these properties, is discussed in the chapter *Page Design*.

Dividing Documents in Books

There are no limits, either in number of documents or number of pages, to the size of a book. We recommend document lengths of from 1 to 50 pages in a book. Because images use more memory per page than text, you should reduce the size of a document as the number of images in it increases.

Each new document in a book begins a new page. Text cannot flow continuously from one document to the next.

Headers and Footers in a Book

Each document in a book has its own Header/Footer properties (Figure 16-7). These properties are not inherited.

Headers/Footers			
Page Layout	Single Sided	**Right 1st Page**	Left 1st Page
Bleed	**No**	Yes	
Diff 1st Header	No	**Yes**	
Diff 1st Footer	**No**	Yes	

Figure 16-7. Header/Footer properties on Page property sheet

If all documents in a book have single-sided layout with no differentiation in headers or footers between the first page and other pages, page layout is not an issue. For other options, you should consider the impact of a particular layout on the overall appearance of the *book*.

As you create new documents in a book, you can set the Page properties you want in each one. If you are dividing an existing large document into more manageable documents of fewer pages, you should set the Page properties for the large document to the settings that most of the documents should have *before* you begin dividing it. You can then make any changes necessary to create a special layout for any one document — turning *Diff 1st Header* on in the first document, for example.

Diff 1st Header & Footer

When you set the *Diff 1st* properties for a document, you affect that document only. If, for example, you want the first page of a document to have no header at all, you can set *Diff 1st Header* to **Yes** and delete all header information in the frame on that page.

Depending on the way you divide a document in a book, you may not want first-page headers or footers to differ from those of other pages in any but the first document in a book. Figure 16-8 shows two situations requiring different treatment of the *Diff 1st Header* property.

Document 1 and Document 2 make up a single chapter. In this case, Document 1 has *Diff 1st Header* on; Document 2 has it off.

In this case, Document 2 begins a new chapter, and so *Diff 1st Header* is on for both documents.
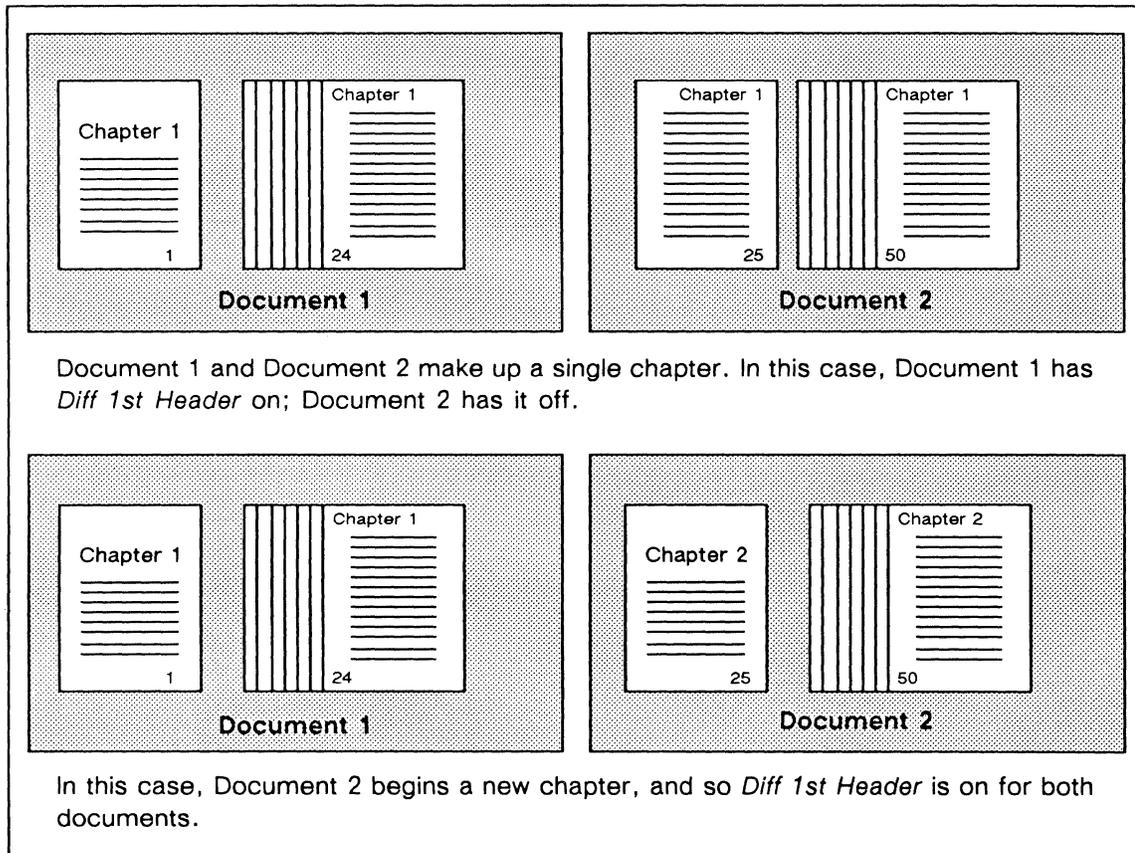
Figure 16-8. Diff 1st Header property in a book

Page Layout

As long as you have the same *Page Layout* setting for all documents in a book, the software maintains the proper header/footer placement in all documents, no matter how many pages each document has. Because of the definition of **Left 1st Page** and **Right 1st Page**, you can have even numbers of pages in some documents and odd numbers of pages in other documents without posing layout problems. **Left 1st Page** means that *page 1* and any other odd-numbered pages are on the *left*, and any even-numbered pages are on the *right*. **Right 1st Page** means that *page 1* and any other odd-numbered pages are on the *right*, and any even-numbered pages are on the *left*.

Figure 16-9 shows how a **Left 1st Page** layout responds to repagination in a book.

Figure diagram showing:

Document 1 (Right 1st Page)
Chapter 1 — page 1-1
Chapter 1 — page 1-2

Document 2 (Right 1st Page)
Chapter 1 — page 1-3
Chapter 1 — page 1-4
Chapter 1 — page 1-5

With **Right 1st Page** on, both page 1, which begins Document 1, and page 3, which begins Document 2, are right-hand pages. Document 1 has *Diff 1st Header* on; Document 2 has it off.

Document 1 (Right 1st Page)
Chapter 1 — page 1-1
Chapter 1 — page 1-2
Chapter 1 — page 1-3

Document 2 (Right 1st Page)
Chapter 1 — page 1-4
Chapter 1 — page 1-5
Chapter 1 — page 1-6

When a page is added to Document 1, the first page of Document 2 inherits an even number. With **Right 1st Page** on in Document 2, the headers and footers display on the correct side of the page because **Right 1st Page** indicates that all even numbered pages, including those numbered by inheritance, should be on the *left*. This assures proper header/footer placement no matter how you divide documents.

Figure 16-9. Page Layout in a book

In general, you should not have different *Page Layout* settings for different documents in a book, unless you are planning a special layout that requires this difference. If you mix layouts, you may produce undesirable placement of your headers and footers when you edit, move, add, or delete documents in a book.

More information on headers and footers is available in the chapter *Header and Footer Frames.*

Autonumbering and Referencing

In a book, any **autonumber stream** available in one document is available in all documents. This allows you to number figures, for example, consecutively throughout an entire book. When you add to, subtract from, or otherwise alter a stream, your changes are reflected in all documents containing that stream.

You need not have an instance of an autonumber stream in every document to maintain the consecutive numbering. If you insert a *List* token showing number *1* in the first document of a book and insert the second token from that stream in the tenth document, that token will show number *2*. (For convenience, we refer to the *first* and *second* document in many of the examples to follow, but the rules described hold true for any documents with consecutive numbers in the same stream.)

Detailed information about autonumbers and autoreferences is available in the chapter *Autonumbering and Referencing*.

Restarting Streams

You can **Restart** a stream in the second of two documents so that you use a fresh *Starting Value* in each document while maintaining all of the other properties of the stream.

To restart an autonumber stream in a succeeding document:

 ✓ To restart an autonumber stream in the second of two documents, you must have instances of that stream already in place in both documents. If no instance exists in the second document, **Create** a Level 1 token.

 ✓ **Position** the text caret before the first token in the second document.

 ✓ Execute the ⌊ Create → ⌋ ⌊ Auto # → ⌋ ⌊ *streamname* → ⌋ ⌊ Restart ⌋ command.

 The autonumber token after the caret must be Level 1. If it is not, you will see the error message: You can Restart only before a Level 1 token or at the end of a stream.

 Once the split has been made, all instances of the stream following the text caret will be renumbered beginning with the Starting Value on the Autonumber Stream property sheet.

Merging Streams

As in single documents, the **Merge** command allows you to join two sets of numbers in different books that belong to the same stream.

To combine two numbered lists in a book into one:

> ✓ **Position** the text caret anywhere after the first token in the second list.

> ✓ Execute the [Create →] [Auto # →] [*streamname* →] [Merge] command.

> *The tokens in the second list will be renumbered to make them continuous with the first list.*

Purging Autonumber Streams in Books

To purge unused autonumber streams from one document in a book, you must purge them from all documents in a book.

To purge all unused autonumber streams from a book:

> ✓ **Open** one of the documents in the book. For convenience, you can use the first document.

> ✓ Execute the [Misc →] [Masters →] [Purge # Streams] command on the Text Location popup menu in any document.

> *The Purge # Stream stickup will appear.*

> The masters for all autonumber streams without instances will be deleted. Please Confirm.
>
> [Confirm] **Cancel**
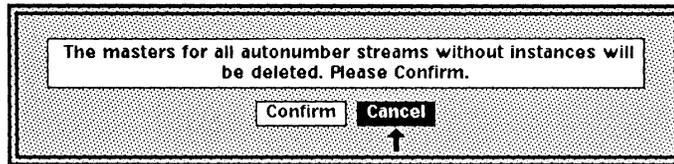
> ✓ **Confirm** to continue with the process.

If no additional stickups appear, there are no streams without instances anywhere in the book and the purge process ends. If there are streams with no instances in the book the Book Purge # Stream stickup appears.

> If any other documents in this book have been saved since the creation of the stream(s) that you are purging, the stream will reappear when those documents are opened or the next time this book is opened. If you wish to totally purge the stream from this book, all of those documents should be opened, the purge command should be repeated, and the documents should be saved.
>
> **Continue**

> ✓ **Continue** on the stickup.

> ✓ **Save** the document.

> ✓ **Repeat** all of the steps above for each document in the book (including those in sub-books). You can close each document after purging and saving.

You can simplify the stream purging process by opening several documents at a time, purging the unused streams in one of them, and saving and closing all of the documents. This purges the streams from all of the documents you saved. Repeat these steps for additional groups of documents until you have purged the unused streams from all documents in the book.

If you use this technique, remember that you can open no more than 16 windows at one time on the desktop. For *documents*, especially if they are long or contain many images, you should open fewer windows at one time so that you do not reach the limit of your workstation's memory.

The *List, Outline, and **New*** streams on the *Create Auto #* submenu cannot be purged.

Gaps between Stream Levels

When you create tokens from a multilevel stream, you cannot skip levels. This is true both within a document and from document to document in a book. For example, if you create a Level 1 token and then try to create a Level 3 token, the software will instead create a Level 2 token. If you specify one level in the first token of a document and get another, check the preceding document to see what level *its* last token uses.

Autoreferences

An autoreference is a token whose value is linked either to the current value of an autonumber token or to the number of the page on which the autonumber currently appears. Autoreferences can be created using either the **Create Ref.** command or the **Copy Ref.** command.

You can create **autoreferences** within or across documents in a book. However, you should be especially careful about cutting autonumbers in a book, since you may cause references to display NO TAG in other documents.

Autonumbers and references to them are linked by a **tag**, which is a unique name identifying the autonumber. If you use the **Copy** command to create a reference, the software creates a tag for the autonumber you copied and inserts that name in the property sheet of the reference. You can also create your own tag for each autonumber.

If you make your own tags, you cannot use the same tag to identify two autonumbers in the same book. If you attempt to enter a duplicate name in the Autonumber Token property sheet, the sheet's status line shows an error message.

If you paste a document into a book, and a tag in the pasted-in document is the same as a tag already in the book, one of these autonumbers will have its tag eliminated. Also, if you **Rename** a document in a book or **Copy** it and **Paste** it back into the book, you disrupt the connection between autonumbers in the document and any references they had. You should avoid using **Rename** in a book.

Page and Autonumber Updates

When you make a change that adds to or subtracts from the number of pages in one document, or you create or delete an autonumber token in a stream that extends across more than one document, these changes are recorded in a summary file as soon as you save the document in which the changes were made. When you open a document in a book, that document checks the summary files of the other documents to determine its starting page number and the starting number for any autonumber streams it contains.

If you have two documents from a book open, creation or deletion of an autonumber token in the first one is immediately visible in the second if it contains tokens in the same autonumber stream. Any autoreferences affected by autonumber changes also are updated immediately. Also, *page* numbers in an open document with **Inherit** on are updated when the number of pages changes in an earlier document.

If another user makes changes to documents in a book you own, you may not see these changes until you close the book and everything in it.

Pasting Documents into a Book

You can cut or copy an icon from any location and paste it into a book. *However, to be sure that it becomes part of a book's numbering streams, you should open and save any document after you paste it into the book.* Documents created in the book automatically become part of the book, but documents pasted from outside must be integrated into the book before their page and autonumbering information is reliably reflected in other documents.

This is the specific behavior of documents in a book when a new document is pasted into the book.

- Documents open at the time you paste will be updated immediately.
- Documents opened *after* you paste and save the new document in the book will be updated when opened.
- Documents opened *before or after* you paste a new document show updated page and autonumbers when you open the new document, even if you have not saved the new document in the book.

Printing and the Book

You can select a book and execute one of the desktop **Print** commands to print all of the book's contents. When you print a book, the documents in it are printed so that the finished book is in proper page order. Just as they do on screen, documents that have their *Starting Page #* set to **Inherit** will be numbered consecutively according to the left/right, top/bottom order of the book. **Inherit** has no effect on the order of printing.

The desktop printing feature handles documents in groups of 20. If you have more than 20 documents in a book, some group of documents may print out of order when you print the book.

When you execute a **Print** command for a book icon, the Directory Printing stickup (Figure 16-10) appears, providing an opportunity to **Confirm** or **Cancel** the command.
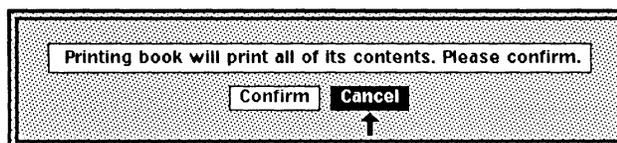
Figure 16-10. Directory Printing stickup menu

Note that although the message in the stickup says that you will print all of the directory's contents, in a book you will print only *documents* and the contents of any *sub-books*. The contents of other icons inside the book are not printed, nor are the contents of other icons in a sub-book.

If a book's contents are edited while desktop printing is in progress, the results of the print operation may be unpredictable. You should not move, cut, or paste icons in the window of a book being printed, and you should not open or edit documents in the book. Also, you should ask other users on your network to avoid access to the book, and, if necessary, you should change permissions on the book to limit access during printing. Changing permissions on the book does not prevent access to its contents through links.

Printing from Inside a Book

You can open a book and print any or all of its contents by selecting the icons and executing the desktop **Print** command. The fact that a non-book icon, such as a folder, is *inside* a book does not prevent you from selecting *it* (rather than the book that contains it) and printing it. If you select all of the book's contents at once and execute the **Print** command, the icons will print in order. Otherwise, the icons will print in the order you select them.

Print Submenus

When you execute the [Print →] [*printername* →] [Document] command, all of the printable contents of the book are printed in order. A header page is printed with any document in the book with *Header Page* set to **Yes**. You can also use the other commands on that submenu (such as **Collated Copies** and **Selected Pages**). However, the results of these commands when executed on a book may not be the results you want if you have not saved all of the documents in the book.

When you select a book and print **Collated Copies**, collated copies of *each document* — not of the whole book — are printed. You can then assemble these into a book.

When you print **Selected Pages** in a book, the correct range will print whether the page numbers are set from a fixed starting page or inherited.

If you select a book and print **Current Page**, the page with the text caret in each document in the book is printed.

For more information on printing open documents and closed icons, see the chapter *Printing*.

Scrolling between Documents

If you have one document in a book open and decide that you want to look at the next document, previous document, or any other document in the book, you can always open the new document using the standard **Open** command. You can also use the **Next** and **Previous** commands on the Page pulldown menu to scroll between documents in a book.

To scroll to the next document in a book:

✔ Use the scroll bars, component bars, Page Pulldown menu, or keyboard commands to reach the last page of the current document.

✔ Point the mouse cursor into the Page box in the Document Header and execute the **Next** command on the Page pulldown menu. You can also use the Next Page keyboard command.

If the document containing the next page is already open, its window moves to the front of the desktop. If it is not open, a stickup menu appears (Figure 16-11).
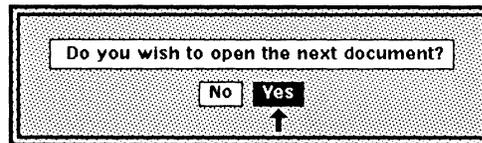
Figure 16-11. Open Next Document stickup

✔ If the Next Document stickup menu appears, select **Yes** (the default) to open the next document or **No** to remain in the current document. If you prefer to use your keyboard to respond to this stickup, you can type **y** (or **n** if you do not want to open the document).

If you select Yes, the next document in the book opens, and the document in which you were working remains open.

You can scroll to the previous document in a book by using the same technique, moving to the beginning of the document you are in and executing the **Previous** command (or its keyboard equivalent) instead of the **Next** command.

When you use the **Open** command to open a document, the document opens at the size and position it had when you last opened it. When you use **Next** or **Previous** to open a document, the document's window duplicates the size and position of the document in which you executed the command.

Only these two commands allow you to scroll between documents in a book. If you want to make sure that you *do not* scroll between documents, you can use any of the other page location commands on the Page pulldown menu (**Go To, First,** and **Last**), or you can scroll in the scroll bar or component bar. You can

also answer **No** on the Next/Previous Document stickup if you want to remain in the current document.

If the next or previous document is in a sub-book, the document window, but not the sub-book window, opens when you execute a **Next** or **Previous** command and confirm on the stickup. Leaving the sub-book closed keeps you from reaching the 16-window limit on the desktop unnecessarily. **Next** and **Previous** do not open documents in non-book directories.

Spell Checking, Search, and Replace

Commands that can cause a document to scroll do *not* cross document boundaries in a book. These include the Spelling commands, the Find commands, and the Replace commands.

Links in Books

Links to documents outside of a book can be used to create several versions of the same document without using as much storage. You can leave the original version of a document in one directory and put a link to it inside the book. A second advantage of using links is that when you edit a document that is used in several different places, links to it automatically access the updated version.

To illustrate how this could work, we provide an example here of two different books sharing the same linked document.

Assume that you have owner's manuals for two automobile models — the Widget and the Pingo — manufactured by the same company. These models are similar, but have many subtle differences that require changes throughout most of the manual. However, the engine maintenance instructions for the two vehicles are the same.

Assume that you have a cabinet called *Manuals* in which you have created books for Widget and Pingo (Figure 16-12).
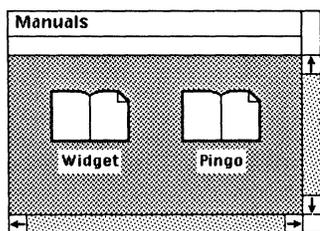
Figure 16-12. Books in a directory window

You write the Pingo manual first, and then the Widget manual (except for the engine maintenance section). Then you [Copy →] [Link to] the *engine* document in the *Pingo* book and paste the link into the *Widget* book (Figure 16-13).
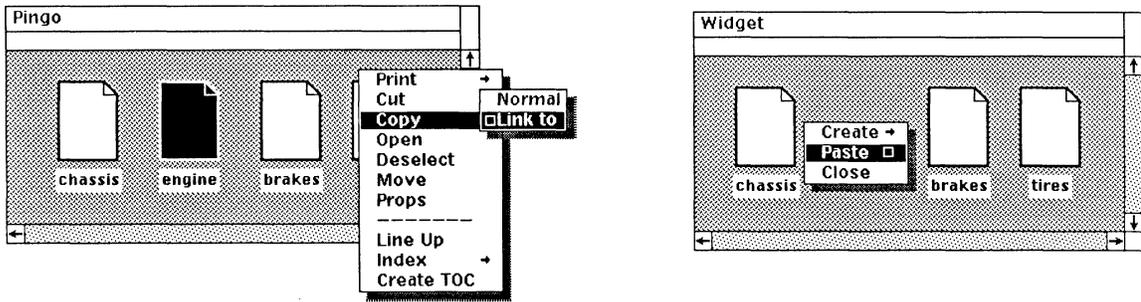
Figure 16-13. Pasting a link into a book

Because the chassis of the Pingo requires little maintenance, the *chassis* document is only three pages long. However, the *chassis* document for the Widget is eight pages long. Even though the *engine* document in the *Widget* book is a link to the *engine* document in *Pingo*, the *Widget* page numbers reflect its longer *chassis* document (Figure 16-14).
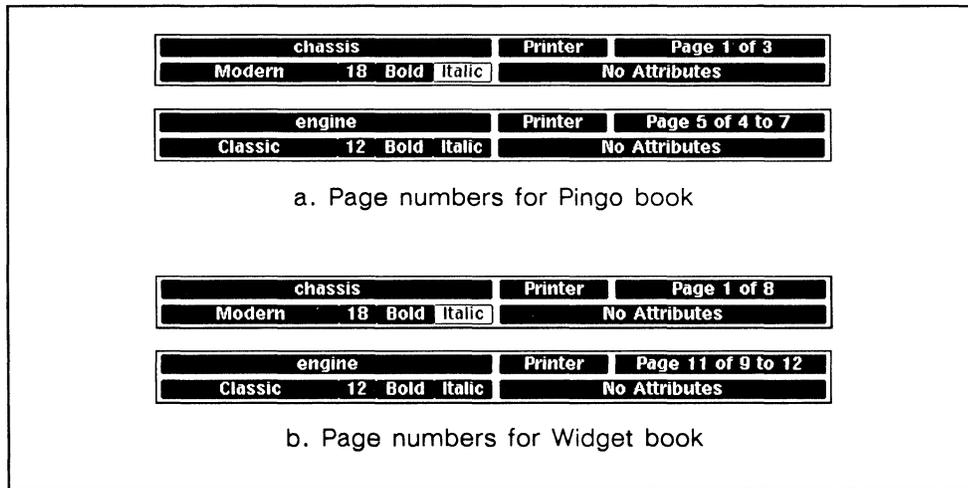
Figure 16-14. Page numbering for links in books

Links and linked objects are described in detail in the chapter *Desktop Management*.

Chapter 17

Indexes and Tables of Contents

In a book, you can create **indexes** and **tables of contents (TOCs)** for one or more documents. Although they have important differences, the indexing and table of contents creation features of the publishing software share several basics.

- The **Index** and **Create TOC** commands are available on the Icon Selected menu in a book window.

- When you execute the **Index** or **Create TOC** commands, you create an ASCII document in the window.

- You can create an index or TOC document that draws information from any number of selected source documents, so long as those documents are in the same book.

- You can create an index or TOC for open documents. For a *closed* document, index and TOC information is drawn from a summary file that reflects what was in the document the last time you saved it. For an *open* document, the *current* TOC and index information is used, even if it includes unsaved changes. An index or TOC document may be inaccurate if it includes information from an edited open document that you close without saving.

- The entries in an index or TOC document use the page numbers, prefixes, and page number style from their source documents.

The **Index** command creates an alphabetically ordered index document from *headings* that you create specifically for that purpose. The **Create TOC** command creates a TOC document from designated *components*, listing the components in the order they appear in the source documents.

The rest of this chapter is divided into three major sections. *Indexes* describes the steps necessary to create an index document and tailor its contents to your needs. *Tables of Contents* provides similar information for TOC documents. The final section *Managing Index and TOC Documents* describes some considerations and techniques appropriate for both types of documents.

Indexes

The elements that make up the indexing feature are:

- indexing **tokens** that mark a text location
- the **Index property sheets** for each token, in which you type the main heading and any subheadings for that location
- the **book directory**, where the commands that create the index document are available
- the **index document**

Figure 17-1 shows part of an index document and the terms used to describe it.
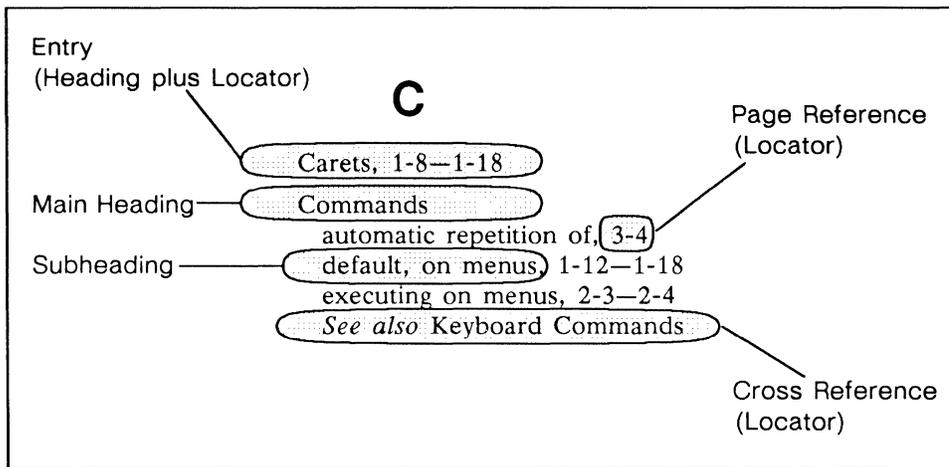
Figure 17-1. Parts of an index document

Headings are the terms or phrases used to describe the information someone might want to look up in an index. They may be the **main headings** that form the top level of organization in an index or more specific **subheadings**.

Locators tell you where to find the information described in the headings. A **page reference** is a locator that gives you a page or range of pages where you can find the specified information. A **cross reference** is a locator consisting of *See* or *See also* plus a heading or category of headings where information on the subject you looked up can be found.

The combination of a heading with a locator creates an **entry**.

We refer to a document from which you make an index as a **source document** and a document produced by an **Index** command as an **index document**.

Index Tokens

The index token marks a location for which you want an index entry. These tokens have width on the screen, but they neither print nor displace other characters in a printed document.

To create an index token in a document:

    📌 Position the text caret at the location where you want an index reference to *begin*.

    📌 With no text selected, execute `Create →` `Index` .

        A small gray index token appears at the caret (see Figure 17-2). Text may move slightly to the right or left to make room for the token. The token is selected and an Index property sheet opens in the upper right corner of your screen.

Cutting, Copying, and Pasting Tokens

To delete an index token, you must select it and then either **Cut** it or delete it using one of the text deletion keyboard commands.

You can copy or cut an existing token and paste it elsewhere in the same or a different document. The token you paste has all of the properties it had in its previous location. Even if you must change some of the properties, copying tokens can be more efficient than creating them when you want to create a new entry that is similar to an existing one.

Displaying Index Tokens

So that you can move, copy, edit, and delete index entries, index tokens are visible on your display screen by default. You can make all of the index tokens in a document invisible using the Misc Show submenu on the Text Location menu. Like other show commands, `Show →` `Index` toggles visibility on and off.

When you save a document, the state of token visibility is saved with it.
Figure 17-2 shows a component with two index tokens visible and the same component with tokens invisible. Notice that the space compression caused by the tokens disappears when they disappear.

KeyboardCommands∷	Keyboard Commands

Figure 17-2. Toggling the Show Index command

Searching for Index Tokens

You can search for index tokens using the Find Index submenu of the Text Location popup menu (Figure 17-3).
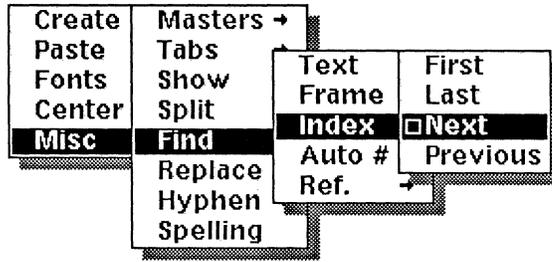
Figure 17-3. Find Index submenu on Text Location popup menu

When index tokens are visible, any token found using the **Find Index** commands is selected and shown in reverse video. The **Find Index** command also works when you have made tokens invisible. In this case, the text caret moves to the position of the token and selects the token. Since the token is invisible and takes up no space, it does not appear in reverse video, but this does not prevent you from executing the same commands you can use when it is visible.

Index Tokens in Microdocuments

You can insert an index token in a microdocument, and that token's information *will* be included in the index document when you execute an Index command. The *Range* for a token in a microdocument cannot extend past the end of the microdocument. If you insert an index token in a shared content frame, the *Range* for that token refers only to the first instance of that frame.

For more information on microdocuments, see the chapter *Text in Frames.* For more information on shared content frames, see the chapter *Text-anchored Frames.*

Index Property Sheets

An Index property sheet (Figure 17-4) opens when you create a new index token. It also opens when you select a token and execute the **Props** command.
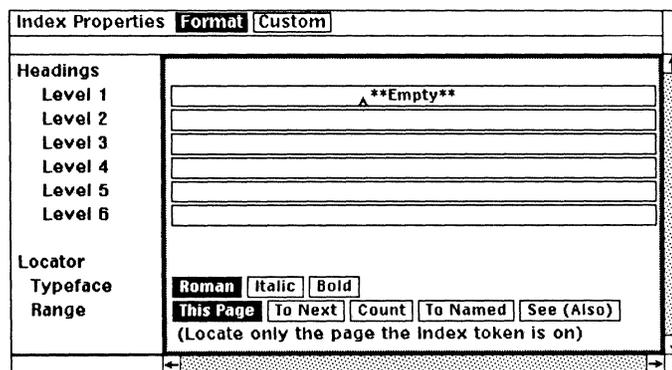
Figure 17-4. Format sheet of default Index property sheet

If the document in which you created the token is the only window open on the desktop, you can open up to 15 Index property sheets simultaneously. This allows you to compare the contents of the sheets and make or modify entries in any one of them without having to close the others. You can also edit the document with Index property sheets open. Any change you make and **Apply** on the Index property sheet becomes available to the next index document you make.

The **Format sheet** of the Index property sheet allows you to create headings, determine the range of pages that will be shown, cross reference an entry, and change the typeface of the page reference. The **Custom sheet** of the Index property sheet allows you to change the sorting order of an entry and to send the entry to a document other than the default document named *Index*.

Inserting Headings

On the Format sheet, there are six rows of *Headings* boxes labeled *Level 1* through *Level 6* (see Figure 17-4). The default property sheet has a *Level 1* heading of **Empty** and no subheadings. If you simply close the property sheet without modification and create an index document, this default heading and the page number the token is on appear as an entry in the index document.

The text you type in *Level 1* represents the main heading for an index entry. In the index document, it always precedes the text in other levels for a given token.

Subheadings

In a manually produced index, you might separate a main heading from a subheading with a comma. On the Index property sheet, such headings should be typed in different Level boxes, without commas. For example, where you would type *Tokens, Index* in a manual index, you should type *Tokens* in the Level 1 box and *Index* in the Level 2 box. This allows the software to choose the best way to format the index document.

Deleting Headings

You can delete a heading on an Index property sheet by positioning the text caret in the box for that heading, using any text deletion keyboard command, and confirming the new (empty) heading with a RETURN or TAB. If you eliminate a level between two remaining levels — for example, if you delete the text in Level 2 while leaving text in both Level 1 and Level 3 — all of the levels below the deleted level move up when you apply the change.

ASCII Markup in Headings

The strings in the Heading boxes go directly to the ASCII loader when you create an index document. Because of this, you can include ASCII markup in any Heading box to create special fonts in some headings.

The publishing software knows ASCII markup format and does not include the markup in the default sorting of entries. However, you should be careful about using open angle brackets (<) in headings since they are used to bracket ASCII markup. If you want a '<' to print in the index, enter it as '<<'.

Interleaf ASCII markup format is discussed in the manual *File Formats*.

Leading and Trailing Spaces

Extra spaces in an index heading can produce uneven layout in the index document and confusion between main and subheadings. To prevent this, any *soft* spaces that you place at the beginning or the end of any heading are stripped from the heading when you execute an **Apply** command. *Hard* spaces are accepted in Index property sheets.

Creating an Index Document

To produce a basic index document, all you need do is create tokens and fill in headings on the property sheets for these tokens. This allows you to produce an index document named *Index* with entries, which all have single-page references. The features that allow you to modify this basic document and its entries are described later in this chapter.

Like other book functions, the Index commands are only effective on document and book icons — folders, drawers, and other objects inside a book are not indexed. An icon can be selected and its contents indexed whether the icon is open or closed. If an icon's window is open, the index document includes any changes you have made in the open window.

Existing index documents in the book where you execute an **Index** command are overwritten by the new index documents of the same name. Since index documents do not have index tokens in them, you can select all of a book's contents for indexing without complication.

To create an index document:

✔ Select the icons you want indexed and execute the ⌷Index →⌷ ⌷Normal⌷ command on the Icon Selected menu.

A document labeled Index appears in the book. This document is stored in ASCII format. If you have other index document names on any Index property sheets, those documents are created at the same time. When first created, index documents are in animation state.

✔ Move the index documents to the position you want them in the book.

If you position index documents ahead of other documents, they could change the page numbering throughout the book.

Master Indexes

The Master index feature is designed to produce a combined index for a multi-volume set in which each volume also has its own index (produced using the **Normal** command). When you execute the ⌊Index →⌋ ⌊Master⌋ command, the software creates an index with the same basic characteristics as a **Normal** index, except that each entry with a page number locator also includes the name of the selected icon from which the entry was extracted.

The names picked up by the **Master** command are only those at the top level of the book you execute the command in. For example, *Vol.1* becomes part of some entries in the index document, but the names of documents inside *Vol.1* do not. No icon names are included for *See (Also)* references in a Master index.

Figure 17-5 shows a book called *RefManual*. It also shows index entries from a single Master index created for both volumes (left), and the same entries as they would appear in two separate Normal indexes (right), one created for *Vol.1* and the other for *Vol.2*.

From RefManual Index (Master)

I

Icons
    creating: *Vol.1* 2-29—2-33
    on the clipboard: *Vol.1* 2-19—2-33
    types of: *Vol.1* *1-9—1-18*

Indentation, component: *Vol.2* 4-23

Interrupt Stickup
    commands on: *Vol.1* 2-5
    crash recovery and: *Vol.1* 2-5
    how to call up: *Vol.1* 2-4
    uses of: *Vol.1* 1-17; *Vol.2* 4-5

Italic Command, text and: *Vol.2* 4-31—4-50

From Volume 1 Index (Normal)

I

Icons
    creating, 2-29—2-33
    on the clipboard, 2-19—2-33
    types of, *1-9—1-18*

Interrupt Stickup
    commands on, 2-5
    crash recovery and, 2-5
    how to call up, 2-4
    uses of, 1-17

From Volume 2 Index (Normal)

I

Indentation, component, 4-23

Interrupt Stickup, uses of, 4-5

Italic Command, text and, 4-31—4-50

Figure 17-5. Normal and Master indexes from identical input

Determining Locator Style

Typeface

The *Typeface* row on the Index property sheet controls the style of the page or cross reference for an entry, allowing you to make it bold, italic, or roman. You can use this feature to highlight certain entries — for example, using italic page numbers to indicate illustrations. This has no effect on the appearance of the *headings* in an index or on the words *See* and *See also*, which are always italic. Figure 17-6 shows the effect of changes in the *Typeface* row on an index entry.
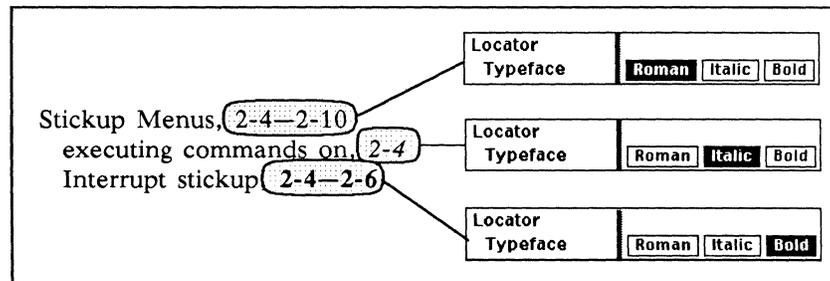
Figure 17-6. Typeface changes for entries in a sample index

If you globally apply a change to *Typeface* on a property sheet, that sheet and any other with identical headings at all levels is affected by the change.

Document-wide Page Style

Several properties that affect all the page references in an index document are determined on the Page property sheet for the documents being indexed. If a prefix is specified in the *Page # Prefix* box, this prefix is included in the page references next to entries in the index document. For example, if you have entered 3. in the Prefix box on a Page property sheet, the page numbers for that document's index entries appear in the form **3.4, 3.8,** etc.

If you manually insert a page number prefix in a source document — that is, if you type it in the footer frame rather than the Page property sheet — page references for index entries from this document do not include the prefix.

The other Page property that affects the appearance of index entries is the *Page # Style* line, which determines whether the numbers are displayed in Arabic, Uppercase Roman, or Lowercase Roman numerals. If you have selected Uppercase Roman as the *Page # Style* in a document, the page numbers for that document's index entries appear in the form *I, III, XII,* etc.

Figure 17-7 shows how the Page property sheet of the source document affects the appearance of page numbers in an index document.

Figure 17-7. Page number attributes on Page property sheet

Determining Range

Range determines the locator — either the range of pages or a cross reference — that is associated with an entry. A comment line below the Range boxes summarizes the results of each choice when it is selected.

This Page

This Page is the default range. The entry for this token in the index document lists only the number of the page on which the token appears.

To Next

To Next means that range of pages listed for this token begins at the token and ends immediately before the next component. If a token is on one page and the component it is in extends to a succeeding page, its entry in the index lists two page numbers.

Count

Count allows you to specify how many components (starting with the one the token is in) are included in the range for an entry. If you select **Count**, the property sheet adds a *Count* row with a box containing the default number **2** (Figure 17-8). You can enter any number up to 100 in this box, and the range of *pages* covered by these components becomes the locator for the entry.

Figure 17-8. "Count" selected on Index property sheet

To Named

To Named is useful for referring to an entire section up to the next subheading. When you choose this option, a line beginning with *End At* and including a box appears at the bottom of the sheet (Figure 17-9). The default name in the box is

the name of the component containing the index token, but you can type the name of any component there.

Locator	
Typeface	`Roman` `Italic` `Bold`
Range	`This Page` `To Next` `Count` `To Named` `See (Also)`
End At	`section` (Locate up to the given component)

Figure 17-9. "To Named" selected on Index property sheet

When you select **To Named,** the *Range* includes all page numbers from the token up to the next instance of the component named in the *End At* box. If the named component begins a new page, this page number is *not* included in the reference.

If no component by the name you insert follows the token, all pages from the token to the end of the document are included in the *Range* for the entry.

See (Also)

If you select **See (Also),** the property sheet provides a box in which you can type a cross reference. The software automatically converts a *See* entry to a *See also* entry when page number references or deeper subheadings also appear under the main heading (Figure 17-10). This allows you to have a page number and a *See also* reference for one index entry.

Headings	
Level 1	`Commands`

Locator	
Typeface	`Roman` `Italic` `Bold`
Range	`This Page` `To Next` `Count` `To Named` `See (Also)`
See (Also)	▲Menu Commands

This is how the entry above is formatted when it is the only one with "Commands" in Level 1.

Commands. *See* Menu Commands

If you add an entry with "Commands" in Level 1 and that entry has a page reference, *See* changes to *See also.*

Commands, 10 *See also* Menu Commands

Figure 17-10. "See (Also)" selected on Index property sheet

Because *See (Also)* tokens are not associated with page numbers, they can be difficult to locate in a document. So that you can more easily find and delete or edit *See (Also)* tokens, you should develop conventions for placing them. One way to do this is to keep them in a separate document at the end of the book.

If you delete an entry be sure to delete any cross references to that entry.

Choosing the Best Range

Your choice of page range should be guided by the stability of the document you want to reference. The most stable ranges are the most limited — **This Page** and **To Next**. Whenever you can use one of these ranges, they are preferable.

For larger ranges, if you are likely to make major changes *within* a section you have marked with an index token but are unlikely to add, change the name of, or delete subheadings, **To Named** is probably your best choice. However, keep in mind that if you change the name of the named component, that name does not automatically change on Index property sheets.

If you are certain that you have said all there is to say about a subject and will not be altering the section describing it, **Count** might be your best choice. The higher the number in the count box, the more likely that editing you do will make **Count** ranges incorrect.

If you globally apply a change to *Range*, this change affects only the token in whose property sheet you made the change and any tokens with identical headings at all levels.

Range Limits

The range for a single index token will not extend from one document to another. For example, if you create an index token in the last component of a document and then set the *Range* as **Count 50**, the real range for that token will be only the page it is on. It will not count ahead 50 components into the next document in a book.

The same limitation is true if you select **To Named** and insert the name of a component the next instance of which is in the next document — the range for that token extends only to the end of the document containing the token.

Global Apply and Index Properties

Like many other property sheets in the publishing software, the Index property sheet allows you to apply a property change to an individual object or to all objects of the same name, and to unify the properties of all objects with the same name. For index tokens, the object names are the Index Document name and the *Level 1* through *Level 6* headings.

The effect of the **Apply** and **Global Apply** commands on index tokens varies depending upon which of a token's properties you have changed, but certain rules govern any globally applied change in an Index property sheet.

- **Global Apply** works for all matching tokens in *one* document. It does not cross document boundaries in the book. To apply a change to all matching tokens in a book, you must globally apply the change in *each* document.

- **Global Apply** in one token affects only tokens that have the same *Index Document* name. All of the general rules about the results of an index property change assume that all tokens in a document share the same index document name.

Unifying Index Properties

When you **Unify** on an Index property sheet, you affect only those tokens whose Headings and Index Document Name are already the same. Therefore, **Unify** only unifies the *Typeface* and *Range* properties for these tokens.

If you **Unify** or **Global Apply** an Index property change when other property sheets affected by the change are open, all affected sheets except the one the command was issued in are automatically closed.

Globally Applying Heading Changes

If you change any heading, **Global Apply** applies that change to all tokens with the same headings *at and above the level where you made the change.* Figure 17-11 shows an example of this principle.

For these index entries . . .

New Technologies electronic publishing market for, 9—35	Office Automation electronic publishing market for, 9—35 suppliers, 36—45

if you make this Level 2 heading change and Globally Apply the change . . .

Headings			Headings	
Level 1	Office Automation		Level 1	Office Automation
Level 2	ᴀelectronic publishing		Level 2	**technical publishing**
Level 3	market for		Level 3	market for

only entries with the same Level 1 and 2 headings as the changed token are changed.

New Technologies electronic publishing market for, 9—35	Office Automation technical publishing market for, 9—35 suppliers, 36—45

Figure 17-11. Global Apply and heading changes

The Custom Sheet

The Custom sheet of the Index property sheet (Figure 17-12) allows you to create separate, named index documents and to insert sort strings that differ from the Heading text for Levels 1 through 6 on the Format sheet.

Index Properties	Format	Custom
Index Document	Index	
Sort Strings		
Level 1	Document Windows	
Level 2		
Level 3		
Level 4		
Level 5		
Level 6		
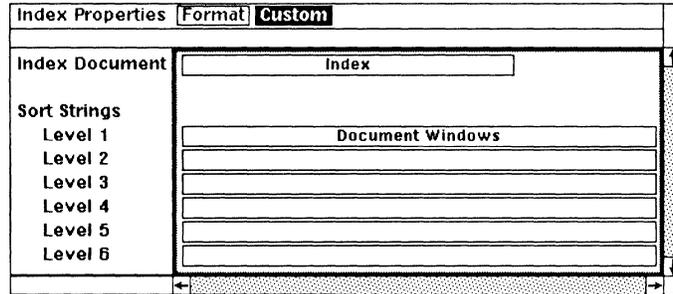
Figure 17-12. Custom sheet of Index property sheet

Index Document Name

By default, the name of the index document you produce is *Index*. However, you can change the name of the index document for any token in a document so that when you execute the **Index** command you create separate indexes for different kinds of information. You can also index certain information both in a main index and in a specialized index.

For example, you might want an index of the people mentioned in a book. For each token you create that refers to a person, you could change the Index Document Name to *People* on the Custom sheet. If you then decided that you want these people referenced in both the main and the *People* index, you could copy each appropriate token, open its property sheet, and change its Index Document name to *Index*.

Do not use the name of an existing document in the *Index Document* box. Do not name other documents in a book *Index*. If index document names and source document names coincide, you could lose your source document when you execute the **Index** command. For more information, see *Naming Index and TOC Documents* later in this chapter.

If you globally apply a change in the name of an index document, all tokens with the old document name acquire the new document name, regardless of how their other properties differ.

Customizing Sort Strings

In an index document, headings at each level are first sorted by initial character into the following classes: hard spaces, symbols, numbers, and letters. Letters are sorted in alphabetical order with word precedence. Word precedence means that an index heading consisting of two words (such as *Document Windows*) comes before a single word that begins with the first word (such as *Documentation*).

Within a group of entries beginning with the same letter, all entries beginning with a capital come before all entries beginning with a lower case of the same letter.

The default sort string for any text in Levels 1 through 6 is the text itself minus any ASCII markup. If you want to ignore word breaks in sorting or if you have other special sorting requirements, you can enter custom sort strings on the Custom sheet. For example, when you change *Document Windows* to *Documentwindows* in the Sort String field, the entry for *Documentation* comes before the entry for *Document Window*. Sort string changes do not affect the spelling or word division of the actual headings.

Sort String Consistency

Changes in sort strings follow the same general pattern as changes in headings, except that any change to a sort string behaves like a **Global Apply**, even if you do not use the **Global Apply** command. This prevents inconsistent sorting of the same heading.

Also, when you *paste* an index token into a document and the token's Index Document and Heading names are the same as those of an already resident token, the Sort Strings of the incoming token are unified to those of the resident token.

The Index Document

The default index document has two-columns per page, no headers or footers, and an 18-point Modern bold initial at the beginning of each letter of the alphabet represented in the index. Entries are in 10-point Classic type and have their headings and page numbers separated by a comma and a space. Figure 17-13 shows a page from a sample index document.

Figure 17-13. Page from a sample index document

Index	Printer	Page 1 of 4
Modern	18 **Bold** Italic	No Attributes

IndexLet
Level1
Level1
IndexLet
Level1
Level2
Level2
Level1
Level1
IndexLet
Level1
Level1
Level2
Level2
Level2
Level1
Level2
Level2
Level2
Level3
Level1
Level2
Level2
Level2
Level2
Level2
Level1
Level1
Level2
Level2
Level1
Level2
Level2
Level1
Level1
Level1
Level1

A

Active Window, 1-13

Alignment, keyboard commands and, 3-11—3-12

B

BACK SPACE Key
   CTRL key alternative, 3-4
   function keys and, 3-2
Back Command, 2-14—2-15
Bullets, keystroke for obtaining, 3-13

C

Carets, *1-8*
Clipboard, 2-18—2-22
   'from' icons on, 2-19—2-20
   overwriting on, 2-20
   purging objects from, 2-21—2-22
Closing
   Desktop, 2-2
   directories, 2-13
   documents, 2-12—2-13
   and saving, 2-13
Commands
   See also Specific commands
   automatic repetition of, 3-4
   default, on menus, 1-12—1-13
   executing on menus, 2-3—2-5
   keyboard, 3-6—3-17
Component Bars, *1-14*
Components
   creating, 2-29—2-30
   Global Apply and, 2-27—2-28
Copy Command
   basic procedure, 2-18
   inside documents, 2-19—2-20
Crash Recovery, 2-5
Creating, basic procedures, 2-29—2-31
CTRL Key Sequences, 3-4—3-5
Cursor Shapes, 1-7—1-8

Cut Command
   basic procedure, 2-19
   inside documents, 2-19—2-20

D

Default Commands, fundamentals of, 1-12—1-13
Desktop
   opening and closing, 2-2
   organization and performance of, *1-2*
Devices, icons for, *1-9*
Directories
   closing, 2-13
   creating, 2-29
   icons for, *1-9*
   opening, 2-12
Document, creating, 2-29
Document Name Token, 3-14
Document Windows, *1-14—1-15*
Documentation, on line, 1-5
Documents
   cancel opening, 2-12
   closing, 2-12—2-13
   icons for, *1-9*
   opening, 2-12
   saving, 2-12
   saving and closing, 2-13
   structured, 1-3

E

Error Handling, in property sheets, 2-28
Error Messages, in status line, 1-16
ESC Key, function keys and, 3-2

F

Fonts
   decimal character values and, 3-12
   keyboard commands and, 3-10—3-11
   text caret and, 1-8
Fractions, keyboard commands and, 3-15—3-16

Figure 17-13. Page from a sample index document

Heading Style

If the only instance of a particular main heading includes one subheading, the software separates the text in Level 1 from the text in Level 2 by a comma and a space but does not put the Level 2 subheading on a new line, as Figure 17-14 shows. The same rule applies to any two adjacent subheading levels. When a second token has identical headings at all levels but has a different page reference, only one set of headings appears in the index document for the two tokens and their page references are separated by commas.
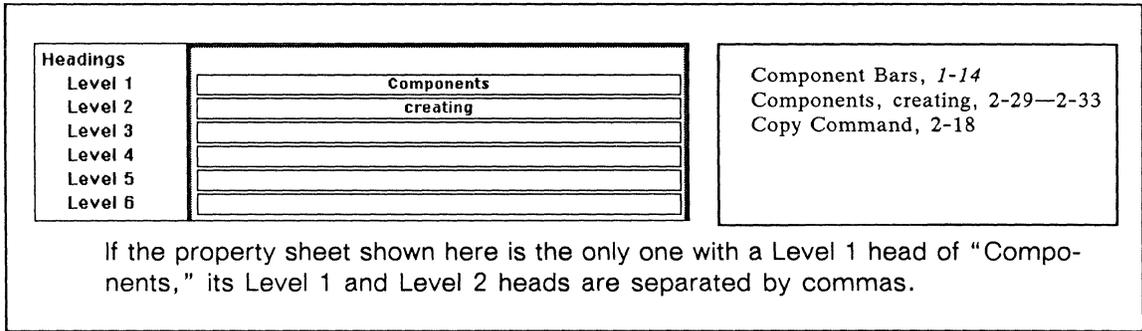
Figure 17-14. Heading levels separated by commas

If a main heading is used more than once and has different subheadings (or a subhead for one token and no subhead for another), the subheadings are indented and placed in their own components (Figure 17-15).
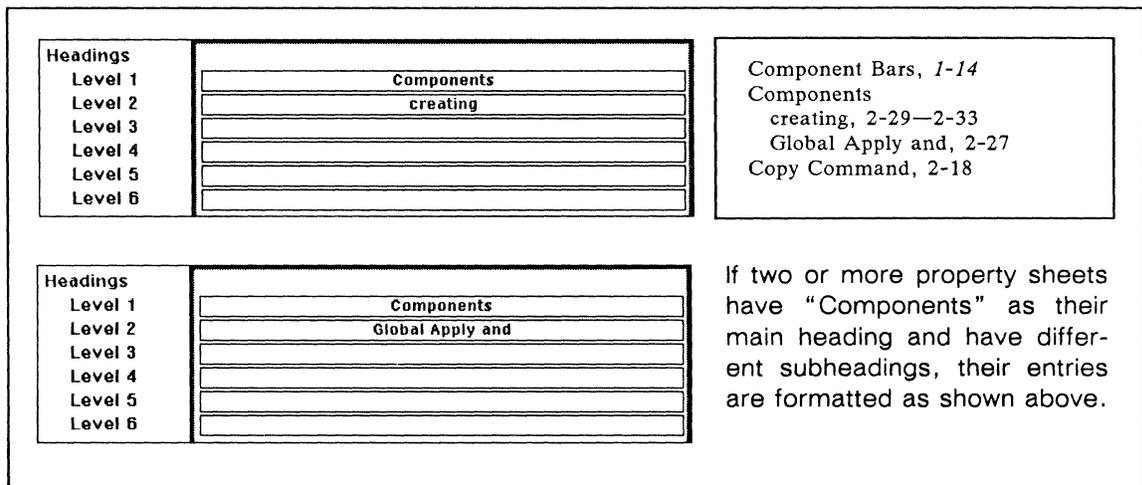
Figure 17-15. Heading levels on different lines

In a Master index, each heading that includes a page reference is followed by a colon, two soft spaces, the icon name (in italics), a space, and then the page references. If more than one icon is referenced for a single entry, a semicolon and two soft spaces precede the second and any additional icon names.

Entry and Range Merging

If you have two or more tokens with identical headings, they become one entry in the index document and their locators are combined. If their page ranges overlap or one is entirely enclosed by the other, these ranges merge in the index document. For example, if the range for one token extended from the middle of page 2 to the middle of page 3 and the range for another token with identical headings extended from the beginning of page 3 to the end of page 3, the range would display as **2—3** in the index document. However, if the same two tokens had ranges of **This Page,** the range for that entry would display as **2, 3.**

Updating and Correcting Your Index

Once you have created an index, you can correct or update it easily. Although you can edit the index document the same way you would any other document, it is much more useful to update or correct the index using tokens and the Index property sheets. After you have made modifications, execute an **Index** command on the Icon Selected popup menu in the book again. Your changes are reflected in a new index.

You may want to change the style of the index document once you are satisfied with its content.

- To eliminate the initial capitals that begin each alphabetical group, select and cut all of the components named *IndexLet*.
- To change the standard punctuation, use the Find and Replace commands.
- To change overall document style, create an index template document into which you can copy the automatically produced index. This template can combine changes in typeface, number and width of columns, subheading indentation, and header/footer content.
- Remove any extraneous entries. If you accidentally leave the default heading on a property sheet or begin any other Level 1 heading with a symbol, those headings appears at the beginning of the index document with a *Symbols* section heading (Figure 17-16).

```
                  Symbols
          % change, 2-26
          **Empty**, 1-1

                     A
          ASCII files, 14-6
```
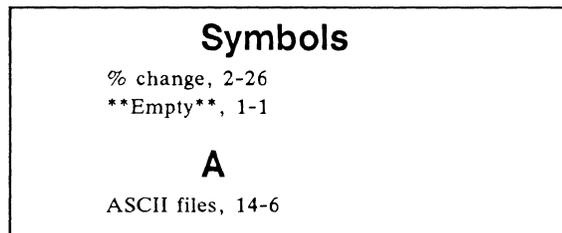
Figure 17-16. Sample index with section heading for symbols

Sample Index

If you want to practice creating an index, copy the document *IndexPractice*, in the *Reference* folder of the *Documentation* drawer in the *System* cabinet. Create a new book, open the book, and paste the copy of *IndexPractice* there.

IndexPractice already contains index tokens, but in the steps below, you will add tokens of your own before creating the index document. This document is a good tool for practicing the variations on standard index document creation, such as using alternate index document names, special sort strings, and different ranges.

To create a sample index:

➤ **Open** the copy of *IndexPractice*.

➤ With the editing caret just after the end of the word *Type-faces* in the title of the document, execute the `Create →` `Index` command.

▣ On the first line of the Index property sheet that opens, type and enter **Samples**.

➤ **Apply** the properties and **Close** the property sheet.

➤ Select the token that you inserted after *Typefaces*, **Copy** it and **Paste** it before the first word in the first paragraph.
Your document contains two tokens as shown in Figure 17-17.

Chapter 8

Typography and Typefaces⌐   ⟵ Index token

Index
token⟶ ⌐If you are a new user of the Interleaf publishing software, this may be your first exposure to multiple typefaces and type sizes, proportionally spaced type, and varied line and word spacing. These new capabilities open up a tremendous number of possibilities for producing more attractive, more readable documents. They also require choices that are not necessary when you are using a typewriter or a daisywheel printer.

Figure 17-17. Index tokens in a sample document

➤ With your second token still selected, execute the **Props** command to open the new property sheet.

▣ In the row labeled *Level 2*, type and enter the words **of in- dexing**. **Close** the property sheet.

➤ **Copy** the second token and **Paste** it on page 2 next to the heading, *Type Specification*.

➤ Execute the **Props** command on the Text Selected menu again.

▣ Type and enter **of type specification** on Level 2 of the property sheet. Change the *Range* to **Count** and enter **15** in the Count box.

➤ **Apply** your changes.

✔ Select the document icon in the book and execute the `Index →` `Normal` command.

✔ **Move** the *Index* document to any location in the book and then **Open** it. Note the entries under *Samples*.

Tables of Contents

The automatic table of contents generation function lets you assemble a TOC document from designated components inside one or more documents. The key elements of the table of contents feature are

- the **Component Custom property sheet**, where you can flag a component for inclusion in a TOC document
- the **book directory**, where the command that creates the table of contents document is available
- the **TOC document**

When you execute the **Create TOC** command, you produce one or more ASCII documents containing tables of contents for the documents or books selected.

Designating the Contents of a TOC

Before creating a table of contents, you should consider which components will be most useful to you. In general, titles, subheadings, and other short components are easiest to use, since they are already in a form appropriate for a table of contents. However, you can include any component or combination of components in a table of contents.

You include a component in a table of contents by modifying its Component Custom property sheet. You actually determine two things by your entry in the *TOC Doc Name* line: whether the component becomes part of any table of contents, and if so, what TOC document it appears in.

Do not use the name of an existing document in the *TOC Doc Name* box. If TOC document and source document names coincide, you could lose your source document when you execute the **Create TOC** command. See the section *Naming Index and TOC Documents* for more information.

To include a component in a TOC document:

✔ **Open** the document.

✔ Open the **Props** sheet for one of the components you want included in the table of contents and select the **Custom** property sheet (Figure 17-18) if it is not already displayed.

Component Properties:	Format	Tab	Page	**Custom**	

Hyphenation	Off 1 2 3 4 **Normal** 6 7 8 9 Full	
Autonumber		
Name	[ ]	
Level	[1]	
TOC		
Doc Name	[ ]	
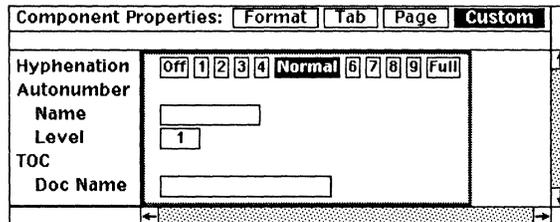
Figure 17-18. Custom sheet of Component property sheet

✔ In the *TOC Doc Name* field, type and enter the name of the table of contents you want that component to appear in.

For example, if you want only one table of contents, you could use toc *as the name in this field.*

✔ If you want all components of this name to appear in the table of contents, execute the **Global Apply** command. If you want only this occurrence of the component to become part of the TOC, **Apply** the change.

✔ Repeat this procedure for each component you want listed in the table of contents.

✔ If you want your TOC document to include entries from more than one document, repeat this procedure for each document.

If you enter a name in the *TOC Doc Name* box for a component, the component and the number of the page it is on become part of the named table of contents document when you execute the **Create TOC** command. If you globally apply after entering this name, every component of that name in the document is picked up in the table of contents.

Because the standard TOC document format uses a specified number of tabs, tabs in the components you include in a TOC can disrupt the appearance of the TOC document.

If you paste a component from one document to another, make sure that the name in the *TOC Doc Name* field is appropriate for the new document.

Components in Microdocuments

Any microdocument component with a name in the *TOC Doc Name* box of its property sheet will become part of a TOC document when you execute **Create TOC**. If you put a *TOC Doc Name* in the property sheet of a component in a shared content frame, the only entry that will appear for that component is the entry for the first instance of the frame.

For more information on microdocuments, see the chapter *Text in Frames*.

Creating a Table of Contents

You can create as many different tables of contents as you like. For example, if you want one table of contents for titles and subheadings and one for illustrations, you can enter one name in the *TOC Doc Name* field on the Component Custom property sheets for *title* and *subheading* and a different document name on the *illustration* Custom sheet. The table of contents at the front of this manual was made by combining three TOC documents: one for chapter titles and subheadings, one for illustration captions, and a one for procedures (or methods).

Existing TOC documents in the book where you execute the **Create TOC** command are overwritten by the new TOC documents of the same name. When you do want to update your TOC documents you can select all of a book's contents without complication, since TOC documents do not have an entry in the *TOC Doc Name* field in their Component Custom property sheets and therefor will not be affected by the **Create TOC** command.

To create a TOC document:

✔ If the documents for which you want to create a table of contents are not in a book, **Cut** and **Paste** them into one.

✔ **Select** the documents and execute **Create TOC**.

One document appears for each of the TOC Doc Names *you have used in your source documents. When first created, TOC documents are in animation state.*

✔ Move the TOC documents to the position you want them in the book.

If you position TOC documents ahead of other documents, page numbering throughout the book could change.

TOC Document Style

Figure 17-19 shows a page from a sample TOC document created from the first three chapters of this manual. If you compare this to the same listings in the table of contents at the front of this manual, you see the results of putting the TOC document into a template.

Figure 17-19. Page from a TOC document

Component Order

Inside the TOC document, the components you designated appear in the same order they appeared in the source document. If you created a TOC for more than one document, the TOC entries are listed in the order the documents appeared in the book.

Component Properties and Content

Normally, the letters "TOC" are appended to component names when they are in the TOC document. However, a component name (including 'TOC') can be no more than nine characters long, and is shortened from the right when necessary. For example, if you had a component called *paragraph* in your source document, it still would be called *paragraph* (not *paragraphTOC*) in the TOC document.

With the exception of their names and the bold setting, all components in a TOC document have the same properties. The font of components in the TOC document is converted to 10-point Classic roman type. Headings are separated from page numbers by two tabs, one of which is dotted. If there was a page prefix in the original document, the TOC document picks it up in each of the entries.

If more than one component name appears in the TOC document, the first component, and any other components with the same name as the first component, are displayed in bold. If, for example, the first component in a TOC document with different components in it is called *titleTOC*, that, and any other instance of *titleTOC* in the document appear in bold.

Page Properties

The TOC document has blank headers and footers. It is 8.5 inches wide and 11 inches high, with the following margins: top — 1 inch; bottom — 1.1 inches; left — 3 inches; and right — 1.5 inches.

The TOC document has no page numbers or other content in its header/footer frames, but it does have a **Lowercase Roman** *Page # Style* and **Inherit** on.

Editing the TOC Document

After you create a TOC document, you may want to edit it. If the editing involves addition or deletion of a component, you should also make these changes by adding or deleting the *TOC Doc Name* for the components in question so that the TOC document is correct the next time you execute **Create TOC**.

As with index documents, you may want to use a template document to alter page and component properties to your specifications.

If you want to change the default punctuation, you can use the Find and Replace commands to insert your changes throughout a TOC document.

As long as the *Page # Style* of the TOC document differs from that used in other documents within the book, neither will inherit page numbers from the other. If you make the styles the same, TOC documents may affect and be affected by the page numbers of the documents on either side of it. To make sure that the TOC document does not change the page numbering of other documents, move it to the end of a book. Other aspects of page numbering in a book are discussed in the chapter *Large Documents in Books*.

Eliminating Extra Tabs

In some cases, you will want to make changes in the tabs produced by the software. If you need only one tab in your TOC entries instead of the two inserted by the software, you can either remove the extra tab in each line or create a template that effectively eliminates one tab. By reducing the difference between tabs to a very small amount — .01 inch, for example — you can save yourself the effort of removing the extra tab in each line. You may also want to change the right tabs produced by the software to left tabs. This, too, can be made simple through the use of left tabs in your TOC template.

See the chapter *Document Templates* for more information on template design and use.

Sample TOC

If you want to practice creating a TOC document, copy the document *IndexPractice*, in the *Reference* folder of the *Documentation* drawer in the *System* cabinet. Create a new book, open the book, and paste the copy of *IndexPractice* there. If you already did this to create a sample index, you can use the same copy and book.

IndexPractice already contains components marked for TOC creation. You can select it and execute the **Create TOC** command to see how the TOC document is created, examine its Component Custom property sheets to see how they are marked for TOC pickup, and change the *TOC Doc Name* on some of the property sheets to create additional TOC documents.

Managing Index and TOC Documents

Executing an **Index** or **Create TOC** command is similar to saving a document. For example, when you execute the **Index** command, any existing index document becomes a *backup version* and the new index document becomes the *document version*. Therefore, if you discover that you really wanted the older version of an index document, you can use the **Revert to Backup** command on the Document Name pulldown menu.

Naming Index and TOC Documents

The default name for an index document is *Index*. There is no default name for TOC documents. In creating TOC document names and special index names, you should develop some unique naming convention that will only be used for these documents. For example, you could begin or end all TOC documents names with the letter **TOC**.

Protecting Existing Documents

You should never use the name of any existing source document as a TOC or index document name. If you do, you could lose your source document. For example, if we named this document *Chapter17* and also used *Chapter17* as the name of its index document, when the document is selected and the **Index** command is executed, the content of the chapter becomes the *backup* version of *Chapter17* and the index information becomes the document version. The second time the **Index** command is executed, the old index information becomes the backup version of *Chapter17* and the information in this chapter is permanently lost.

Although strictly speaking you need only worry about coinciding names in the same book, you should avoid any coincidence between index or TOC names and other document names. Otherwise, your documents could be in jeopardy when you paste them from one place to another. Never use the name *Index* for a source document.

Saving Documents

Since you can always make a new index or TOC document from your source documents, you might not need to maintain copies of them on the desktop. However, if you do not plan to modify the source documents, or to add or delete documents from the book, you might want to open index or TOC documents after you create them and execute the `Save →` `Fast` command so that the documents open more quickly the next time. You could then change the name of the index and TOC documents to prevent them from being overwritten.

Using Revert to Update Documents

You may want to leave your TOC or index documents open while you are editing the source documents they came from. This can be a very efficient way of locating and editing Index property sheets and the Component property sheets of TOC entries.

Using the **Revert** command, you can update TOC or index documents without closing them.

To update an open index or TOC document:

> ✔ Make any necessary changes in the source documents.
>
> ✔ Select the icons for the source documents or books for which you want new index or TOC document.
>
> ✔ Execute the **Index** or **Create TOC** command.
>
> ✔ If the window for the index or TOC document is not visible, select its icon and execute the **Open** command to bring it to the front of the other windows.
>
> *Executing* **Open** *for an already open icon has the same effect as executing* **Front** *on that icon's window.*
>
> ✔ Execute the **Revert to Document** command in the open index or TOC document. Your latest changes to the source documents are reflected in the document version of the TOC or index document when you revert.

Chapter 18

New and Modified Features

This chapter summarizes the major changes in the Interleaf publishing software between Release 2.5 and Release 3.0. Subjects presented here briefly are explained in greater detail in the appropriate chapters of the *Reference Manual,* Volumes 1 and 2.

Opening Documents in 3.0

Any document created in Release 2.5 can be opened in Release 3.0. As with previous releases, compositional integrity from release to release is not guaranteed. You may find that line and page breaks in a given document will differ when the document is opened in Release 2.5 and 3.0.

Compatibility

Once a document is saved in Release 3.0, it can no longer be opened in Release 2.5. If you intend to keep 2.5 on your disk for a transition period, we suggest that you maintain a distinct separation between documents you want opened in Release 3.0 and those you do not.

Here is one way to separate your 2.5 documents.

> ✔ Using Release 2.5, create a drawer and label it *2.5only*. Open the drawer.

> ✔ Select all of the desktop objects you want to be able to use in 2.5. Copy them and then paste them into your *2.5only* drawer.

> ✔ Do not use Release 3.0 software to edit or save the objects contained inside the *2.5only* drawer.

Because of the many additional features in Release 3.0, you are not likely to maintain Release 2.5 long after installation of the new software.

Beta Release Customers Only

There is no guarantee that documents created in beta versions of the software will open in the official 3.0 release. If you are able to open such documents, be sure to save them in Release 3.0. Documents created with the Beta release which are *not* saved in Release 3.0 may not open in future minor releases of software.

New Desktop Features

Streamlined Desktop Access

With Release 3.0, you will see a new login stickup. You type your login name
(and your password if necessary). What happens next depends on how your sys-
tem administrator has configured your system.

The publishing software may load immediately and put you directly on your desk-
top. If so, when you close your desktop you will automatically be logged out from
your workstation.

You may see an operating system prompt. In that case, you still have to type **ops**
to access your desktop or **logout** to log out from your workstation.

If you enter and exit the publishing software without seeing the operating system
prompt, you will not be able to suspend the publishing software to use an operat-
ing system shell. You should instead access the operating system using terminal
windows.

Timeout Feature

After 30 minutes in which you do not do any work on your desktop — no key-
strokes or mouse movements — the publishing software will checkpoint any open
documents and an Interleaf screensaver will appear on your screen. To continue
your work, just move the mouse or strike a key and your desktop will reappear.

The Default Desktop

In this release of the publishing software, we supply two objects on a newly-
created desktop: the *clipboard* and a *System* cabinet. The *System* cabinet contains a
Documentation drawer with on-line documentation, a modified *Graphics* cabinet, a
modified *Samples* drawer, and a new cabinet named *Create*. The *Templates* cabinet
is now inside the *Create* cabinet.

The Create Cabinet

The *Create* cabinet allows you to control which commands appear on the Desktop
Create submenu. In the simplest case, if you have a document template that you
use often, you can just put it in your *Create* cabinet, and it will appear on the
Create menu.

A more significant use of the *Create* cabinet can occur if more than one person is
using the Interleaf publishing software. Your system administrator can set up
cabinets in a central location that contain departmental and company templates,
and all users can create links to them in their *Create* cabinets. Everyone then has
access to the most recent templates through his or her Desktop Create menu.

Please refer to the chapter *Document Templates* for more information about the *Create* cabinet.

Changing Icon Type

Object property sheets now show an additional line called *Icon Type*. The name of the current icon type is in reverse video. When more than one box is shown, you can change the type of the icon. For more information about changing icon types, see the chapter *Desktop Management*.

Desktop Links

Using the `Copy →` `Link to` command for selected icons, you can create an icon that is a *link* to another object on your desktop or any object inside any directory on your desktop. A link gives you an alternative access point for the object to which it is linked. Links have an additional line on their Object property sheet called *Pathname*. You can edit this pathname so that the link points to any object accessible on your network. For more information on desktop links, see the chapter *Desktop Management*.

Document Locking

Sometimes when you select and attempt to open a document, the Document Locked stickup appears. This stickup indicates that a **lock**, which is a separate file related to the document, is in place for that document. Locks warn you when a document you attempt to open is already open, either through links or by a user logged in under the same name at a different workstation. This warning helps prevent loss of information. For more information about locks, see the chapter *Desktop Management*.

The Virtual Terminal

The **Paste** command has been added to the Terminal Nothing Selected menu. It enables you to paste text copied from the virtual terminal back into the terminal, or from one terminal to another, or from an *external* document into a terminal.

The **Flush Input** command appears on the Terminal Nothing Selected menu if you enter more than 256 characters (the UNIX command-line limit) or if you inadvertently paste more than 256 characters of text at a system prompt. In either case you would hear a beep and would not be able to continue without executing the **Flush Input** command.

For more information on the virtual terminal, see *Appendix B*.

New Text Processing Features

Component Master Definitions

With this release, the publishing software uses a new method for identifying and storing the **master definition** of a component. In previous releases, component master definitions were derived from the *first* occurrence of each component name in the document. Now, for each uniquely named component the software maintains a master definition that is independent of individual occurrences of the component. Essentially, this allows you to maintain master definitions without having instances in a given document.

- When documents created with a previous version of the software are opened in Release 3.0, the software automatically creates master definitions for each named component based on the first instance of each component.
- The component names that appear on certain Component submenus compose the list of component masters.

Since component master definitions are no longer linked to the first occurrence, the **Select 1st of Each** command has been eliminated.

For more information on component master definitions, see the chapter *Text Processing*.

Component Popup Submenus

Once you create 26 component masters in a document, the Component submenus that list the master names split into two submenus, each containing 13 names. There is no limit to the number of masters; the menus will just keep subdividing.

Tab Settings

The Tab sheet of the Component property sheet has been expanded, enabling you to set 30 tab stops per line.

Spell Correction and New Dictionaries

When you locate a misspelled word using the **Spell** command, you now have the option of viewing a list of alternative spellings/words. From this list you can select a word to automatically replace the highlighted word.

The **Add to Dictionary** command now adds the selected word to a permanent, user-specific *desktop dictionary*. The dictionary is represented by a dictionary icon.

In addition, your system administrator can create a *system dictionary* on his or her desktop. This dictionary is designed to store in one place the proper names and

site-specific terminology used frequently by many individuals on a network. Individual users can create a link from their system cabinet to the system dictionary.

The three dictionaries, the *master* dictionary published by Houghton-Mifflin (available with Release 2.5), the *desktop* dictionary, and the *system* dictionary work in conjunction to identify suspect words and to determine the hyphenation of words.

For more information, see the chapter *Spell Checking and Hyphenation*.

Microdocuments

Within frames you can now create a new type of text object called a **microdocument**. When a microdocument is open, you can perform all of the usual text processing and editing operations on text within the microdocument. When a microdocument is closed, it behaves like a diagramming object.

Like a document, a microdocument is composed of components and has its own distinct set of page properties. You can create many microdocuments in a single frame. Each microdocument can be a maximum of one page.

All of the tools for performing document-wide word searches, spell checking, numbering, referencing and indexing extend into microdocuments, except those in header and footer frames. Microdocuments currently cannot contain frames.

For more information on microdocuments, see the chapters *Text in Frames* and *Text as a Diagramming Object*.

Autonumbering and Referencing

The autonumbering feature allows you to number anything in your document — lists, footnotes, figures, headings, etc. — and to have the numbers automatically updated when you add or subtract items, or when you rearrange them. You can choose one of five different numbering schemes and have a prefix and/or a suffix automatically inserted with the number.

By default, every document has two streams, *List* and *Outline*. You can have as many other streams as you like. The streams can be intertwined. Streams can also be restarted, which allows you to create two similar lists without having to create a new stream.

The autoreferencing feature allows you to put references, like *Figure 20 shows...* or *On page 20 we discuss...*, in your documents and to have the reference automatically updated when you insert a new figure or another page of text.

Most autoreferences are linked to autonumber tokens. An autoreference token can either display the current value of the autonumber token it refers to or the number of the page on which the autonumber appears.

For more information on autonumber streams and autoreferences, see the chapter *Autonumbering and Referencing*.

Search and Replace

With Release 3.0, there are menu alternatives to the keyboard commands for search and replace. You can use **Find** to locate strings of text, frame anchors, index tokens, autonumber tokens, and autoreference tokens throughout a document. All document searches extend into microdocuments as well.

With **Replace** you can replace one string of text with another string of text. You can replace one instance at a time, globally, or from the current location to the end of the document.

For more information, see the chapter *Search and Replace*.

Revision Tracking

With this release, there are now three text attributes for tracking revisions — Strikethrough, Underline, and Revision Bars. Their status is displayed in a new Document Header box called the Attributes box. You can use this box to turn attributes on and off, or use the Attributes submenu on the Fonts submenu in text. In addition, a new entry on the Component property sheet allows you to toggle attributes for an entire component.

The new Custom sheet of the Page property sheet provides control over the placement of revision bars in a document. On the new Printer property sheet, you can control the printing of revision bars, underlines, and strikethroughs.

For more information on revision tracking, see the chapter *Document Management*.

New Page Makeup Features

Property Sheets

There is no longer any Document property sheet. You now make hyphenation choices on the Page property sheet and choices about printing on the new Printer property sheet.

There are now two Page property sheets: the Normal sheet and the Custom sheet. On the Normal sheet, the meaning of the Right 1st Page and Left 1st Page choices has been modified to guarantee that odd and even pages throughout a book maintain correct header/footer layouts.

On the Custom sheet there are a number of variables related to vertical justification that, in previous releases, were set by the software and could not be changed by users. In this release, you can modify values for these variables, if you so choose.

For more information, see the chapter *Page Design* or *Header and Footer Frames*.

Printing

There is now a Printer property sheet on which you determine, among other things, whether or not you want text attributes (revision bars, underlining and strikethroughs) to appear in your printed copy.

You can also indicate whether a given document will be printed double-sided or manual feed. Please note that some output devices, for example, the Lasercomp typesetter, do not support double-sided or manual feed jobs.

The command **alpr** is no longer used to print ASCII files from the operating system. You must use the **lpr** command.

For more information on printing, see the chapter *Printing*.

Frames

Release 3.0 introduces the concept of named frames. The publishing software now associates a **master definition** with all frames of the same name. Essentially, this feature changes the way frames inherit their properties (definitions).

In previous releases, the master definitions of frames were associated with the default text-anchored frame *types*.

Now you can change the properties of a frame, rename it (to create a new master), and simply select it on the Create Frame submenu any time you want to duplicate the frame in your document.

The first time you open a document in Release 3.0 that contains frames created in previous releases, the software will name the frames according to their placement properties. Master definitions will be assigned based on Release 3.0 default definitions. Existing instances that differ will not be modified to match the master definitions. However, any new instances you create will reflect the master definitions and, therefore, not necessarily match existing frames of the same name.

Another new feature of frames is *Shared Contents*. If you set this property to **Yes** on the Frame property sheet, frames of the same name within a single document can share their contents. This enables you to make a change to the contents of one frame and have the change automatically occur in all other frames of the same name.

For more information on master definitions and shared contents, see the chapter *Text-Anchored Frames*.

New or Enhanced Keyboard Commands

Most of the commands that you have used in text will also work in the new *microdocuments* available with this release. Several keyboard commands have been added or changed.

With this release, CTRL-l, CTRL-c, and CTRL-r control the alignment of both text strings and variable-width microdocuments.

Function	Keystroke	Result
Text String	CTRL-o	When this CTRL key sequence is combined with one of the alignment commands, a text string is created within an open frame.
Underline	F6	This toggle key turns underline on and off at the caret. It underlines selected text that is not underlined. It removes underlining in selected text that is underlined.
Strikethrough	F7	This toggle key turns strikethrough on and off at the caret. It strikes through selected text that does not have strikethrough on. It removes strikethrough in selected text that has strikethrough on.
Replace to End	CTRL–x CTRL–q	Replaces a string of text with another string from the current location to end of document.
British Pound Sign (£)	ESC [L]	Inserts a British Pound sign.
Section Mark (§)	ESC [⋮]	Inserts a section mark.

For more information on keyboard commands, see chapter *The Keyboard*.

New Diagramming Features

Arcs

Arcs have been added to the list of diagramming primitives available. In addition to the modifications you can make to any primitive, you can also edit the shape and extent of an arc. For more information see the chapter *Creating and Modifying Arcs*.

New Poly Definition and Polys with Holes

The definition of what in previous software releases was called either a poly or a polygon has been broadened and brought under the umbrella term, **poly**. The results of the new definition are: polys can now consist of combinations of all of the primitives instead of just lines; and you can now create polys with holes in them. Each of the characters in the Serif, SanSerif, and Poly fonts, in the *Fonts* folder of the *Graphics* cabinet, is an example of a single poly. For more information see the chapter *Objects in Diagramming*.

Dashed and Dotted Widths

Objects with width — lines, boxes, arcs, ovals — can be given dashed and dotted edges with the **Dashes** commands on the Object Selected Props submenu. You can also specify a dashed or dotted pattern for new objects using the Create Misc Defaults command. For more information, see the chapter *Modifying Diagrams*.

Isometric Grid and Isometric Projections

You can now use an isometric (tri-axial) grid to create and manipulate three dimensional looking diagrams. You can also project a selected object onto one of three isometric planes. For more information see the chapter *Advanced Diagramming Concepts*.

View Commands: Zoom, Center, and Shift

To make detailed diagramming work easier, you can use the **Zoom** commands to enlarge the diagram up to 16 times original size, shrink it to 1/2 actual size, and toggle between zoomed and actual size. The **Center** and **Shift** commands help you quickly move parts of the diagram into a position where you can work on them conveniently. For more information see the chapter *Basic Diagramming Concepts*.

Gravity Radius

The **Gravity Radius** command lets you specify how close the active control point on one object can come to a gravity point on another object before gravity forces them together. This distance is called the *gravity radius*. The default gravity radius is 4 pixels. You can set a new radius of between one and eight pixels. For more information, see the chapter *Basic Diagramming Concepts*.

Free-hand Drawings

The **Create SubEdit Draw** command allows you to create free-hand drawings, using the mouse as a "pen." You can edit and manipulate a drawing just as you would any other poly. For more information see the chapter *Advanced Diagramming Concepts*.

Wipe-Select

The **Wipe** command on the Select submenu lets you select objects by wiping over them with a special cursor. This feature allows you to quickly select and cut the many short lines which can make up the curved sections of free-hand drawings. For more information, see the chapter *Basic Diagramming Concepts*.

Animation of Plotter Objects

When you execute an animation command, such as **Move**, on a plotter object, the object is replaced by a box which outlines its perimeter. This greatly increases the speed with which these complex objects can be moved. The object is redisplayed in full when it is deselected.

The *Serif* and *SansSerif* fonts in the *Fonts* folder in the *Graphics* cabinet are animated in the same way.

New Document Management Features

The Book Directory

A book is a special directory that creates a relationship among the documents it contains. It supports consecutive and interactive page, figure, and list numbering from document to document. It also provides the ability to create indexes and tables of contents automatically. By dividing a document into smaller documents within a book, you retain much of the unity of a single document while gaining the ease of handling smaller units. For more information, see the chapter *Large Documents in Books*.

Automatic Indexing

The automatic indexing feature is designed to let you index concepts, not just key words and phrases. You insert tokens at the locations that you want to reference in a document and then fill in a property sheet with the information you want connected to that token. You can specify the range of pages you want the entry to cover and make cross-references to other entries.

When you execute the **Index** command in a book, an index for one or more documents is assembled.

For more information, see the chapter *Indexes and Tables of Contents*.

Tables of Contents

This release of the publishing software includes an improved table of contents generation feature. Instead of selecting all of the components you want in the table of contents each time you create it, you enter the name of a table of contents document on each component's Custom property sheet. Once you make this entry, these components will automatically become part of an ASCII table of contents document every time you execute the **Create TOC** command in a book.

As with indexing, you can create a single table of contents for as many documents as you want. The TOC document is formatted in 10-point Classic type no matter what the original component settings are.

For more information, see the chapter *Indexes and Tables of Contents*.

Optional New Features

Equations

The new Equation editor allows you to create and lay out equations in frames. To edit an equation, you open the Equation Edit sheet and enter commands that specify what characters are in the equation and the function of each character.

You can create superscripts, subscripts, fractions, square roots, integrals, summations, and other expressions. For example, to create the expression, a^b, you would enter the text "a sup b".

If you don't like the character placement that the equation editor chooses, there is another set of commands that allows you to adjust it. The Greek alphabet and many special mathematical symbols are available.

For more information on equations, see the chapter *Equations*.

Typefaces

If you have purchased certain optional fonts, you can create the **bold-italic** typeface. Optional fonts also provide a greater choice of point sizes. For details on optional fonts, see the *Interleaf Typeface Catalog*.

The Image Editor

The Picture Editor has been modified extensively to make it possible to edit continuous tone images. The enhancements also increase the possibilities for editing line art images. New features include:

- **Zoom** magnifies the image so that you can see the effects of your changes.
- The crosshair brush makes it easier to identify pixel boundaries in a magnified image.
- **Clone** allows you to copy one part of an image onto another.
- **Pickup** allows you to choose any shade of gray in the image by pointing at it.
- **Color Numeric** lets you choose by number any of the 256 shades of gray available.
- **Erase** replaces the Brush -> Color -> None command in Release 2.5 and provides additional power to undo previous changes.
- The status line now reports the color, shape, and size of the brush and the degree of magnification for the image.

The Contrast Editor has been enhanced so that you can map the two colors of line art images — black and white — to any two shades of gray.

The new **Update Picture** command in the Frame Editor allows you to permanently alter the boundaries of an image. Normally, the original image is preserved even if you have edited it so that only a small portion is visible.

Tracing

The optional **tracing tablet** feature lets you use the diagramming system to create accurate copies of line art. The tracings you create, which consist of standard diagramming objects, can be edited like any other diagram. For instance, you might trace a street map using lines, polys, free-hand drawings, and arcs. The Tracing Tablet package includes:

- a magnetic tracing tablet
- a magnetic tracing puck
- a magnetic tracing stylus

Index

Italicized numbers refer to pages on which figures appear.

L

N

Newsletter, sample layout using microdocuments, 11-26—11-28, *11-27*

Newsletter Template, *13-6*

NO TAG, 7-13

Notes Document, Graphics cabinet, 25-2

Null String, 6-9

Numbering. *See* Autonumbers

O

Object Editor, definition of, 19-1

Object Properties
  changing, 15-7—15-12
  defaults and, 15-7
  permissions and, 15-8

Object Property Sheet, 15-5—15-11, *15-6*
  change and access time on, 15-9
  changing permissions on, 15-8
  and group, 15-7
  Icon Type, 15-8
  of linked object, *15-13*
  and object ownership, 15-6
  and others category, 15-7

Objects in Diagramming, 19-1
  See also Graphic Fonts; Graphics Cabinet, documents in; Microdocuments; Text Strings
  aligning, 21-18—21-22
  animation state of, *19-11*
  arcs, 22-1—22-8
  clipboard and, 21-15—21-16
  control and anchor points on, *20-3, 20-19*
  creating, 2-30—2-31, 20-2—20-3, 25-25
  cutting, pasting, and copying, 21-15—21-17
  default properties of, 20-2, 24-24—24-29
  definition of, 20-1
  duplicating, 21-14—21-15
  editing, 24-1—24-3
  effect of Gravity and GridAlign on, 20-3
  filling, 20-14—20-16
  gravity points on, 19-18—19-19
  grouping, 20-10—20-13
  in header/footer frames, 12-2—12-3
  with holes, *20-15*, 20-15—20-16
  images and, 26-2
  locks for, 24-10—24-17
  moving, 21-1—21-4
  paths, *20-14*

polys, 20-14—20-15
primitive, 20-1—20-3
properties of, 20-4—20-9
rotation of, 21-8—21-14
selecting, 19-2—19-9
selection state of, *19-10*
sizing, 21-4—21-8
splines, 20-17—20-19
types of, 20-1
undoing changes to, 21-23—21-24

On-line Documentation, Reference folder, Math Notes, 27-12, 27-14

Open Command, and permissions, 15-10

Origin Line
  bar and line charts, *28-31*
  Chart Customize sheet, *29-12*

Orphan, control of, 9-12—9-13

Outline Template, 13-7

Oval Point, definition of, 25-5

Ownership, of desktop objects, 15-6—15-7

P

Page
  orientation, 9-3—9-4
  properties, 9-1—9-2
    See also Page Makeup
  range, on Index property sheets, 17-9
  width and height, 9-4

Page # Prefix, on Document Page property sheet, use of in books, 16-7

Page # Style, on Document Page property sheet, effect on documents in books, 16-5

Page Box, in document header, *4-39*

Page Commands, on Page pulldown menu, 4-39—4-41

Page Makeup
  See also Microdocuments, unique layouts using
  Allow Break After, 9-13—9-16
  Allow Break Within, 9-13
  Begin New Page, 9-10
  double-sided layout, 12-12—12-13
  multiple column. *See* Multicolumn Documents
  orphan and widow control, 9-12—9-13
  property sheets used in. *See* Component Page Property Sheet; Document Page Property Sheets
  single-sided layout, 12-11—12-12

Page Number Token, 3-13

stacking order of, 2-14—2-15
Window popup menu, 2-13

Wipe-Select Mode, 19-8—19-9
cursor and, 19-8

Write-Only Objects, 15-11

Z

Zoom Command, in diagramming, 19-21—19-22