XYQUEST

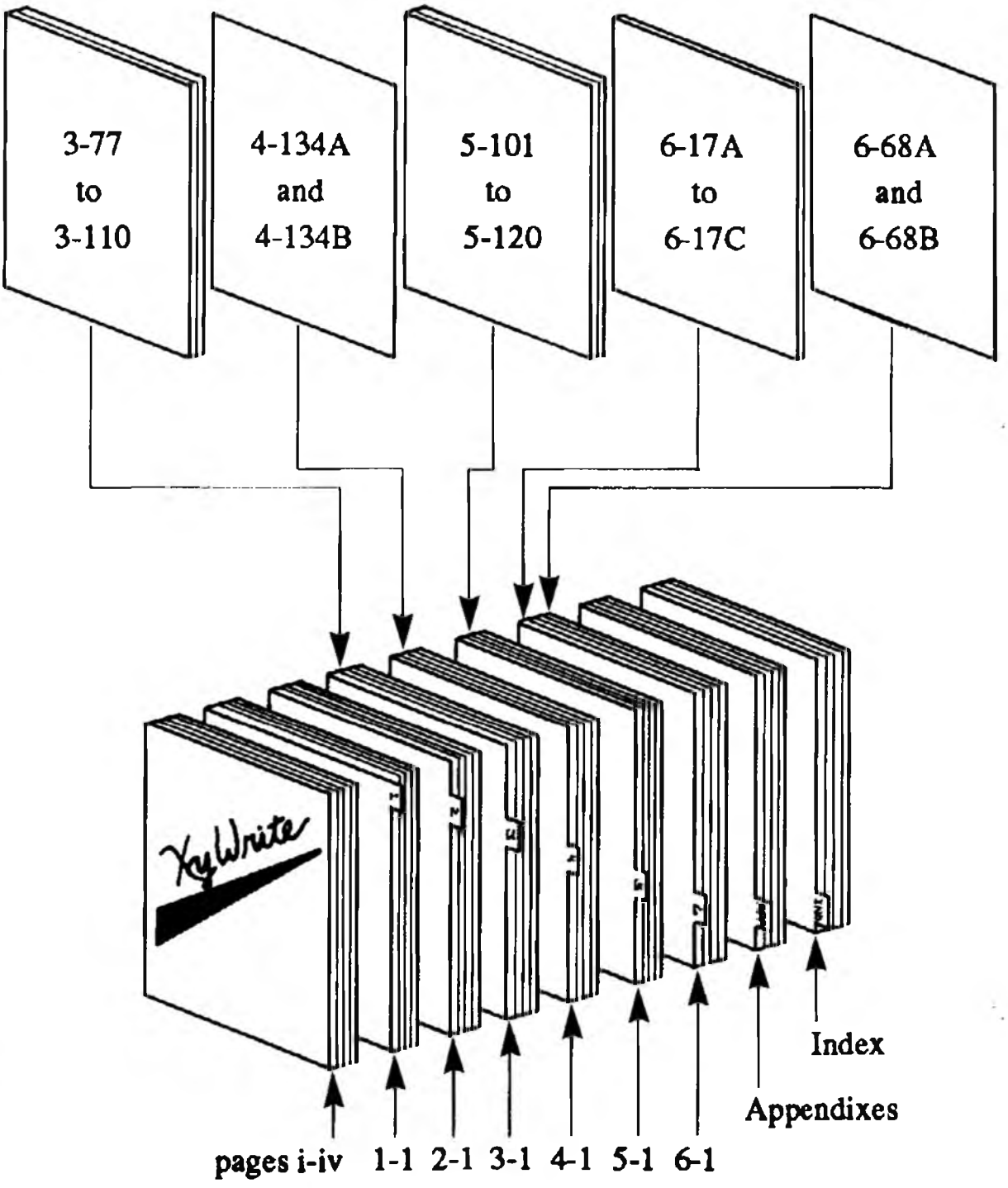
Second Edition (October 1986)
Document Control No. 002396-01

XyQuest, Inc.
P.O. Box 372
Bedford, MA 01730
~~(617) 275-4439~~
(508)671 0888

Manual written and produced by
Douglas Kramer and
DFM, CAV, JKH, NJT, CTS, SR, KL, SGL, PMT, WL, WK, RG, HP, GLM, AE, SER, CYC, RWF, DNH, AFP, RG, SJR, RLN, ABD and DLN.

And many thanks to JR and those XyWriters whose comments have helped to make a better manual.

How to Insert Tabs into the Reference Guide. The *Reference Guide* has six sets of changes/enhancements and eight thumb-tab dividers. If you need assistance, use the illustration below to insert these changes and dividers into the manual in their correct order.



XYQUEST, INC.
P.O. Box 372
Bedford, MA 01730
(617) 275-4439

Welcome to XyWrite III.

XyWrite (pronounced zi'-write) is a word processing program which integrates text preparation and output formatting. XyWrite operates on IBM personal computers and many compatibles. All files created by XyWrite are *pure ASCII*.

This manual describes the complete set of XyWrite commands and procedures. It is intended as a reference guide for all users, at virtually any level of experience.

This manual was written using XyWrite on an IBM PC/AT with two floppy drives (no hard disk) and 256K of memory.

This book is a *reference guide* and is organized by topics. As such, the easiest way for a novice to find information is by using the **Index,** which is complete and well cross-referenced. Also refer to the tutorials for step-by-step instruction. As you become more familiar with the program, it will become clearer what information is in each chapter. Chapter 1 contains general helpful information. Chapters 2, 3, and 4 cover the basic elements of XyWrite. Chapter 5 has procedures for more involved and highly useful applications. Chapter 6 is for more intermediate users — it contains information of benefit to someone who wants to customize XyWrite.

1 Getting Started

2 Filing

3 Editing

4 Formatting

5 Special Features

6 Customizing

Appendixes

Index

Features. XyWrite allows you to:

- Work on many documents at once (up to nine).
- Review the formatted page *before* you print it out.
- Print in the background while you return to work on the same or other documents.
- Run any DOS command from XyWrite.
- Store text to a Save/Get key, for ready retrieval and insertion into your document. (Known as the Glossary function in other word processors.)
- Create forms by setting up a file so that you can type into only the blank fields.
- Generate a Table of Contents and an Index with automatic sorting and page numbering based on phrases you mark in your document.
- Paginate a document automatically with widow/orphan control and unbreakable blocks.
- Use Mail Merge to produce form letters.
- Record keystroke sequences and create custom procedures with user programming features.
- Perform arithmetic.
- Print mathematic symbols and foreign language characters using character substitution tables.
- Insert footnotes and endnotes in your document.
- Treat multiple files as if they were a single, large document. The files can be printed together with continuous page numbering, and an all-encompassing Table of Contents and Index.
- Reassign keys on the keyboard with different characters, functions, and combined functions.
- Enter text into multiple columns with word wrapping within columns.
- Print proportionally spaced text with automatic hyphenation.
- Read and write ASCII files — Xywrite files are pure ASCII.

And best of all XyWrite is FAST!!

The Manual Set. XyWrite provides the following manuals:

- *Installation Guide*
- *Quick Start Tutorial*
- *Basic Word Processing Tutorial*
- *Applications Tutorial*
- This *Reference Guide*

The *Installation Guide* provides directions for installing XyWrite on your personal computer.

The *Quick Start Tutorial* teaches the minimum set of skills necessary to start using XyWrite within twenty minutes of installation.

The *Basic Word Processing Tutorial* describes commonly used XyWrite commands.

The *Applications Tutorial* assists you in mastering particular tasks, such as building a Table of Contents and Index, writing form letters and programming XyWrite.

About This Manual. This manual is written to help users at virtually every level of experience. You can use this reference guide even while you're learning from the tutorials. Each section is self-contained, so that you can read sections in any order. For example, you will find the description on footnotes all in one place, including both the footnote procedures and commands.

The description of each command follows this structure:

Purpose: Why would I want to use this command?
Action: What steps do I take to use this command?
Notes: What else should I know about this command?
Format: What is the precise syntax of the command?

Format can appear either at the beginning or end of a section.

Entering Commands. You enter commands on the Command Line, which is the top line of the display. The following rules apply:

- Always start at the leftmost position of the Command Line. Press F5 to do this.
- Use either upper or lowercase letters (or any mix).
- Insert a single space immediately after the command name (if it is to be followed with arguments).
- Separate multiple arguments with a comma. (Some commands, such as TYPE, COPY, RENAME, allow use of a blank space.)

Once you have typed the command on the Command Line, you execute it by pressing either F9 or ↵.

For example, any of these will work:

```
CMtype chapter.doc,1-3
CMtype chapter.doc 1-3
CMTYPE CHAPTER.DOC,1-3
```

Notation. Throughout this manual we have used the following conventions.

- *filename* — Anywhere you see the term *filename*, you may substitute the more general term *filename.ext*. For example, to call up an existing document:

  CALL *filename*

  can be replaced by:

  CALL *filename.ext*

- *d:* — Drive specification A:, B:, C:, D: or any other drive. The examples in this book generally assume you are using a system with only two drives. Hence we often refer to the DOS prompt as "A>" or "B>." If your system has other drives, you can of course use C>, D>, or any other drive.

- *path* — The path is a series of directory names separated by backslashes. You use paths only if you use subdirectories. Subdirectories are most commonly used only on hard disk systems. For more information on paths, refer to the Overview section at the start of Chapter 2, Filing.

- *italic* — The use of italic with commands is reserved for words (or characters) which are themselves to be replaced, such as *filename*. When a statement can be typed literally, letter for letter, we use the following font instead:

  call chapter.doc

- *#* — The pound sign means one single character — any letter (A-Z) or number (0-9).

- *n* — represents any number. (The number can contain more than one digit.)

- CM — Refers to the Command Line, located at the top of the screen.

- blue text — Indicates which keys you press to perform an action, in sections titled "Action":

  Type: F5 call chapter.doc ↵

The symbols in the left column are used throughout this manual to represent keys on the keyboard. For a description of these keys, refer to the Keyboard section in Chapter 3, Editing.

A to Z	Character Keys on the center portion of the keyboard
0 to 9	Number Keys in either the top row or on the numeric keypad
F1 to F10	Function Keys F1 through F10 to the left of the keyboard
Tab	Tab Key (Located just above the Control Key)
Ctrl	Control Key
Shift	Shift Key
Alt	Alternate Key
Space Bar	Space Bar Key
Backspace	Backspace Key (Located above the Enter key)
↵	Enter Key (also known as the Return key)
Esc	Escape Key
Break	Break Key (the same as the Scroll Lock key)
Caps Lock	Caps Lock Key (toggle key)
Scroll Lock	Scroll Lock (toggle key)
Num Lock	Numeric Lock (toggle key)
←, ↑, →, ↓	Cursor Keys
PgUp, PgDn	Page Up and Page Down Keys
PrtSc	Print Screen Key
Del	Delete Key
Ins	Insert Key
+, −, *, /, =	Math Symbol Keys
Home	Home Key
End	End Key

What You Need. To use Xywrite, this is what you need:

- The XyWrite diskette.
- An IBM Personal Computer or equivalent.
- 256K bytes or more of main memory.
- IBM PC-DOS 2.0 or higher.
- A monochrome or graphics adaptor with an 80-column display.
- One disk drive.
- A printer.

If your system has more than the minimum number of bytes required, Xywrite makes the most of the extra memory. It uses up to 640K of memory. More memory results in faster performance, especially when editing large files.

EDITOR.EXE is the only file that is *essential* to running XyWrite. Your Printer File is needed to print any of your files. If you copy XyWrite to another disk, you must include these files. The other files are needed only if you want them. For example, the Startup File would start XyWrite with your own commands, the Help files would give you access to the Help Screens, the Keyboard File would give you your own set of keys, and so on:

Essential Files

XyWrite Program:	EDITOR.EXE
Printer File:	*filename*.PRN

Accessory Files

Startup File:	STARTUP.INT
Help Files:	LONG.HLP, SHORT.HLP
Hyphenation Dictionary:	DICTION
Keyboard File:	*filename*.KBD
Save/Get File:	*filename*.SGT
Character Subst. File:	*filename*.SUB

XyWrite Files Are Pure ASCII. XyWrite files contain nothing but the 256 ASCII characters shown in Appendix A —XyWrite does *not* insert control characters into your document. Therefore, you can transport files to and from other ASCII systems.

Help. You have three sources of help available to you: On-Screen Help, the set of manuals, and XyQuest telephone assistance. Each is described briefly here.

On-Screen Help. Help is only a keystroke away. There are help files on-disk you can load to help you at any time. You can choose either a long version (LONG.HLP) or a short version (SHORT.HLP). To view a Help file once loaded:

Press: Alt F9

Manual Set. Unless you've become familiar with how the chapters are divided, the best place to look for information is in the index.

Telephone Assistance. If the first two sources cannot help you, then by all means call or write us at the following location. You can get information on-screen with the HELP command:

Type: F5 help ↵

The address and telephone number are:

XyQuest, Inc.
P.O. Box 372
Bedford, Mass. 01730

(617) 275-4439

It will help us if you do the following:

1. When you call, please be sitting at your computer with the problem at hand.
2. Have the following information available when you call us. Type HELP as shown above for this information:
   - XyWrite I, XyWrite II or II+, or XyWrite III
   - Version Number of program
   - Serial Number of program

CONTENTS

The content of this chapter is arranged alphabetically by command, as follows:

OVERVIEW

The chart on the facing page shows how the commands in this section relate to each other. They are categorized as follows:

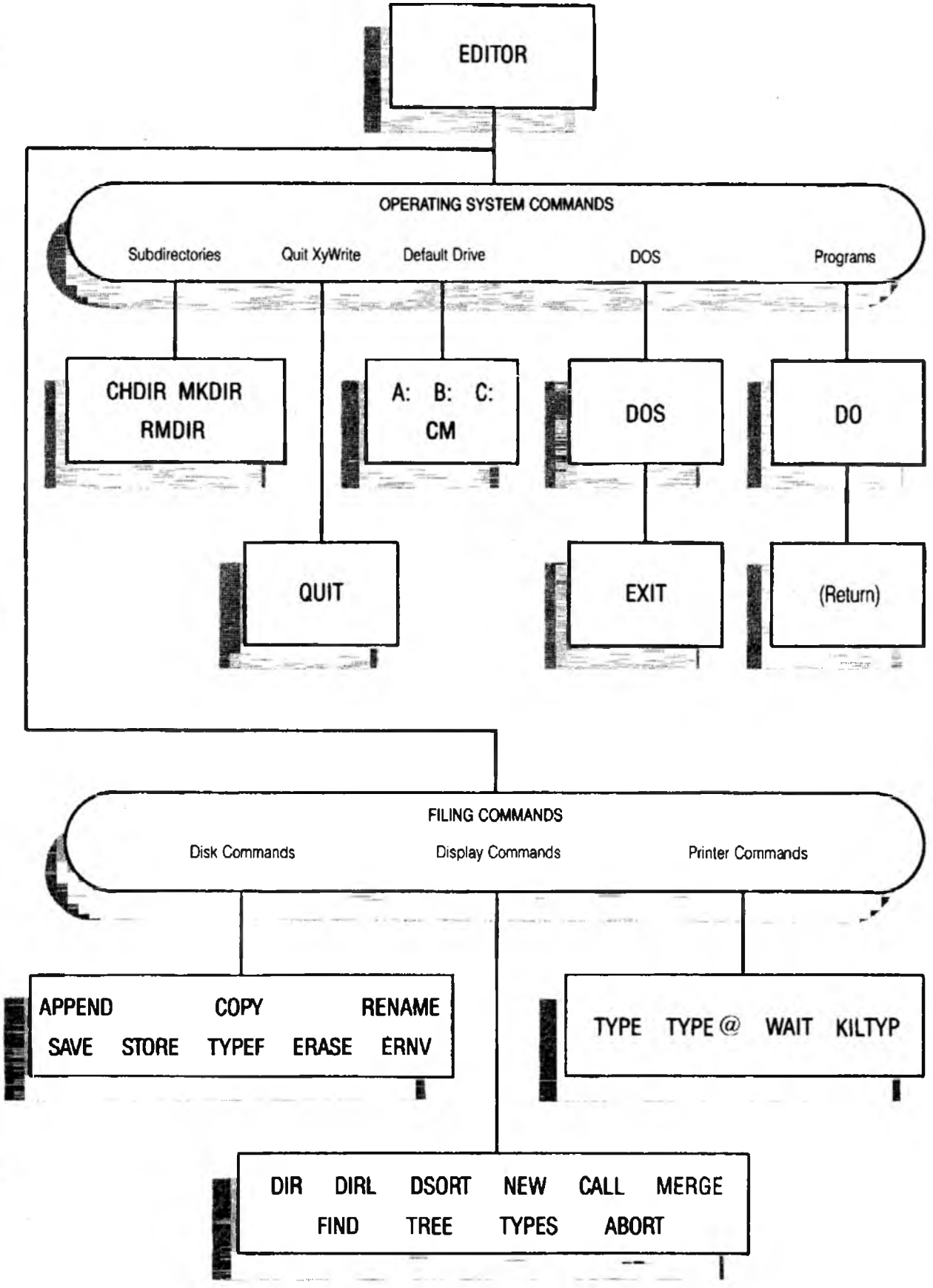
- **Filing Commands** — which handle only *entire* documents, or files, rather than individual pages, words, or paragraphs. For example, NEW creates a new document, and STORE saves the entire document onto disk.
- **System Commands** — which enable you to enter XyWrite and run DOS under XyWrite. In addition, the DO command allows you to run other programs (besides DOS) under XyWrite.

The Filing commands are further broken down into three categories:

- **Display Commands** — which load files to the display, and clear files from the display, *without altering the files on disk*. For example, CALL CHAPTER.DOC loads a copy of the file CHAPTER.DOC from the disk to the display; the original file CHAPTER.DOC remains on the disk.
- **Disk Commands** — which save files to the disk, and erase files from the disk. (The disk is, of course, the place where files are stored.)
- **Printer Commands** — which send files to the printer.

Several other specialized Filing commands appear in later chapters — commands such as LDPRN, LDSGT, LDKBD, LDSORT, LDDICT, NEF and NEP.

Immediate vs. Embedded Commands. All of the commands in this chapter are immediate commands. This means that when you type a command, it executes *immediately* on pressing ↵ (Enter). In contrast, the commands in Chapter 4 are Embedded Commands — on pressing ↵ they are embedded in the text as characters, ready to execute when output to the printer. For a description of Embedded Commands, refer to the Overview section at the start of Chapter 4, Formatting.


EDITOR
OPERATING SYSTEM COMMANDS
Subdirectories
Quit XyWrite
Default Drive
DOS
Programs
CHDIR MKDIR
RMDIR
A: B: C:
CM
DOS
DO
QUIT
EXIT
(Return)
FILING COMMANDS
Disk Commands
Display Commands
Printer Commands
APPEND COPY RENAME
SAVE STORE TYPEF ERASE ERNV
TYPE TYPE @ WAIT KILTYP
DIR DIRL DSORT NEW CALL MERGE
FIND TREE TYPES ABORT

Use of Path. The following description of paths is applicable to you only if you use subdirectories. (Subdirectories are used mainly on hard disk systems.)

A *path* is a series of directory names separated by backslashes (\). If a path begins with a backslash, XyWrite starts its search from the root directory; otherwise, the search begins at the current directory. Thus, you can specify a path in either of two ways:

\dir1\dir2\dir3	Relative to *root* directory
or	
dir1\dir2\dir3	Relative to *current* directory

Calling a File Using a Path. Any of the XyWrite commands which use a *filename* allow you to specify a *path* along with the filename. For example, you can CALL a file as follows:

CM call *d:path\filename*	Format
CM call a:\memos\vacation	Example

Result: The file VACATION is called from subdirectory MEMOS. The path and filename both appear in the NM field at the top of the display, whether or not you include them in the CALL command.

Storing a File Using a Path. The path appearing in the NM field also indicates where the file will be stored. Therefore, when you store a file, you return it to the subdirectory from which it was called. Use the command:

CM store

Result: The file is stored back in the subdirectory given by the path shown in the NM field. According to the previous example, VACATION in subdirectory MEMOS is updated.

To STORE a file in some other directory, you must specify the path to that directory when you issue the STORE command.

For more information on paths, refer to "Specifying a Path to a File" in your IBM DOS Manual.

PURPOSE

Any drive can be designated the default drive — a diskette drive, hard drive, or a RAM (virtual) drive. Setting the default drive is done by entering the drive letter on the command line. For example:

CM b:

When you set the default drive, you are telling XyWrite which drive to use when a filename is given without a drive letter prefix. For example, if drive B were made the default drive, then the following statement would call CHAPTER.DOC from drive B:

CM call chapter.doc

There are two options for setting the default drive:

- Setting the Default Drive *(Option 1)*
  CM *d:*
- Setting the Default and Save-Drives *(Option 2)*
  CM *d:,d1:*

Option 2 allows you to specify an additional *save-drive* (here denoted *d1*). If you use Option 2, then every time you SAVE or STORE a file, a copy is made not only to the normal default drive, but also to save-drive *d1*. Thus, Option 2 allows you to keep up-to-date copies on other drives. (You can specify several save-drives, if you wish.)

ACTION *(Option 1)*

Setting the Default Drive.

To set the default drive to drive A, for example:

Type: F5 a: ↵

ACTION *(Option 2)*

Setting the Default and Save-Drives.

First you specify the normal default drive (drive C in the following example); you then specify the save-drives immediately afterwards, separated by commas (in this case, only one save-drive, drive B). Do not include a space after the comma.

Type: F5 c:,b: ↵

Result: Now every time you SAVE or STORE a document, XyWrite will save identical copies on drives C and B. Drive C is the default drive for all other commands (e.g., DIR, CALL, TYPE, MERGE). When you specify one or more save-drives, you cannot SAVE or STORE to just a single drive.

NOTE #1 **Example of Two Save-Drives.** In Option 2 (above), to specify *both* drives B and A as save-drives in addition to default drive C:

Type: F5 c:,b:,a: ↵

NOTE #2 **Typical Uses for Save-Drives.**

- RAM Drive Backup.
  If you work on a RAM Drive, you can make sure your files are also stored on your floppy drive by setting it to be a save-drive.
- Two-Drive System.
  If you want to back up files onto another disk as often as possible, then you can specify your second drive as the save-drive. The only drawback is the additional time it takes to save to the second drive.

NOTE #3 **Default Drives at Startup**. By adding the default drive as a line in your STARTUP file, you can have XyWrite automatically switch default drives at startup.

FORMAT

CM *d:* *(Option 1)*

CM *d:,dl:* *(Option 2)*

- *d:* is the drive letter (A:, B:, C:) desired for the normal default drive. Always include the colon (:).
- *dl:* is the save-drive you specify. You may specify any number of save-drives, separated by commas. Do not include spaces before or after the commas.
- This is an immediate command (it is not embedded).

EXAMPLES

CM a: *(Option 1)*

CM c:,b: *(Option 2)*

PURPOSE

ABORT clears the document from the display window and memory. Any changes made to the file since it was last saved are lost and cannot be recovered. ABORT never affects any files on disk. The form for ABORT is:

CM **ABORT**

ABORT has two main uses. (The procedure is the same for both.)

- Clearing the Display.
  If you have called up a document merely to view, and you have *no changes* to save, use ABORT (rather than STORE) when done — ABORT is quicker.

- Undoing a Big Mistake.
  If you have made a disastrous mistake in editing a file, the ABORT command lets you throw away the working document which contains those errors. (See the tip on the next page.)

ACTION

Clearing the Display

To clear a document from the display:

Type: F5 abort ↵

Result: The display is cleared. Note that if there is more than one document open when ABORT is executed, the document which is currently *active* is the one which is aborted. (See "Active Window" in the section "Window.")

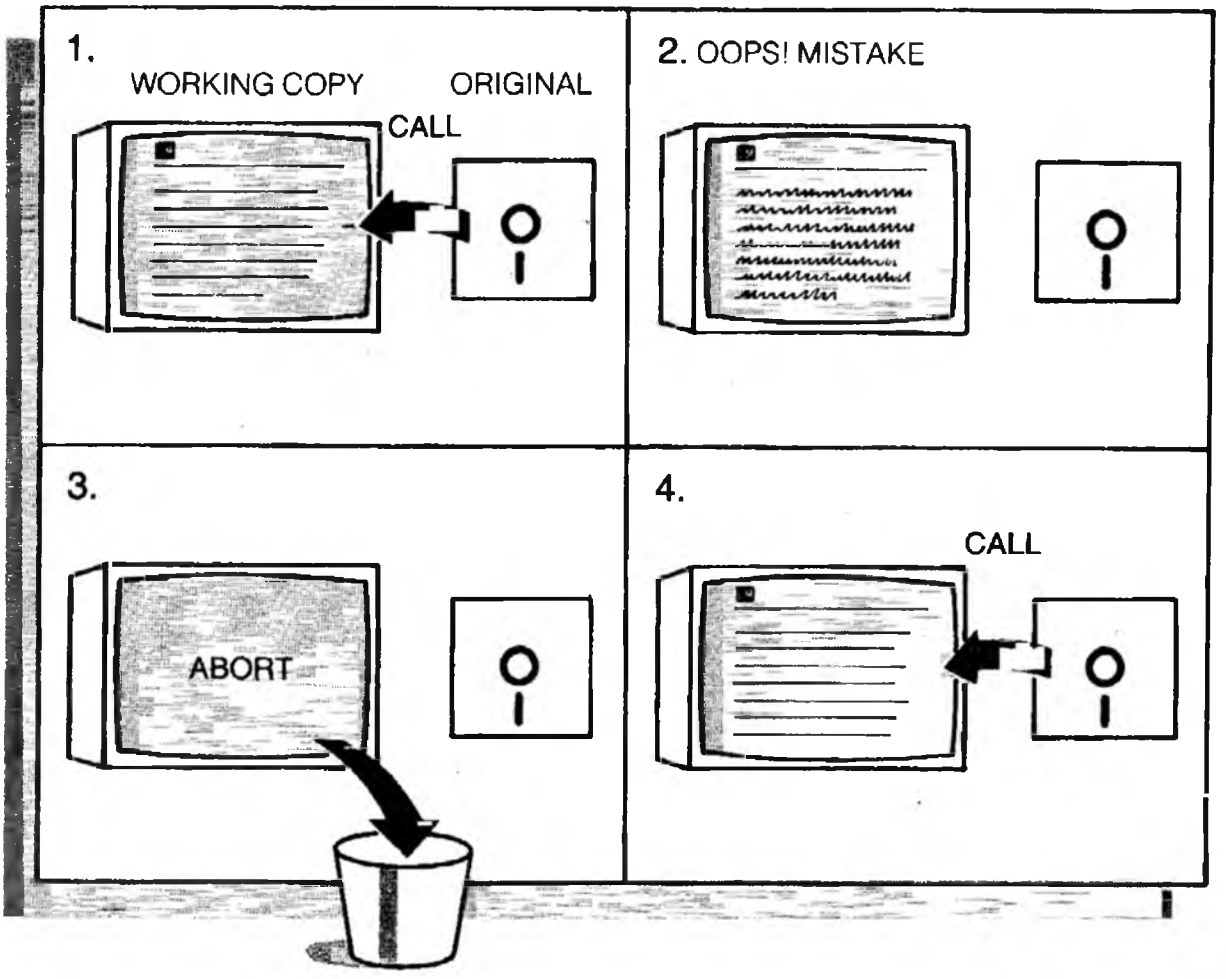
FORMAT CM **ABORT**

- ABORT is an immediate command (it is not embedded).

ABBREV CM ab

TIP

How to Undo a Big Mistake. (See the illustrations.) Whenever you CALL a document (1), a *copy* is sent to the display; the original remains on the disk. As you make changes to the document, the changes are made only on the copy in memory, not on the original file on disk (until you SAVE it). Thus, if you make a gross blunder with a search-and-replace (2), then you can ABORT the goofed-up document (3), and still retrieve the original document (4).



PURPOSE

The **APPEND** command adds text to the end of a file on disk. The text you append can be copied from the file on screen or from another file on disk.

If you type the APPEND command with only one filename (Option 1), XyWrite looks for defined text within the file on your screen. If it finds defined text, that text is appended to the file you named; otherwise, XyWrite adds the entire current file to the end of the named file.

If you list two files after the APPEND command (Option 2), XyWrite copies the first onto the end of the second. The first file remains unchanged.

ACTION *(Option 1)*

Appending the Current File to a Stored File.
To append the document you have on screen (or defined text within that document) to another file, say CHAPTER on drive A, for example:

Type: F5 append a:chapter ↵

Result: The current file (or defined text) is now copied to the end of CHAPTER.

ACTION *(Option 2)*

Appending a Stored Document to Another.
To append TIMECARD to the end of REPORT:

Type: F5 append timecard,report ↵

Result: TIMECARD is added to the end of REPORT. TIMECARD is unchanged.

NOTE

Release Defined Text. When adding the current file to another, it's smart to strike the F3 key (Release Define) before using APPEND; otherwise, you may be adding some off-screen defined text instead of the entire file as you intend.

FORMAT

CM **APPEND** *d:filename,d:targetfile*

- *d:* (optional) is the drive where the file is stored
- *filename* (optional) is the file to be added to another
- *targetfile* is the file to which text is added

EXAMPLE

CM append b:report,c:customer.ltr

PURPOSE

The **CALL** command loads a copy of the named document from the disk into memory and the display for viewing and editing. The general form for CALL is:

CM **CALL** *d:filename*

CALL works by *copying* the file to the display. The original document remains safely on the disk. Thus, you may edit or even ABORT the displayed document without disturbing the original file. (Only when you SAVE the document back to the disk does the file on the disk change.) There are two ways to call a file:

- Calling a Document by Typing Its Name.
  (Option 1) This is the way you would normally expect to execute the CALL command.
- Calling a Document by Pointing at Its Name.
  (Option 2) This is a clever way to call a document — by pointing at its filename in the directory.

ACTION *(Option 1)*

Calling a Document by Typing Its Name.

If you already know the name of the file you want displayed, use this procedure. Refer to the illustration below:

1. Start with the display cleared of any document. To clear the display, use STORE, ABORT, or open a new window (to find out how, see "Windows," Chapter 3).

   (You can eliminate this step with a setting that automatically opens a new window for a CALL or DIR command. See Note #4 below.)

2. Let's say the name of the file you want to call is CHAPTER.DOC on drive B:

   Type: F5 call b:chapter.doc ↵

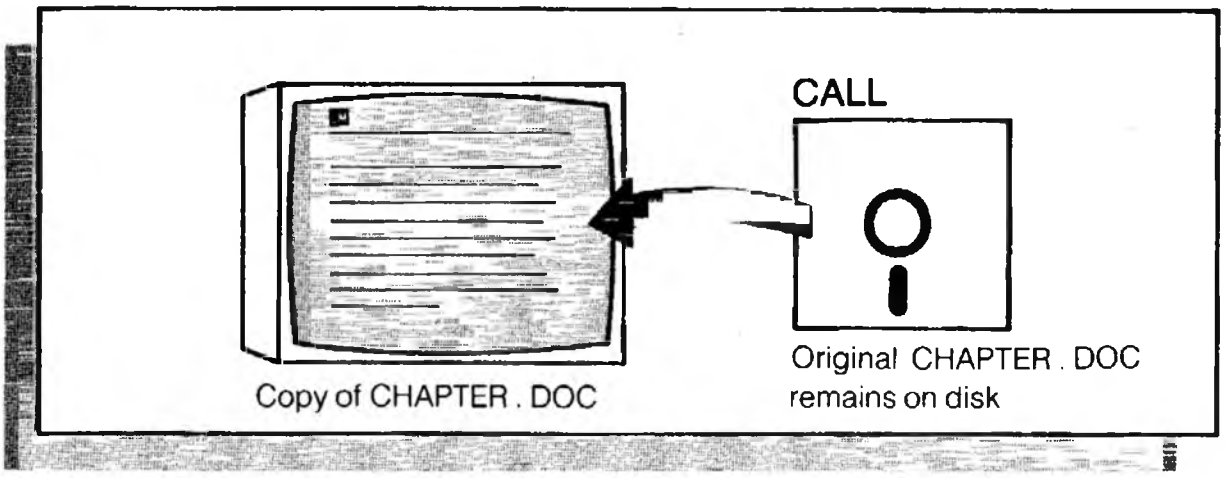
Result: This example calls the file named CHAPTER.DOC to the display from disk drive B. (If the "b:" were omitted, XyWrite would look on the default drive for the file.)

NOTE #1 **Recalling the Previous File.** If you omit the filename, CALL brings up the most recently opened file.

NOTE #2 **Saving the Document.** CALL loads the document into memory. The document remains in memory while you work on it. Any changes you make to the document are not saved on disk until you SAVE or STORE it. Because memory is lost if the power fails, it is a good idea to save the document occasionally, say every 15 minutes, especially after significant changes.

When you STORE or SAVE a document (without specifying a drive, directory or filename), it is always returned to the directory from which it was called, even if you changed directories while the document was displayed.

NOTE #3 **Calling with Global Filenames**. You can substitute the wild cards * and ? for characters in a filename with CALL, in the same way you use them with the DIR command—for example: CALL *.DOC. When you use a global filename with CALL, the first file is displayed on the screen. After you have finished reviewing or editing the displayed file, press Ctrl N to display the next file that matches the global filename you specified in the CALL command. You will see the message "Save edits (Y/N), or ignore (I)." Press "Y" to store the current file and display the next one; press "N" to abort the current file and display the next one; press "I" to keep the current file on the screen. Repeat this procedure until you see the message "File not found."



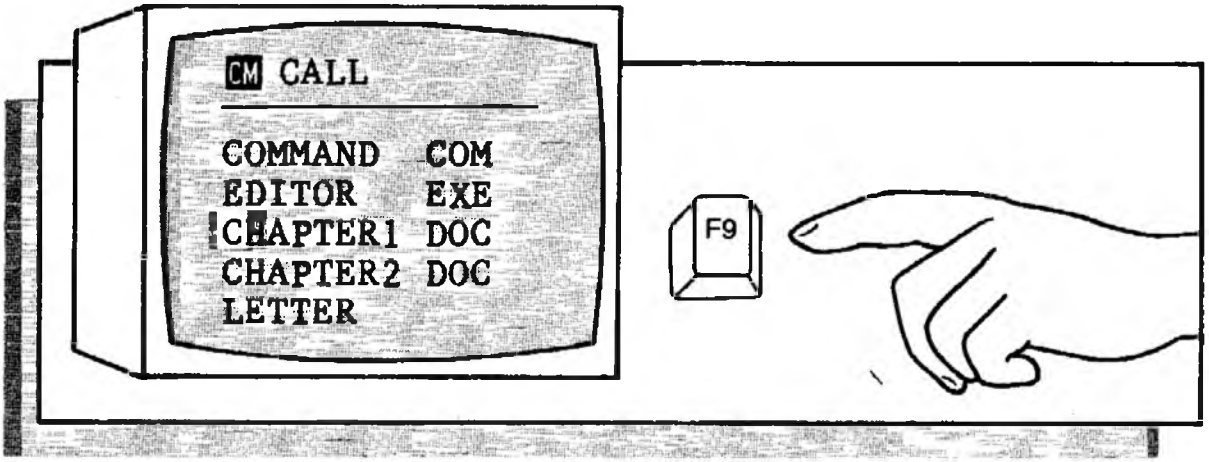
ACTION
(Option 2)

Calling a Document by Pointing at Its Name.
If you don't quite remember the name of the file you want displayed, then use this method.

1. Start with the display cleared of any document. If you need to clear the display, use STORE or ABORT. (Alternatively, you can use CALL to open the document in a second window, by following the procedure given in Note #4.)
2. Call the directory for the drive you want (drive B, for example):

   Type: F5 dir b: ↵
3. Using the cursor up and down keys, move the cursor onto the desired filename.
4. Type: F5 call ↵

Result: This executes the CALL command and displays the document.



NOTE #4 **Automatic New Window.** You can set XyWrite to open a new window automatically (if the current window has a file in it) whenever you enter DIR or CALL. You do this by putting an NW (New Window) setting in the Printer File:

nw=0 CALL and DIR do *not* open a new window.
nw=1 CALL, DIR or NEW opens a new window (if a file is already open in the current window).

This setting also resets the window whenever you issue an Abort command to clear the screen. The default is nw=0.

NOTE #5 **Other ASCII Files.** You can call files from other word processors to the display if they are strictly ASCII text. Wordstar requires a conversion; phone XyQuest for the conversion procedure.

ALSO SEE **Related Commands.** Two other related commands call special kinds of files: CAP (Call Program File), which calls a program file, and CAF (Call Form Document), which calls a form document.

FORMAT CM**CALL** *d:filename*

- *filename* (optional) is the name of the file you want to display. If it is omitted, the previously called file reappears.
- *d:* (optional) is the drive letter (A:, B:, or C:) you specify for the drive you want. If you omit the drive letter, the default drive is used.
- CALL is an immediate command (it is not embedded).
- EDIT is identical to CALL — you may use either term. (EDIT is abbreviated ED).

EXAMPLE CM**call b:chapter.doc**

ABBREV CM**ca b:chapter.doc**

PURPOSE

Three commands are covered here: **CHDIR**, **MKDIR**, and **RMDIR**. These commands are useful to you only if you use subdirectories. (Subdirectories are used mainly on hard disk systems.) The general forms are:

- Changing the Current Directory

  CM **CHDIR** *d:path*
- Making a New Directory

  CM **MKDIR** *d:path*
- Removing a Directory

  CM **RMDIR** *d:path*

Type these commands on the Command Line just as you would any other XyWrite commands. These three commands are described further in your DOS manual. The *path* is described in the Overview section earlier in this chapter.

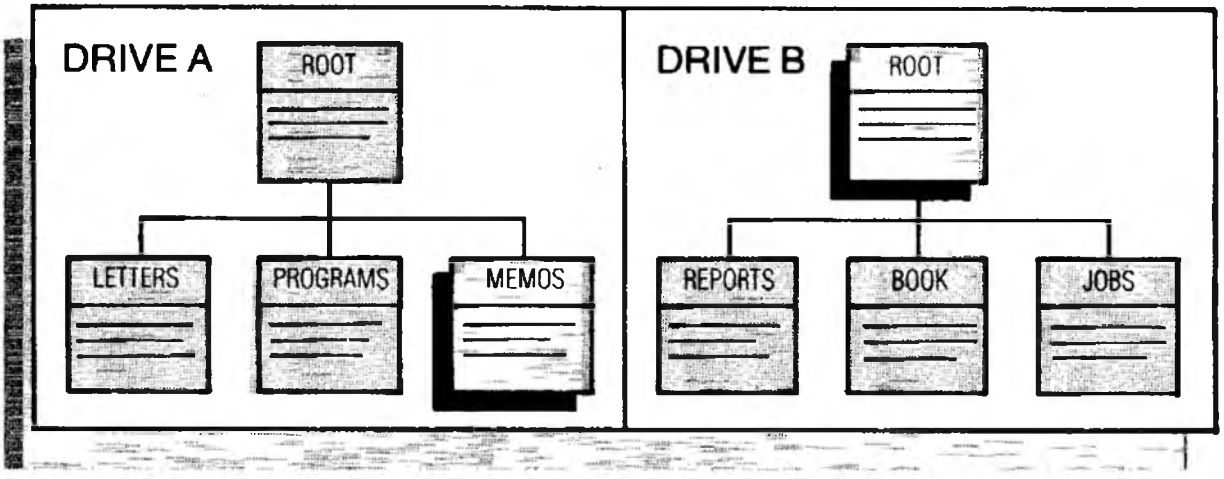
ACTION

Changing the Current Directory.

To change the current directory on drive A to MEMOS, for example:

Type: F5 `chdir a:memos` ↵

Result: The subdirectory MEMOS is now the current directory on drive A. Each drive has its own current directory, as illustrated below.



Another way to change to a subdirectory is to point at its name in a tree:

1. Type: F5 tree b: ↵

   Result: A complete display of all the subdirectories on Drive B appears. CD (Change Directory) appears on the Command Line.

2. Move the cursor to the line containing the subdirectory you want to switch to.

3. Press: F9

Result: The subdirectory you selected is now the current directory on Drive B.

ACTION **Making a New Directory.**

To make a new directory called, say, MEMOS:

Type: F5 mkdir memos ↵

Result: The new directory is created off the current directory of the default drive.

ACTION **Removing a Directory.**

To remove a directory called MEMOS, first delete all of the files from that directory. Then, if necessary, use CHDIR to select a directory other than MEMOS as the current directory. Finally:

Type: F5 rmdir memos ↵

FORMAT

CM **CHDIR** *d:path*

CM **MKDIR** *d:path*

CM **RMDIR** *d:path*

- *d:* is the drive you want to affect (drive A, B or C)
- *path* (optional) is as described in your DOS Manual
- CD is an abbreviation for CHDIR.
- These are all immediate commands.

EXAMPLES

CM chdir a:memos or CM cd a:memos

CM mkdir a:memos

CM rmdir a:memos

PURPOSE

CM (Command) lets you display the *default drive, current path, current window number* and *defined-text indicator* at the top of the screen by replacing the CM, PRMPT and NM fields as follows:

- The **default drive** (C:) in place of CM
- The **current path** (\REPORTS) in place of PRMPT
- The **current window number** and **defined-text indicator** in place of NM

The defined-text indicator is a mark that lets you know if a block of text is defined.

ACTION

Displaying Default Drive, Window Number and Patl

To display the current drive, window number and path in the header:

Type: F5 cm d,w,p ↵

To return all three fields to normal, type: F5 cm ↵

NOTE #1 **Setting One or Two Fields.** Alternatively, you can display any one (or two) of these settings by setting only the fields you want, for example, type:
F5 cm d,p ↵

NOTE #2 **Defined-Text Indicator.** If you use the w option, you will be able to track the status of defining text:

1. Normally the window number is displayed as a number and a blank space: 1
2. When you start defining a block of text using F1, a hyphen appears: 1-
3. When you complete the definition by pressing F1 again or when you press one of the define-unit keys (e.g. F4), an equal sign appears: 1=
4. And when you release the defined block by pressing F3, the blank space appears again: 1

FORMAT CM d,w,p

- d changes the CM to the default drive letter.
- w changes the NM to the current window number.
- p changes the PRMPT to the current path.
- CM with no letters resets to the normal header.
- CM is an immediate command.

EXAMPLE CM cm d,w,p

PURPOSE

The **COPY** command is similar to the COPY command in DOS. You can copy a file and give the copy a new name (Option 1), or you can copy a file to a *different* drive or directory but keep the same name (Option 2).

Unlike DOS, COPY does not recognize wild cards (? or *). You must copy files one at a time. But XyWrite lets you easily select and copy files from a directory (Option 3).

Below we explain these three ways to copy files:

- **Copying a File to a Different Name.** *(Option 1)*
- **Copying a File to a Different Location** *(Option 2)*
- **Copying Files from a Directory.** *(Option 3)*

ACTION *(Option 1)*

Copying a File to a Different Name.

Let's say we want to copy the file MEMO to REPORT:

Type: F5 copy memo,report ↵

Result: Two identical files now exist with different names (in the same directory). Note the comma is optional—you can use a space instead.

ACTION *(Option 2)*

Copying a File to a Different Location.

To copy the file REPORT from the current drive to a different directory on Drive B:

Type: F5 copy report,b:\business ↵

Result: A copy of REPORT is now on Drive B in directory BUSINESS (with the same name).

By leaving off the second drive and filename, you can copy a file from another location to your current directory. For example, to copy REPORT from Drive B:

Type: F5 copy b:report ↵

Result: REPORT is copied into your current directory. (Since a target file isn't named, XyWrite uses the original name.)

NOTE #1

Copying the Current File. To COPY a file that is currently displayed, you must first save it to disk. COPY looks for and copies only files *saved on disk* (rather than the version on your screen). (Use the SAVE and STORE commands to copy the screen version of a file.)

ACTION *(Option 3)*

Copying Files from a Directory.

This procedure makes it easy to copy many files from one disk or directory to another. To copy files from drive A to drive B:

1. Type: F5 b: ↵ (destination drive)
2. Type: F5 dir a: ↵ (source drive)
3. Type: F5 copy (*without* pressing ↵)
4. Type: F10 (puts cursor in the directory)
5. Now move the cursor down the list and stop on the first filename you wish to copy.

   Press: F9 (to execute the COPY command)

Result: The file is copied to Drive B. Note the cursor has moved to the next name. Repeat Step 5 until you have copied all the files you want from Drive A to B. If your disk fills up, you get the message ERROR WRITING TO DISK and the cursor does not move to the next name.

NOTE #2

Duplicate Filename. If the filename you specify for the copy already exists, Xywrite displays the message "File Already Exists, Overwrite It?" Press "Y" if you want the copy you are making to supersede the document that already exists with that filename. Press "N" to abort the COPY command so you can reissue it with a new filename.

If you have been running utilities under DOS that allow you to access files from other directories without specifying the path, you may erroneously receive the "File Already Exists, Overwrite It?" message. If the filename in question does not appear in the current directory, ignore the message and proceed with COPY by pressing Y.

FORMAT

CM **COPY** *d:filename,d:targetfile*

- *d:* (optional) is the drive letter
- *filename* is the name of the file to be copied
- *targetfile* (optional) is the name of the new file. If omitted, the file is copied under the original name.
- COPY is an immediate command

EXAMPLE

CM copy b:chapter.doc,c:

PURPOSE

DIR (Directory) displays the names of files on the drive you specify. (If you use subdirectories, DIR lists only the files in the subdirectory and drive you specify. TREE shows you all subdirectories on a disk.)

You've got two options: you can display *all* the files, or just the files you want. The second option is very handy if your directory has more than a screenful of files.

- Displaying a Complete Directory. *(Option 1)*
  CM **DIR** *d:*
- Displaying a Partial Directory. *(Option 2)*
  CM **DIR** *d:globalname*

ACTION *(Option 1)*

Displaying a Complete Directory.

To display the names of all files in the current directory on the default drive:

1. Start with the display cleared of any document. (Either press Alt F10 to open a second window or STORE the current file).
2. Type: F5 dir ↵

Result: The list of filenames resembles that in the illustration on the following page.

You can eliminate Step 1 with a setting that automatically opens a new window for DIR or CALL. See the note "Automatic New Window" earlier under the CALL command.

To display the names of files on another drive, say drive B (when the default is drive A), you would type the following:

Type: F5 dir b: ↵

ACTION *(Option 2)*

Displaying a Partial Directory.

To display a *partial* list of files in the directory, use DIR with a global filename. (Global filenames are described in detail right after the Notes section.) For example, to list all files which begin with the letter f, you would:

Type: F5 dir f*.* ↵

Result: All files on the default drive beginning with the letter f are listed. The global filename here is f*.* .

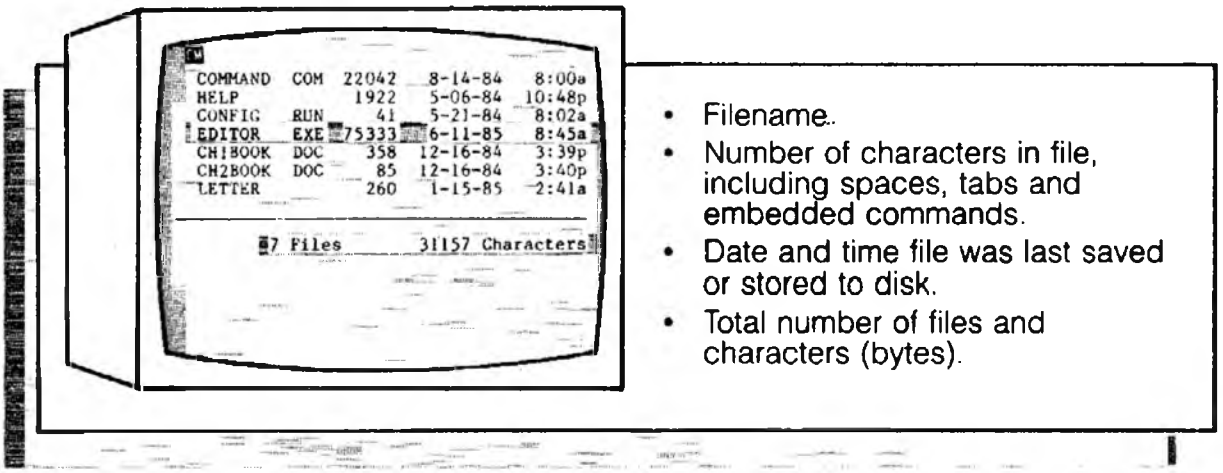
NOTE #1 **Keeping the Directory Up-To-Date.** Whenever you execute DIR, you get a list current to that moment. That list is not automatically updated when you delete files, for instance, from the command line. You must execute the DIR command again to update the list.

NOTE #2 **Saving the Directory.** You can save the directory to a file on disk. After you type DIR to display the directory, then type SAVE or STORE. The directory is saved to a file named DIRECTRY.TMP (unless you specify a different name). (Notice there is no "O" in the filename - this keeps it to eight letters.)

NOTE #3 **Printing the Directory.** You can easily print out the directory. Type DIR to display the directory, then enter TYPE.

NOTE #4 **Sorting the Directory.** The directory defaults to listing files alphabetically by filename. Use the DSORT command to sort by extension, date, or to sort in reverse order.

NOTE #5 **Viewing Subdirectories.** If you use subdirectories, refer to the TREE command, later in this chapter, to see how to display *all* the subdirectories on a drive. DIR shows only the current directory.



DETAIL

Global Filenames.
A global filename includes either (or both) of the characters **?** (question mark) and * (asterisk). You use a global filename when you want to list a partial directory.

The Asterisk (*).

- When used with DIR, an asterisk (*) means that *any* character(s) can occupy the remaining positions in the filename or extension.
- If the asterisk appears *alone* in the filename or extension, read the asterisk as the phrase "all files".
- If any characters precede the asterisk, read it as the phrase "all files that begin with . . ." those characters.
- The asterisk represents any number of characters, from one to eight.

The Question Mark (?).
While an asterisk represents any number of characters, the question mark (?) represents only one. Use it in a filename or extension when executing the DIR command to indicate any character can occupy that position. Question marks can appear in any combination in a global filename. For example: CM chap?.doc would find CHAP1.DOC and CHAP2.DOC but not CHAP10.DOC because 10 is two characters, not one.

You can mix the question mark and asterisk in a single filename. For example: dir ch?book.*

FORMAT

CM **DIR** *d:* *(Option 1)*

CM **DIR** *d:globalname* *(Option 2)*

- *globalname* (optional) is the global filename which generates the partial list you want.
- *d: (optional)* is the drive letter (A:, B:, or C:), to specify the disk drive you want listed. If you omit the drive letter, the default drive is used.
- DIR is an immediate command.

EXAMPLES

CM dir b:

CM dir b:ch?book.*

PURPOSE

The **DIRL** (Long Directory) command lists the names of files in a directory along with the first few lines of text in each file. DIRL enables you to browse through your files.

Also covered here is the DI setting, which lets you modify how the long directory is displayed.

ACTION

Displaying a Directory Along with Lines of Text.
To display a directory along with the first lines of text from each file (for filenames ending in .DOC for example):

Type: F5 dirl *.doc ↵

Result: A list appears, showing all filenames in the current directory ending in .DOC, and including lines of text under each filename. It might look like this:

```
CHAPTER DOC              122      1-22-86 2:31p
In the beginning there were a few furry little creatures.  They were
a gregarious type of animal that loved human company.

STAFF DOC                3291     1-22-86   2:29p
Memo:       Vacation Time
To:         Employees
From:       Tom
Date:       Jan. '87

-------------------------------------------------

  2 Files    3413 Char.    3805184 Free
```

NOTE #1

The exact number of text lines displayed will vary somewhat from file to file, due to program rules that attempt to cut off displayed text at a sensible point (such as a period).

ACTION

Modifying the Long Directory.

XyWrite lets you control three facets of the DIRL display: 1) show the file size in kilobytes (rather than bytes), 2) change number of text lines displayed, and 3) pack the lines of text (by removing carriage returns). The purpose of packed text is to let you view more of the file in a few lines (especially if the file starts out with blank lines).

To change these settings, put the DI setting in the Printer File. Its format is:

di=*k,l,p*

k is a number the file size is to be divided by (typically set k to equal 1024). The default for k is 1.

l is the approximate number of lines of text displayed. The default for l is 3.

p packs the text. Set p to 1 to remove carriage returns; 0 to keep carriage returns. The default is 0.

To display the file size in kilobytes (KB) and to display about 4 lines of packed text, add this setting to your Printer File:

```
di=1024,4,1
```

(See "Printer File" in Chapter 6 for information on how to change Printer File settings.)

In the previous example, 3291 is divided by 1024 and rounded up to 4 kilobytes. The entry would look like this — the text appears disordered since it is packed:

```
STAFF DOC          4          1-22-86   2:29p   Memo:
Vacation Time                 To: Employees     From:
Tom    Date: Jan. '87  We are proposing that each employee be
eligible for 18 paid days each calendar year.  In addition to the
major national holidays, each
```

FORMAT

CM **DIRL** *d:globalname*

- *globalname* (optional) is the global filename (used with DIR)
- *d:* (optional) is the drive letter.
- DIRL is an immediate command.

EXAMPLE

```
CMdirl c:*.doc
```

PURPOSE

DSORT (Directory Sort) sets the order that filenames are displayed in directories. Once you set DSORT, it affects the listing of all directories (on all drives) and stays in effect until you change it.

You can sort by filename, extension, size or date. You can sort in forward or reverse order. You can add a one-line header to the directory, which shows the name of the directory.

ACTION

Sorting the Directories

To set the order in which filenames are sorted:

Enter DSORT along with one or two of the sort parameters F, D, E, S. (When you include two parameters, the first takes precedence over the second). To *reverse* the order, add an R. To add a header, add an H. Separate all parameters with commas. For example:

Type: F5 dsort f,e,r,h ↵

Result: This DSORT setting means that whenever you type DIR, the filenames are listed by filename (F) in reverse order (R), from Z to A. Two files having the same filename are in turn sorted by extension (E). A header (H) is included at the top of the list.

EXAMPLES

Examples of Sorting Directories. These examples illustrate the different ways you can sort directories.

F5 dsort f ↵	Sorts by filename, from A-Z.
F5 dsort f,r ↵	Sorts by filename, from Z-A.
F5 dsort e ↵	Sorts by extension, A-Z.
F5 dsort f,h ↵	Sorts by filename, A-Z, with a header.
F5 dsort f,e,r ↵	Sorts by filename in reverse order (Z-A) and then by extension, also in reverse order.

F5 dsort f,e,r,h ↵	Same as the previous example, but with a header added.
F5 dsort e,f ↵	Sorts by extension, then filename.

NOTE #1 **Setting The Sort Order.** By adding the DSORT command as a line in your STARTUP file, you can automatically set up the order when you load XyWrite. See the Startup File in Chapter 6.

NOTE #2 **XyWrite Default.** If you give no DSORT command in the STARTUP.INT file, the default is DSORT F (alphabetically by filename with no header).

NOTE #3 **Clearing the Sort Key.** If you give the DSORT command alone, as follows, then the directory is displayed as it would be in DOS. (This is the order that files are recorded on sectors on the disk.)

Type: F5 dsort ↵

NOTE #4 **How DSORT Works.** DSORT sorts the filenames as they are read off the disk — it does *not* rearrange the filenames as recorded on the disk.

FORMAT CM **DSORT** *order,modifier* Directory Sort

- *order* is any one or two (separated by commas) of:
  - f sort by filename
  - e sort by extension
  - s sort by size
  - d sort by date and time
- *modifier* (optional) is either or both of:
  - r sort in reverse order
  - h add a header to top of directory

EXAMPLE CM dsort f,e,r,h

PURPOSE

DO lets you run a program under XyWrite — programs such as Lotus 1-2-3, DBase III, WordProof, BASICA, and others (memory permitting). When you quit the program, control returns to XyWrite, exactly as you left it. The DO command works only with programs whose filenames have the .COM or .EXE extensions. The simplified form of this command is:

CM **DO** *d:program filename*

ACTION

Running Programs Under XyWrite.
This example would first load BASIC, then would automatically run the BASIC program called TRAINER.

Type: F5 do basic trainer ↵

Result: When you are done with BASIC (that is, when you type the word SYSTEM), control returns to XyWrite, exactly as you left it.

NOTE #1

DOS Requirements. The DO command requires DOS 2.0 or higher. This command also has a memory requirement: there must be enough memory to load the desired application in addition to XyWrite and any open XyWrite files.

NOTE #2

Path is Allowed. If you use subdirectories, you can include a path in the DO command:

CM **DO** *d:path\program filename*

This means that you can access programs in other directories without switching directories. A special case of this is Note #3.

NOTE #3

Running DOS Commands. You can run DOS internal commands, external commands and batch files with the DO command. This command is equivalent to the XyWrite DOS command (described next) with one exception — if you use subdirectories, you'll find DO more powerful than the DOS command because DO allows you to specify a path. Its form is:

CM **DO** *d:path*\COMMAND /C *command*

The term COMMAND refers to COMMAND.COM. The term *command* refers to any DOS comand you want to use. In fact, at this point you can enter *any* command (along with its arguments) that is allowed at the DOS prompt. For example, if you are currently working on drive B, you can use COMMAND.COM in the root directory of drive A (A:\) as follows. Here are two typical uses.

```
CMdo a:\command /c chkdsk a:
CMdo a:\command
```

The last example switches to DOS; to return, type EXIT at the DOS prompt (A>).

If you have a floppy drive system with extra memory, it's a good idea to create a RAM (virtual) drive and copy COMMAND.COM into it. If the RAM drive is drive D, you would then specify **do d:\command**. This gives you immediate access to DOS. However, in order for DOS to recognize this COMMAND.COM, you must include in your AUTOEXEC.BAT file the command (for this example):

```
set comspec=d:\command.com
```

FORMAT **CMDO** *d:path\program filename*

- ***program*** is the program you want to run. The program must have a .EXE or .COM extension. Omit the extension when entering the name.
- ***filename*** (optional) is the file you want to run once the program is up and running.
- *d:* (optional) is the drive letter (A:, B:, or C:) you specify for the drive you want.
- *path* (optional) is the sequence of directory names from the current or root directory to the program. Further described in your DOS manual.

EXAMPLE

```
CMdo a:basic trainer
```

PURPOSE

The **DOS** (Disk Operating System) command suspends XyWrite, then loads and runs DOS. With Option 1, the familiar DOS prompt (A> or B>) appears, allowing you to run any DOS commands or programs you wish. When done, you type EXIT to return to XyWrite, exactly as you left it. XyWrite is *frozen* in place while in DOS.

You have two options. Option 1 is as described above. Option 2 runs just a single DOS command *or batch file*, and immediately returns control to XyWrite when done. (In Option 2, DOS /C stands for "DOS Command".) Use Option 2 when you want to run a DOS command from a program file *(User Programming)*.

- Running DOS Under XyWrite. *(Option 1)*
  CM **DOS**

- Running a DOS Command or Batch File. *(Option 2)*
  CM **DOS /C** *command*

ACTION *(Option 1)*

Running DOS Under XyWrite.
To suspend XyWrite and run DOS:

1. Select the drive that contains COMMAND.COM to be the default drive (if not already set). For example, if COMMAND.COM is on drive A:

   Type: F5 a: ↵

2. Type: F5 dos ↵

   Result: The display clears and the DOS prompt (A> or B>) appears. (If you use subdirectories, COMMAND.COM must be in the current directory for this to work.)

3. Now that you have entered DOS, you are fully in DOS and can use any of its commands (except MODE, PRINT, and GRAPHICS — you should not run any program that remains resident after execution). For instance, you can copy disks, check disks (CHKDSK) or change the system date (DATE). In addition, with DOS 2.0 or higher, you can run most any program, such as Lotus 1-2-3, DBase III, or BASIC.
4. To return to XyWrite, at the A> prompt:

   Type: exit ↵

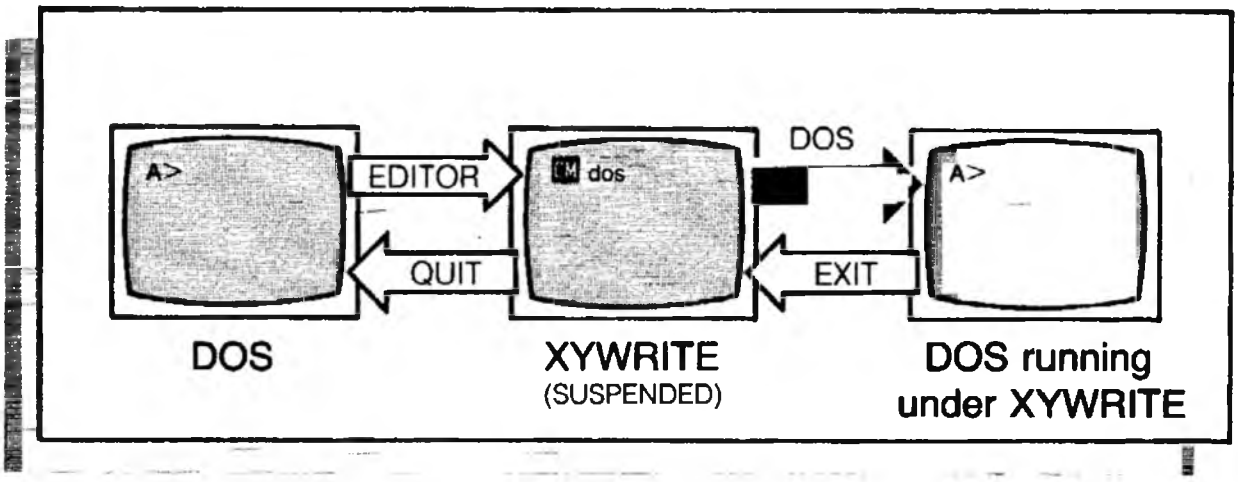
NOTE #1 **XyWrite is Suspended.** To demonstrate how XyWrite truly is suspended while control is with DOS: If you are in the middle of printing a document when you enter DOS, the printing stops; then, when you return to XyWrite, printing continues exactly where it left off.

ACTION *(Option 2)* **Running a DOS Command or Batch File.** To run only a single DOS command or batch file (for example, DATE)

1. Make sure the default drive has COMMAND.COM on it (here again we'll assume it's on drive A):

   Type: F5 a: ↵

2. Type: F5 dos /c date 8-13-87 ↵



Result: The display switches to DOS, the command (DATE) is executed, and then control is immediately returned to XyWrite. All DOS commands can be run using this method (except those that remain resident, such as MODE, PRINT, and GRAPHICS).

NOTE #2 **DOS Requirement.** COMMAND.COM must be present in the current directory of the default drive in order to use the DOS command.

NOTE #3 **Stopping in DOS**. To stop in DOS in order to review the results of a command, create a batch file and include PAUSE in it. For instance:

```
chkdsk %1
pause
```

NOTE #4 **Subdirectories.** If you use subdirectories, refer to the note "Running DOS Commands" under the DO command. It allows you to specify a path with DOS commands.

NOTE #5 **DOS vs. DO.** The following shows the XyWrite DOS command and the corresponding DO command. You can see how the DO command allows you to specify a drive and path, while the DOS command does not.

- Running DOS Under XyWrite.
  CMDOS
  CMDO *d:path*\COMMAND

- Running a DOS Command or Batch File.
  CMDOS /C *command*
  CMDO *d:path*\COMMAND /C *command*

ALSO SEE

Related Commands. The following DOS commands are found elsewhere in this section, and can be executed from the command line on their own: A:, APPEND, COPY, DEL, DIR, ERASE, CHDIR, MKDIR, RENAME and RMDIR.

The QUIT command also switches control to DOS, but quits XyWrite altogether. Unlike the DOS command, any changes to a document which have not been saved are lost.

FORMAT

CMDOS *(Option 1)*

CMDOS /C *command* *(Option 2)*

- *command* (optional) is any DOS command or batch file.
- DOS is an immediate command.

EXAMPLE

CMdos /c format a:

PURPOSE

EDITOR loads and runs XyWrite from DOS. You use EDITOR when you first start up XyWrite, as the illustration shows.

If you want, you can type in a filename after the word EDITOR — that file will be loaded into the display once XyWrite is running. The two choices are:

- Running XyWrite. *(Option 1)*
  A>**EDITOR**

- Running XyWrite and Calling a File *(Option 2)*
  A>**EDITOR** *d:filename*

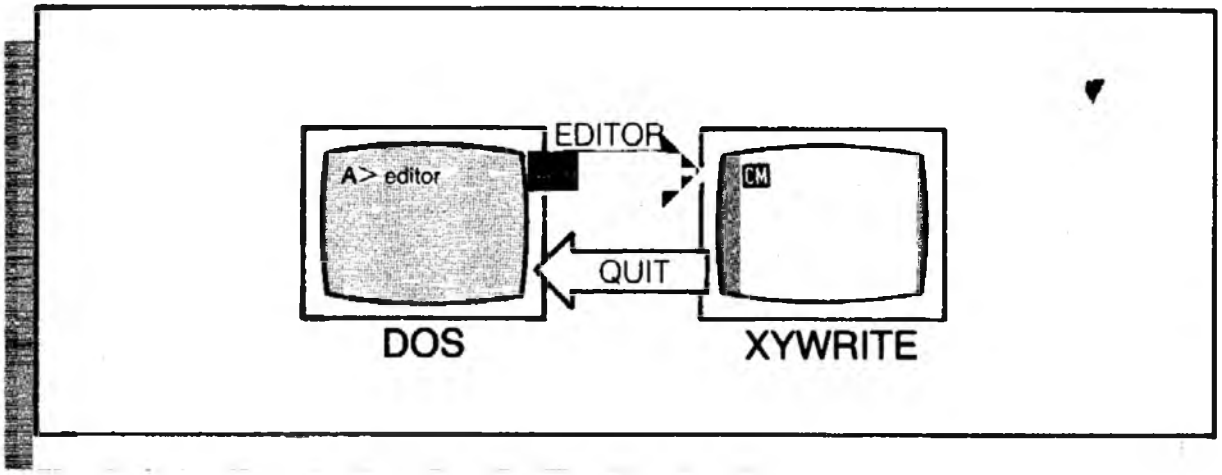
ACTION
(Option 1)

Running XyWrite

Start in DOS at the A> prompt. To run XyWrite:

Type: editor ↵

Result: XyWrite is loaded and displayed, the file STARTUP.INT is run, and finally the serial number screen appears. Press any key to clear the message from the display.



ACTION *(Option 2)*

Running XyWrite and Calling a File

Start in DOS at the A> prompt. To run XyWrite and call a file, type EDITOR followed by the filename:

Type: editor chapter.doc ↵

Result: XyWrite is displayed, and the file CHAPTER.DOC is loaded into the display (from the default drive).

TIP

Renaming Editor. You can use the RENAME command to change the EDITOR.EXE file to some other name, such as XY.EXE. Then, to start XyWrite, at the A> prompt, type XY.

ALSO SEE

Related Commands. The QUIT command switches control back to DOS when you are finished using XyWrite.

The STARTUP.INT file automatically initiates XyWrite when EDITOR is used. You can edit this file to create your own default configuration.

FORMAT

A>**EDITOR** *(Option 1)*

A>**EDITOR** *d:filename* *(Option 2)*

- EDITOR is entered at the DOS prompt, unlike most commands in this manual.
- *filename* is the name of the file you want called to the display once XyWrite is running.
- *d:* (optional) is the drive letter (A:, B:, or C:) where the filename is located. If you omit the drive letter, the default drive is used.
- EDITOR is an immediate command.

EXAMPLES

A>editor *(Option 1)*

A>editor b:chapter.doc *(Option 2)*

PURPOSE

ERASE deletes the named file from the specified drive. It does not affect the display or memory. (The terms *erase* and *delete* are used interchangeably.) The general form is:

CM **ERASE** *d:filename*

You have a choice of two different ways to delete a file (the result is the same either way):

- Erasing a File by Typing Its Name *(Option 1)*
- Erasing a File by Pointing at Its Name *(Option 2)*

ACTION *(Option 1)*

Erasing a File by Typing Its Name.

To erase a file:

Type: F5 erase chapter.doc ↵

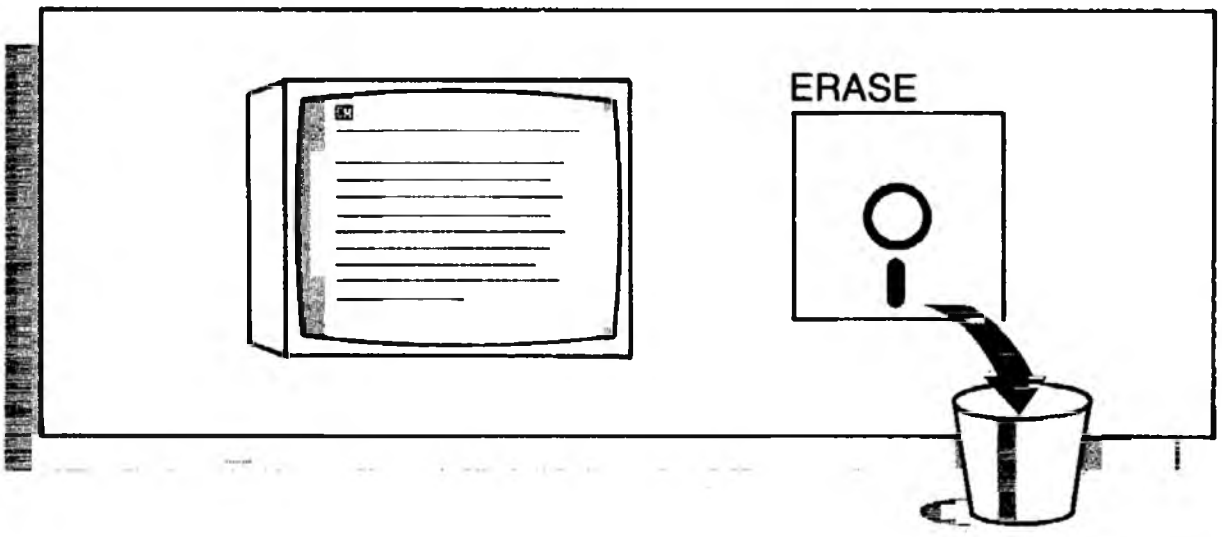
Result: This command erases the file CHAPTER.DOC from the default drive (no drive letter is given).

ACTION *(Option 2)*

Erasing a File by Pointing at Its Name.

There are three steps: (1) Call the directory for the drive you want (drive B, for example), (2) point the cursor to its filename and (3) erase it.

1. Type: F5 dir b: ↵
2. Type: F5 erase (*without* pressing ↵)
3. Type: F10



Now move the cursor down the list and stop on the filename you wish to delete.

Press: F9

Result: The file is erased from the disk. You can continue down the list and delete other files.

Do not hold down the F9 key! The autorepeat would issue several delete commands before you could see their effect on the screen. In such a case, to stop more files from being deleted, press Ctrl Break.

NOTE #1 **Erase Prompt.** You can reduce the risk of inadvertently erasing files with ERASE (or DEL) by instructing XyWrite to prompt you with "Do you wish to erase? (Y/N)". Pressing "Y" will then erase the file. To enable this prompt to display, add the following setting to the Printer File:

```
ep=1
```

To disable this prompt, change the setting to ep=0. See Chapter 6 for information on how to change Printer File settings.

If you set up XyWrite with ep=1, you can bypass the prompt by using ERNV (Erase, No Verify) instead of ERASE or DEL.

NOTE #2 **Recovery of a File.** Once a file is deleted, it cannot be recovered easily. The best bet is to make backup copies often. (See Option 2 of "Setting the Default Drive.") To recover a file, you must use a utility program designed specifically to un-erase DOS files.

FORMAT CM **ERASE** *d:filename*

- *filename* is the name of the file you want to erase.
- *d:* (optional) is the drive letter (A:, B:, or C:) you specify for the drive you want. If you omit the drive letter, the default drive is used.
- ERASE is an immediate command.
- DEL is identical to ERASE — use either term.

EXAMPLES

```
CMerase b:chapter.doc
CMdel b:chapter.doc
```

PURPOSE

We have described how you can set up XyWrite to reduce the risk of inadvertently erasing files — so that whenever you use ERASE (or DEL) to erase a file, you will get the message "Do you wish to erase?" (see Note #1 under ERASE). If this message gets in your way, use ERNV instead of ERASE.

The **ERNV** (Erase, No Verify) command allows you to delete files from your disk *without* getting any prompt. You might want to use ERNV to delete a long list of temporary files, for example. There are two ways to use ENRV:

- Erasing a File By Typing Its Name *(Option 1)*
- Erasing a File By Pointing at Its Name *(Option 2)*

ACTION *(Option 1)*

Erasing a File by Typing Its Name.

To erase a file without receiving the verifying prompt, type ERNV just as you would ERASE. For example:

Type: F5 ernv b:chapter.tmp ↵

No verifying prompt is displayed, even if the Erase Prompt is set in the Printer File (EP=1).

ACTION *(Option 2)*

Erasing a File by Pointing at Its Name.

To erase a list of files, you need to erase them one at a time. For example, to delete files with extension .TMP from drive A:

1. List a group of files to delete: F5 dir a:*.tmp ↵
2. Type: F5 ernv (without pressing ↵)
3. Move the cursor to the filename and press: F9

Result: The file is erased from the disk. Repeat Step 3 until you have deleted each of the temporary files.

FORMAT

CM **ERNV** *d:filename*

- *filename* is the name of the file you want to erase.
- *d:* (optional) is the drive letter you specify for the drive.

EXAMPLE

CM ernv chapter.tmp

PURPOSE

EXIT returns you to XyWrite from DOS as shown in the illustration below. You type EXIT at the DOS prompt (EXIT is not a command you can type at the XyWrite command line). EXIT works only if DOS is running under XyWrite — that is, if DOS was entered using the DOS command (described earlier in this section). The form is:

A>EXIT

ACTION

Returning to XyWrite from DOS.
Start in DOS at the A> prompt. To return to XyWrite:

Type: exit ↵

Result: XyWrite is displayed and control is resumed where you left off; see the illustration.

NOTE

EXIT vs. EDITOR. Notice that you use EXIT rather than EDITOR to re-enter XyWrite. EDITOR would unnecessarily load and run a second copy of XyWrite.

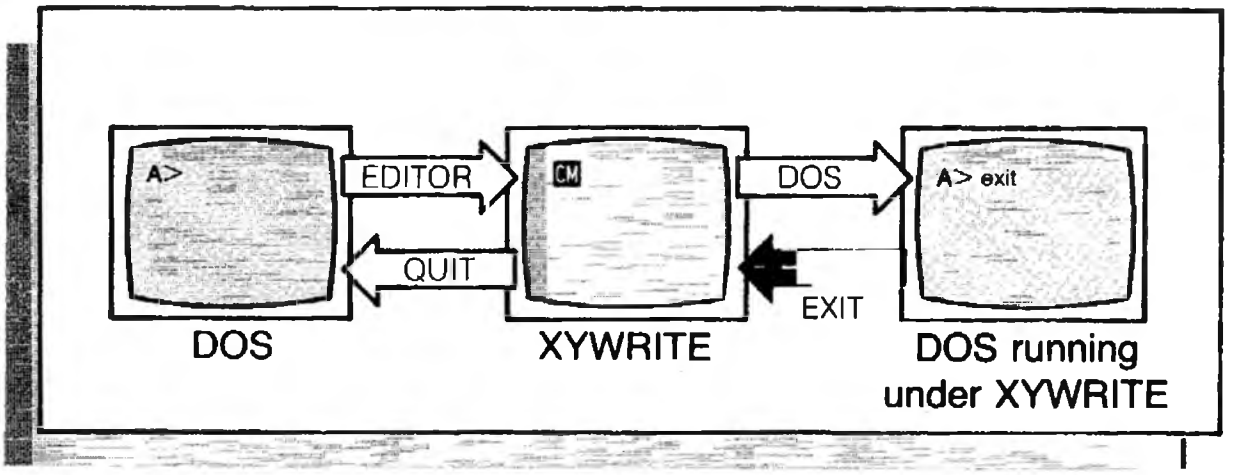
ALSO SEE

The DOS Command. EXIT is used in conjunction with the XyWrite DOS command.

FORMAT **A>EXIT**

- EXIT is entered at the DOS prompt, unlike most commands in this manual.
- EXIT is an immediate command.

EXAMPLE **A>exit**



PURPOSE

The **FIND** command searches through all files on a disk for a file you specify. FIND is especially useful if you use subdirectories (such as on a hard disk) — FIND searches through *all* subdirectories and lists all occurrences of that filename.

This command fills several needs. The most obvious is to locate a file anywhere on your disk. Since FIND displays *all* files with the same name, it will reveal if you've saved the same file in more than one subdirectory.

FIND also uses wild card symbols to find sets of files. For example, FIND *.* displays a list of *every* file on a disk.

ACTION

Finding a File.
To find a file on a given disk, say CHAPTER1 on drive C, start at an empty window, then:

Type: F5 find c:chapter1 ↵

Result: All files named CHAPTER1 will be listed:

```
\NOVEL\CHAPTER1        14840    9-29-86   1:49a
\TEXTBOOK\CHAPTER1     23216    7-17-86   11:22p
```

Note that FIND does *not* sort the files. The CALL command appears on the command line — to call up a file, simply move the cursor to that filename and press F9

ACTION

Finding Groups of Files Using Global Filenames.
To find files with similar names, use the same wild card symbols you use with the DIR command: question mark (?) and asterisk (*). For example:

Type: F5 find a:chapter*.* ↵

Result: All files on drive A starting with CHAPTER will be displayed.

FORMAT

CM **FIND** *d:filename*

- *d:* (optional) is the drive letter
- *filename* is the file or group of files you want to list, and can include wildcards ? and *.

EXAMPLE

CM find c:chapter.doc

PURPOSE

KILTYP (Kill Type) stops the outputting of a document to a printer (TYPE). It also stops the printing of a document to a file (TYPEF).

ACTION *(Option 1)*

Stopping the Printout of the Current Document.
To kill the printing of a file:

Type: F5 kiltyp ↵

Results: This causes XyWrite to *immediately* stop sending text to the printer. However, the printer will continue printing until its text buffer (or the print spooler) is empty. (Some printers have no buffer and will stop immediately.) If you wish, you can stop the printer by turning it off and back on; however, if you use continuous feed paper, you must then adjust the paper so its top edge is in line with the print head.

ACTION *(Option 2)*

Clearing the Printer Queue.
If you have more than one file queued to the printer (by issuing more than one TYPE command), to kill the printing of the current file and to clear the printer queue of all other files waiting to be printed:

Type: F5 kiltyp q ↵

TIP

Kill and Re-Start. If you have a printer problem and must use KILTYP to stop a file, you can re-issue the TYPE command starting from the page at which the printing stopped. For example, to start CHAPTER.DOC at page 5, type:

F5 type chapter.doc,5- ↵

FORMAT	CM **KILTYP**	*(Option 1)*
	CM **KILTYP Q**	*(Option 2)*

- Q (optional) kills *all* files waiting in the printer queue.
- KT is the abbreviation for KILTYP.
- KILTYP is an immediate command.

EXAMPLE	CM kiltyp	*(Option 1)*
	CM kiltyp q	*(Option 2)*
ABBREV	CM kt q	

PURPOSE

MERGE copies the entire text of the file you specify into the document you are working on. The text is inserted at the cursor location. Its general form is:

CM**MERGE** *d:filename*

The obvious application for MERGE is to copy one entire document into the one you're working on. However, this command is also great for inserting so-called *boilerplate* text into a file — that is, text which is used repeatedly over and over. Keep each boilerplate section in a separate file. (In this respect, MERGE is like a Save/Get, except here you have the advantage of being able to go directly into the file to edit the text.)

ACTION

Merging One Document into Another.
To copy a file named BOILER.PLT from drive B into the document you are working on:

1. Move the cursor to the position where you want the text to be inserted.

2. Type: F5 merge b:boiler.plt ↵

Result: This command copies the contents of BOILER.PLT from drive B into the document currently open. The text is inserted at the cursor location in the text field. The cursor finishes up at the end of the inserted text.

FORMAT CM**MERGE** *d:filename*

- *filename* is the file to be copied into your document
- *d:* (optional) is the drive letter (A:, B:, or C:) you specify for the drive you want. If you omit the drive letter, the default drive is used.
- MERGE is an immediate command (it is not embedded).

EXAMPLE CMmerge b:chapter.doc

ABBREV CMme b:chapter.doc

PURPOSE

NEW creates a new document with the name you specify. (This new document is created in memory — see Note #1.) As an option, when you create a new document, you can choose to copy an existing file into it at the same time. The formats for these two choices are:

- Creating a New Document
  CM**NEW** *newfile* *(Option 1)*

- Creating a New Document and Copying Into It
  CM**NEW** *newfile,existingfile* *(Option 2)*

ACTION *(Option 1)*

Creating a New Document.
To create a new (empty) document:

1. Start with the display cleared of any document. To clear the display, use STORE or ABORT to make room for the new document. (Alternatively, you can use NEW to open the document in a second window by following the procedure given below in Note #2.)

2. Decide on a name for your new document. For example, CHAPTER.DOC. (For rules on allowable filenames, see "Naming a File" which follows.)

3. Type: F5 new chapter.doc ↵

Result: This example allows you to begin typing a new (empty) document called CHAPTER.DOC into memory. XyWrite will not create the document if one with the same name already exists on the disk. (The new file is not actually created *on disk* until you SAVE or STORE it.)

ACTION
(Option 2)

Creating a New Document and Copying Into It

To create a new document and copy an existing file into it:

1. Start with the display cleared of any document. If you need to clear the display, use STORE or ABORT. (Alternatively, you can use NEW to open the document in a second window, by following the procedure referred to in Note #2.)
2. Decide on a name for your new document. For example, CHAPTER.DOC. Also decide which existing file you would like to copy into the new one, say LESSON.ONE.
3. Type: F5 new chapter.doc,lesson.one ↵

Result: This example creates a new document named CHAPTER.DOC, and copies the existing file named LESSON.ONE into it. (Notice there is no space after the comma.)

FORMAT

CM **NEW** *d:newfile* *(Option 1)*

CM **NEW** *d:newfile,d:existingfile* *(Option 2)*

- ***newfile*** is the name of the new document being created.
- ***existingfile*** (optional) is the name of the existingfile to be copied (merged) into the new document.
- ***d:*** (optional) is the drive letter (A:, B:, or C:) you specify for the drive you want. If you omit the drive letter, the default drive is used.
- NEW is an immediate command (it is not embedded).

EXAMPLE

```
CM new b:chapter.doc
```

ABBREV

```
CM ne b:chapter.doc
```

NOTE #1 **Saving the New Document.** NEW creates the new document in memory, not on the disk. The document is not saved on disk until you SAVE or STORE it. (Therefore, if you ABORT a new document without first storing or saving it, the document is lost.)

NOTE #2 **Creating a Second Document.** You can also use NEW to display a second document alongside the first. The procedure is only slightly different — see "Windows" in Chapter 3.

NOTE #3 **Naming a File**. The rules for naming a file in XyWrite are the same as they are in DOS. The general format is:

filename.ext

- *filename* is from one to eight characters in length
- *.ext* (optional) is an extension, which can be one to three characters in length. There are no reserved extensions in XyWrite.
- Valid characters include:
  Letters A-Z, numbers 0-9
  ! @ # $ % ^ & ( ) _ - { } ' '
- Invalid characters are:
  * + = , . ? " / \ [ ] ; : | < >

  and ASCII characters less than 32.

Any place in this manual where a filename appears, you can preface the name with a drive letter (A:, B:, or C:). (If the drive letter is left off, the default drive is used.) For example, drive B is specified:

Format *d:filename.ext*

Example b:chapter.doc

Reserved Names. Certain names are reserved for special uses and cannot be used as filenames: STARTUP.INT, AUX, CON, COM1, COM2, LPT1, LPT2, LPT3 and PRN.

PURPOSE

The **RENAME** command lets you change the name of a stored file. It is identical to the RENAME command in DOS.

ACTION

Renaming a Document.
To change the name of a file on disk, use RENAME. For example, RENAME from MEMO to REPORT (on drive A):

Type: F5 rename a:memo,a:report ↵

Result: The file is changed from MEMO to REPORT. (You will no longer find the name MEMO in the directory.)

If you do not specify a drive, XyWrite assumes the file you are renaming is located in the current drive and directory. The drive and directory must be identical for both old and new filenames; RENAME cannot be used to transfer a document to a different directory or drive.

NOTE

Optional Comma. You can enter RENAME without the comma, like you do in DOS:

F5 rename a:memo a:report ↵

FORMAT

CM **RENAME** *d:oldname,d:newname*

- *d:* (optional) is the drive where the document is stored
- *oldname* is the filename to be changed
- *newname* is the new filename

EXAMPLE

CM rename sample.doc,chapter.doc

ABBREV

CM ren sample.doc,chapter.doc

PURPOSE

The **SAVE** command saves the displayed document to a file on disk. Unlike STORE, it does *not* clear the file from the display. There are two ways to save a document, the first being the one you typically use:

- Saving a Document Under Its Own Name *(Option 1)*
  CM **SAVE**

- Saving a Document Under a Different Name
  CM **SAVE** *d:filename* *(Option 2)*

You can save *part* of your document with SAVEDEF (Save Defined Block). You define a block of text, and then use SAVEDEF instead of SAVE:

- Saving a Block of Text
  CM **SAVEDEF** *d:filename* *(Option 3)*

ACTION *(Option 1)*

Saving a Document Under Its Own Name.
To save the displayed document under its own name to where it came from:

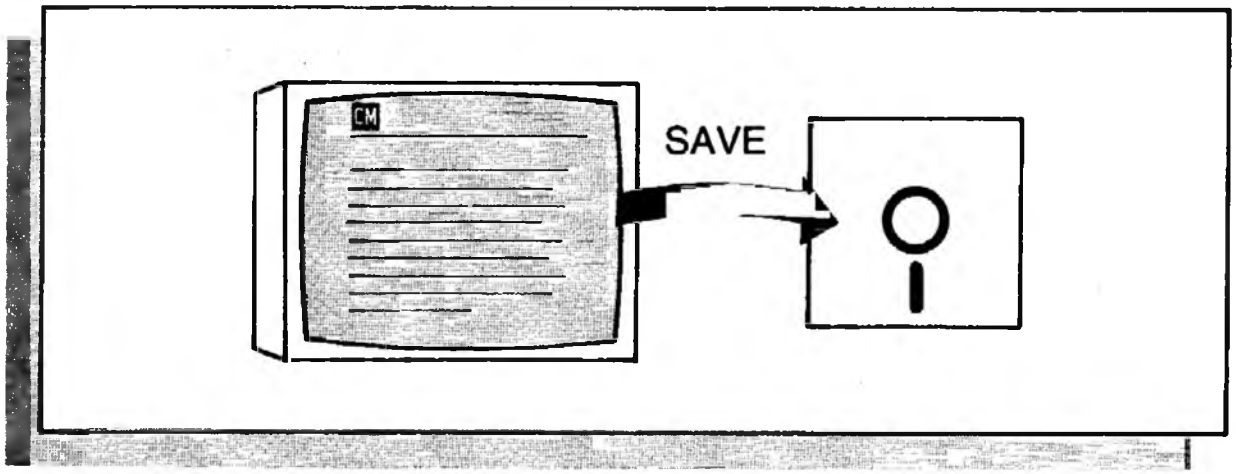
Type: F5 save ↵

Result: The document is saved to where it came from. Because you have *not* specified a drive, directory or filename, the document is saved to the name at the top center of the display.

NOTE #1

Saving to Another Drive. If you want to save the displayed document to another drive but keep the same name, you can save with just the drive letter:

Type: F5 save b: ↵



ACTION *(Option 2)*

Saving a Document Under a Different Name.
When you want to save a document under another name, include that filename in the command. For example:

Type: F5 save b:exercise.doc ↵

This example saves the displayed document to drive B under the name EXERCISE.DOC. If this is a new filename, XyWrite goes ahead and creates a new file with that name. However, if this filename already exists, XyWrite will ask if you want to write over the existing file.

ACTION *(Option 3)*

Saving a Block of Text.
You can save a block of text you have defined:

1. Define the block of text you want saved on the disk.
2. Decide on a name for the file, say, ITEM.
3. Type: F5 savedef b:item.doc ↵

Result: The block of text is now copied to disk in the file ITEM.DOC. The original block remains highlighted in the document. This new file can be called to the display whenever you desire for editing.

TIP #1

Save Frequently. As a precaution, it's a good idea to SAVE your document frequently, say every ten minutes. Then if a power failure were to occur, you would lose only the changes you made since you last saved.

TIP #2

Quick Save. To simplify frequent saves, type the word SAVE on the command line (as shown above), and leave it there while you write or edit. Then, whenever you want to save the document, simply press F9 to execute the SAVE. Another fast way to save is to assign the SAVE command to a Save/Get key. See Saving a File, under "User Programming."

NOTE #2 **Windows.** If there is more than one document open when SAVE is executed, the document where the cursor is located is the one which is saved. (See the section on Windows.)

NOTE #3 **DOS Devices.** You can SAVE directly to DOS Devices — for example SAVE LPT1. The text is sent directly, without the benefit of the Printer File or Character Substitution File. For more information, see TYPEF.

NOTE #4 **SAVE %.** You can save the contents of a Save/Get key to a file on disk with SAVE %. For example, SAVE %A saves the contents of Save/Get A to a file it names A.SAV.

FORMAT

CM**SAVE** *(Option 1)*

CM**SAVE** *d:filename* *(Option 2)*

CM**SAVEDEF** *d:filename* *(Option 3)*

- *filename* is the name under which you want to save the displayed document. (If the filename is omitted, the document is saved under its own name.)
- *d:* (optional) is the drive letter (A:, B:, or C:) you specify for the drive you want. If you omit the drive letter, the default drive is used.
- SAVE is an immediate command (it is not embedded).

EXAMPLE

CMsave b:chapter.doc *(Option 2)*

CMsavedef b:item.doc *(Option 3)*

ABBREV

CMsa b:chapter.doc *(Option 2)*

CMsad b:item.doc *(Option 3)*

PURPOSE

STORE saves the displayed document to a disk drive and clears the display. You are then free to call another document to the display, or quit XyWrite. You have two ways to store a file, the first being the more frequently used:

- Storing a Document Under Its Own Name
  CM**STORE** *d:* *(Option 1)*
- Storing a Document Under a Different Name
  CM**STORE** *d:filename* *(Option 2)*

ACTION *(Option 1)*

Storing a Document Under Its Own Name.

To store the displayed document to where it came from:

Type: F5 store ↵

Result: The file is stored to the filename shown at the top center of the display.

To store a file to a different directory, for example TEST:

Type: F5 store \test ↵

(Note: If there were no directory named TEST, the file would be stored to the filename TEST)

To store it to another drive, such as drive B:

Type: F5 store b: ↵

ACTION *(Option 2)*

Storing a Document Under a Different Name.

To store the displayed document under another name, include that filename with the command. For example:

Type: F5 store b:\book\chapter ↵

Result: This example stores the displayed document to a file named CHAPTER in the \BOOK directory on drive B. If this is a *new* filename, this command creates a new file with that name. If this filename already exists, XyWrite will ask if you want to write over the contents of that file.

TIP

Windows. If there is more than one document open when STORE is executed, the document where the cursor is located is the one which is stored. (See the section on Windows.)

ALSO SEE

Storing to Two or More Drives at Once. You can tell XyWrite to store to *two different* drives each time you execute STORE (or SAVE). This is great for keeping backup files. Refer to the procedure "Setting the Default Drive and Save-Drives" in section "A:" earlier in this chapter.

FORMAT

CM **STORE** *(Option 1)*

CM **STORE** *d:filename* *(Option 2)*

- *filename* is the name of the file where you want to store the displayed document. If you omit the filename, the document is stored under its own name.
- *d:* (optional) is the drive letter (A:, B:, or C:) you specify for the drive you want. If you omit the drive letter, the default drive is used.
- STORE is an immediate command.

EXAMPLE

CM store b:chapter.doc

ABBREV

CM st b:chapter.doc

PURPOSE

The **TREE** command displays *all* the subdirectories on a given drive, shown as a tree. Filenames are not shown.

CD (Change Directory) appears on the command line whenever a TREE is called up. So after you position the cursor at the desired subdirectory, pressing F9 switches to that subdirectory without leaving the tree. You can also execute the DIR and RMDIR commands this way.

ACTION

Displaying a Tree of Subdirectories on a Disk.
To display the tree of subdirectories for drive C:

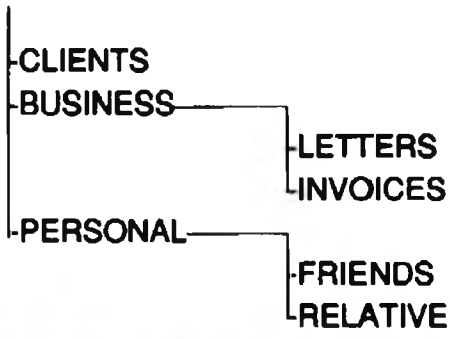
1. Type: F5 `tree c:` ↵

   All subdirectories are displayed. The first line (which has no name) is used to switch to the root directory. Those located immediately off the root directory are displayed along the left margin. (See illustration below.)

2. The CD (Change Directory) command appears on the Command Line whenever a tree is generated. To change subdirectories, move the cursor anywhere on the line containing the subdirectory name. (For names which are spaced away from the left margin, you needn't move the cursor onto the name—just onto the same line.)

3. Press: F9

Result: The subdirectory you selected is now the current directory.



FORMAT CM **TREE** *d:*

- *d:* (optional) is the drive you want listed.
- TREE is an immediate command.

EXAMPLE CM `tree c:`

PURPOSE

TYPE sends text to your printer for printout. The text can originate from a document stored on disk (Option 1) or from the displayed document (Options 2 and 3).

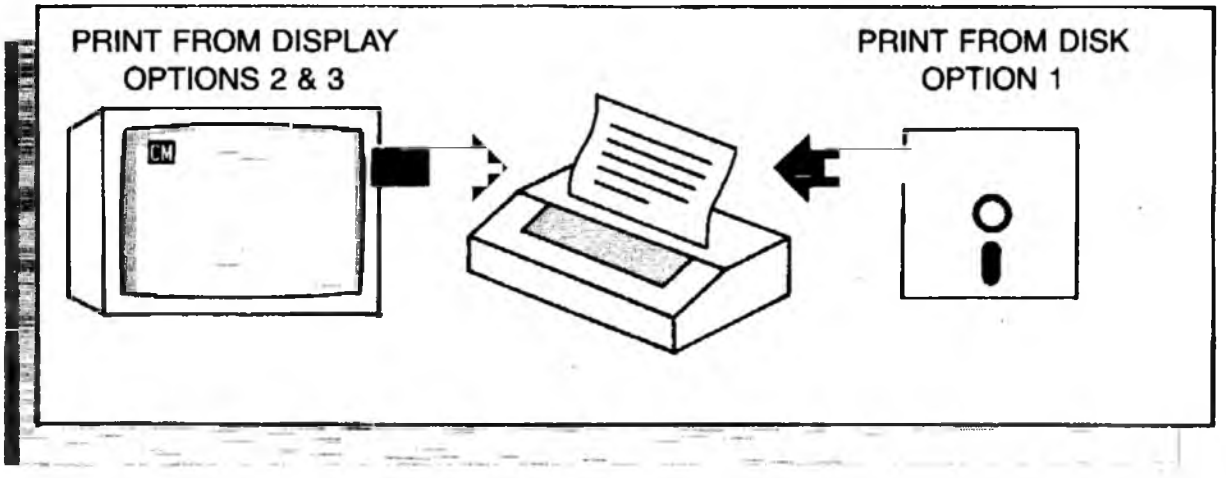
Option 1: To print a stored document, include its file-name. Option 2: In its simplest form, TYPE by itself prints the document currently displayed. In either case you can print any range of pages you wish, print odd and even pages to produce two-sided copies, and also request the printer to pause after each page, for sheet feeding. The general, simplified forms are:

- Printing a Stored Document *(Option 1)*
  CM **TYPE** *d:filename*
- Printing the Displayed Document *(Option 2)*
  CM **TYPE**

Option 3: If you define a block of text on the screen before you enter the TYPE command, then only that block is printed. This is a special case of Option 2:

- Printing a Defined Block of Text *(Option 3)*
  CM **TYPE**

You can print several files by executing the TYPE command once for each file. The files will automatically be queued by XyWrite to print.



ACTION *(Option 1)*

Printing a Stored Document.
To print a file directly from a disk:

1. Make sure the printer is turned on and the On-Line (Ready) light is on.
2. Type: F5 type chapter.doc,3-12/23-26,P ↵

Results: This prints pages 3 thru 12 and 23 thru 26 from file CHAPTER.DOC, pausing after each page (press ↵ to resume). Once printing begins, you are free to continue editing while printing continues in the background. You are free to change default drives or change the current directory. (However, do not remove the floppy disk while the document is printing — printing will stop.)

EXAMPLES

Examples of Printing a Stored Document. The document being printed is named CHAPTER.DOC.

CM type chapter.doc	Prints all pages, without pausing.
CM type chapter.doc,3	Prints page 3 only.
CM type chapter.doc,3-	Prints from page 3 to the end of file.
CM type chapter.doc,-12	Prints from the start of file to page 12.
CM type chapter.doc,3-12	Prints pages 3 thru 12 without pausing.
CM type chapter.doc,3-12/18/23-26	Prints pages 3 thru 12, 18, and 23 thru 26.
CM type chapter.doc,3-12,p	Prints pages 3 thru 12, pausing after each page.
CM type chapter.doc,,p	Prints *all* pages, pausing after each page. Notice the two commas when no pages are specified.
CM type chapter.doc,3-12,ep	Prints only *even* pages 3-12. (See Note #14)

ACTION *(Option 2)*

Printing the Displayed Document.
To print the version of the file currently being displayed:

Type: F5 type ,3-12,p ↵

Results: This prints pages 3 through 12 of the document currently being displayed, pausing after each page. Once printing begins, you are free to continue editing while printing continues in the background.

When you issue TYPE, XyWrite first copies the displayed document to a file called PRINT.TMP. The printer prints from this file, freeing you to return to the original file (or any other file) to continue editing and saving the document. (The previous contents of PRINT.TMP is overwritten each time you print from the display.)

EXAMPLES

Examples of Printing the Displayed Document.
These examples are similar to those on the previous page except the filename CHAPTER.DOC is omitted. The commas must be included as shown.

CMtype	Prints all pages from the display (without pausing).
CMtype ,-12	Prints from the start of file to page 12.
CMtype ,3-12,p	Prints pages 3 thru 12 of the displayed file, pausing after each page.
CMtype ,3-12/18/23-26	Prints pages 3 thru 12, 18, and 23 thru 26.
CMtype ,,p	Prints all pages, pausing after each page.
CMtype ,3-12,e	Prints only *even* pages from 3 to 12. (See Note #14)

NOTE #1

Load Printer File. When printing, you should have the correct Printer File loaded. Look in your STARTUP.INT file where you should be able to find the name of your Printer File. See "Startup File" in Chapter 6 for more information.

ACTION *(Option 3)*

Printing a Defined Block of Text.
To print a block of text currently being displayed:

1. Use F1 (or any other define key) to define the block of text you want to print. Include any embedded commands (such as TS, IP, LM, RM, PG) you want to affect the printout.
2. Type: F5 type ↵

Results: This prints the defined block of text. However, beware that all embedded commands *outside* the defined block are ignored. Therefore you might not get the format you expected (Option 2 does not ignore embedded commands, and so is often preferable to Option 3.)

NOTE #2 **Set Page Number.** When you use the SP (Set Page No.) command to change page numbers, those new numbers are used by TYPE and displayed in the Page-Line indicator (P-L). If you prefer to have TYPE refer to the *sequential* page number (e.g., the tenth page out of the printer, regardless of printed page number) then use the DEFAULT command to change the SQ setting to 1 (the default is 0). You can also enter SQ=1 in the Printer File (see Printer File in Chapter 6).

NOTE #3 **Kill Printer.** To stop the current file being printed, type KILTYP (or KT) on the command line, for Kill Type. The printer will not stop printing until the text stored in its internal buffer empties. Use KT Q to kill all documents in the queue (see the next note).

NOTE #4 **Printing Multiple Copies.** To print several copies of the same file, execute the TYPE command once for each copy. (Once you have TYPE on the Command Line, press F9 once for each copy you want.) To print several *different* files, again execute the TYPE command with the filename once for each printout. The files are automatically queued to the printer by XyWrite.

NOTE #5 **Chain Printing.** To print several related files, use the TYPE @ command. You can specify the files be treated either separately or "as one" for page numbering, Table of Contents, and Index. See the TYPE @ command later in this chapter.

NOTE #6 **Windows.** In Options 2 and 3, if several documents are displayed, the *current* document is printed — that is, the file in which the cursor is located.

NOTE #7 **Print Screen.** You can print all 25 lines of the screen (command line and all) by typing F5 func pr ↵. (This is is equivalent to the DOS print screen feature.)

NOTE #8 **Formatting.** When you issue the TYPE command, XyWrite automatically formats the file before printing it. *Format* means to carry out the embedded commands (represented by triangles in the text)—that is, inserting any running header, running footer, footnotes, margin offset, page numbers, and widow/orphan page break decisions. TYPE also prepares the file for the printer, by inserting printer codes defined in the Printer File for bold, underline, reverse, superscript, subscript, and whatever else is specified (microjustification).

NOTE #9 **Printer Fonts.** You can print with different styles (pica, elite, proportional) by inserting PT1, PT2 or PT3 embedded command in your document. See "Printer Types" in Chapter 4.

NOTE #10 **Character Substitution.** You can remap the set of characters sent to the printer, if you wish. Refer to the Substitution tables in the "Printer File" section of Chapter 6.

NOTE #11 **Related Commands.** TYPE has a family of commands. Each is fully described elsewhere in this manual.

TYPE formats a file and outputs it to a printer.
TYPES formats a file and outputs it to the display.
TYPEF formats a file and outputs it to a disk file.
TYPE @ is for chain printing.
TYPE + is used for Mail Merge.
TYPE % is used for printing Save/Gets.

NOTE #12 **Pause and Prompt While Printing.** You can insert the PR (Prompt) and PA (Pause) commands anywhere in the text. These enable you to stop printing at that point and display a message such as "Change to Bold Printwheel." Refer to "Printer Pause" in Chapter 4.

NOTE #13 **Double-Spacing.** It's very easy to print your document double or triple-spaced. Use the LS (Line Spacing) command described in the formatting section.

NOTE #14 **Double-Sided Printing.** If you want to print on both sides of the page, use the "o" and "e" options:

1. First print the odd-side of the pages:

   F5 ty report,,o ↵

2. Remove the pages and re-orient them as required for printing on the other side.
3. Print the even pages: F5 ty report,,e ↵

NOTE #15 **Optional Commas.** You can enter TYPE without the commas. Just insert a blank space for each comma shown in the examples.

ALSO SEE **Related Commands:** KILTYP, WAIT, TYPEF, TYPES, TYPE @.

FORMAT

CM **TYPE** *d:filename,a-b,m* *(Option 1)*

CM **TYPE** *,a-b,m* *(Options 2 & 3)*

- *filename* is the name of the file to be printed. If omitted, the displayed file or defined block is printed.
- *d:* (optional) is the drive (A:, B:, or C:) you specify for the drive you want. If you omit the drive letter, the default drive is used.
- *a-b* (optional) is the range of pages to be printed. If omitted, all pages are printed. You can specify up to five ranges with *a-b/c-d/e-f/g-h/i-j*.
- *m* is a modifer:
  - O (print odd pages only)
  - E (print even pages only)
  - P (stop printing after each page)

  P can be used with O or E in any order
- TYPE is an immediate command.

EXAMPLE CM type chapter.doc,3-12,ep *(Option 1)*

ABBREV CM ty chapter.doc,3-12,ep

PURPOSE

TYPEF (Type to File) prints a document to a file. We use the term "*print* to a file" because XyWrite processes the document exactly as it would for the printer, but sends it to a file instead. We call this file the **target file**.

The target file is printer-ready, complete with printer codes. It also fully incorporates any running headers, footers, footnotes, page breaks, page numbers, and other embedded format commands called out in the text.

You will find TYPEF useful for diagnosing and for making other internal checks. Specific uses include:

- Diagnosing the Printer File. Printing a file with TYPEF allows you to see exactly which printer codes XyWrite inserts into the file.

- Checking the Character Substitution File. You can view the text in the target file to see if XyWrite is substituting characters as you expect.

- Viewing On-Screen Justification. If you are doing *whole-space* justification, the target file you produce with TYPEF will show justification on-screen.

- Producing a File with CR/LF at End of Every Line. (TYPES also does this.)

- Stripping a File of All Embedded Commands. This allows you to eliminate all format commands unique to XyWrite — for example «MDBO» and «MDUL». You might find this helpful when sending a file to someone using a different word processor. You would load the printer file STRIP.PRN before using TYPEF. (See the chapter on Customizing.)

- For sending files to other devices, such as a port (COM1 or COM2) or line printer (LPT1 or LPT2). Simply use the device name as the targetfile.

Just as the TYPE command has three options, so does TYPEF:

Option 1 is the most general way to print to disk. You can specify any filename on disk and print to a target file. You can either specify the target file by name, or omit the name, and Xywrite will name it FO.TMP.

- Printing a Stored Document to Disk *(Option 1)*
  CM**TYPEF** *d:filename,d:targetfile*

Option 2 allows you to print the *displayed* file to disk. The general form is the same as Option 1, except you omit the first filename. (You can omit the target file name as well — XyWrite will name it FO.TMP.) Note the presence of a comma.

- Printing the Displayed Document to Disk *(Option 2)*
  CM**TYPEF** *,d:targetfile*

If you define a block of text on the screen before you enter the TYPEF command, then only that block is printed to disk. This is a special case of Option 2:

- Printing a Block of Text to Disk *(Option 3)*
  CM**TYPEF** *,d:targetfile*

With any of these three options you can print just a single page or specific ranges of pages, if you wish. (For simplicity, page numbers are left off of the three options shown above.)

ACTION *(Option 1)*

Printing a Stored Document to Disk.

To print a file to disk:

1. Have in mind the name of the file you want to print to disk. We'll use CHAPTER.DOC.
2. Decide on a name for the target file. We'll use RESULT.DOC. (If you omit this name, XyWrite will use FO.TMP.)
3. Type: F5 typef chapter.doc,result.doc,2-5 ↵

Results: This prints pages 2 through 5 of the file CHAPTER.DOC to a file named RESULT.DOC.

ACTION *(Option 2)*

Printing the Displayed Document to Disk.

The following is the simplest use of the TYPEF command.

1. Display the document that you want printed to disk.
2. Type: F5 typef ↵

Result: The document is printed to the target file named FO.TMP.

EXAMPLES

You also have the option in step 2 of naming the target file and printing only certain pages.

CM typef ,result.doc	Prints the displayed document to RESULT.DOC
CM typef ,result.doc,2-5	Prints only pages 2 thru 5 of the displayed document to the file RESULT.DOC
CM typef ,,2-5	Prints only pages 2 thru 5 of the displayed document to the file FO.TMP.

NOTE #1

Range of Pages. The same rules that apply to the TYPE command apply to TYPEF for printing a specified range of pages. See the examples under the TYPE command.

NOTE #2

What Specifically Does TYPEF Do? The following changes are made to a file when it is processed by TYPEF.

1. All embedded triangles are removed and replaced with hard text and spaces, exactly as with TYPES. These changes are listed under the TYPES command.

2. TYPEF also processes the file through the Printer File. As a result, the target file has printer codes embedded in it, as exemplified by the following statements.

   - All character modes are converted to printer codes. For instance, in the Diablo Printer File «MDSU» is removed and replaced with the escape sequence ▶D.
   - Justification codes are inserted into the target file. (If using whole-space justification, you can view the text justified on-screen by calling the target file to the display.)
   - Line Ending (LE), Paragraph Ending (PE), File Begin (FB), File End (FE) and Page End (PG) printer codes are inserted into the target file.
   - Any DEFAULT settings specified in the Printer File take effect (unless they are explicitly overwritten by settings in the text). Common examples are Form Depth (FD), Top Margin (TP) and Right Margin (RM).

3. TYPEF processes the file through the Character Substitution File. This enables you to modify the way characters are printed. You could for example cause the number zero to be printed as the combination 0, backspace (◘), /, making the result Ø.

You can create other Printer Files to suit your particular needs: (1) For preserving the character modes, or (2) For converting the codes to typesetting commands for the front end of a typesetter.

NOTE #3 **Printing a Target File.** Once you use TYPEF to create a target file on disk, you can print it using TYPE — it will come out the same with any Printer File. The conversion to printer codes has already taken place with the original TYPEF.

NOTE #4 **DOS Device Names.** You can "print" directly to or from DOS devices by using the device name in place of the filename — for example, TYPEF Chapter, LPT 1. Valid devices are: CON (Console keyboard or screen), COM1, COM2, LPT1 and LPT2. Text is sent directly from the file or display without conversion.

FORMAT

CM **TYPEF** *d:filename,d:targetfile,a-b* *(Option 1)*

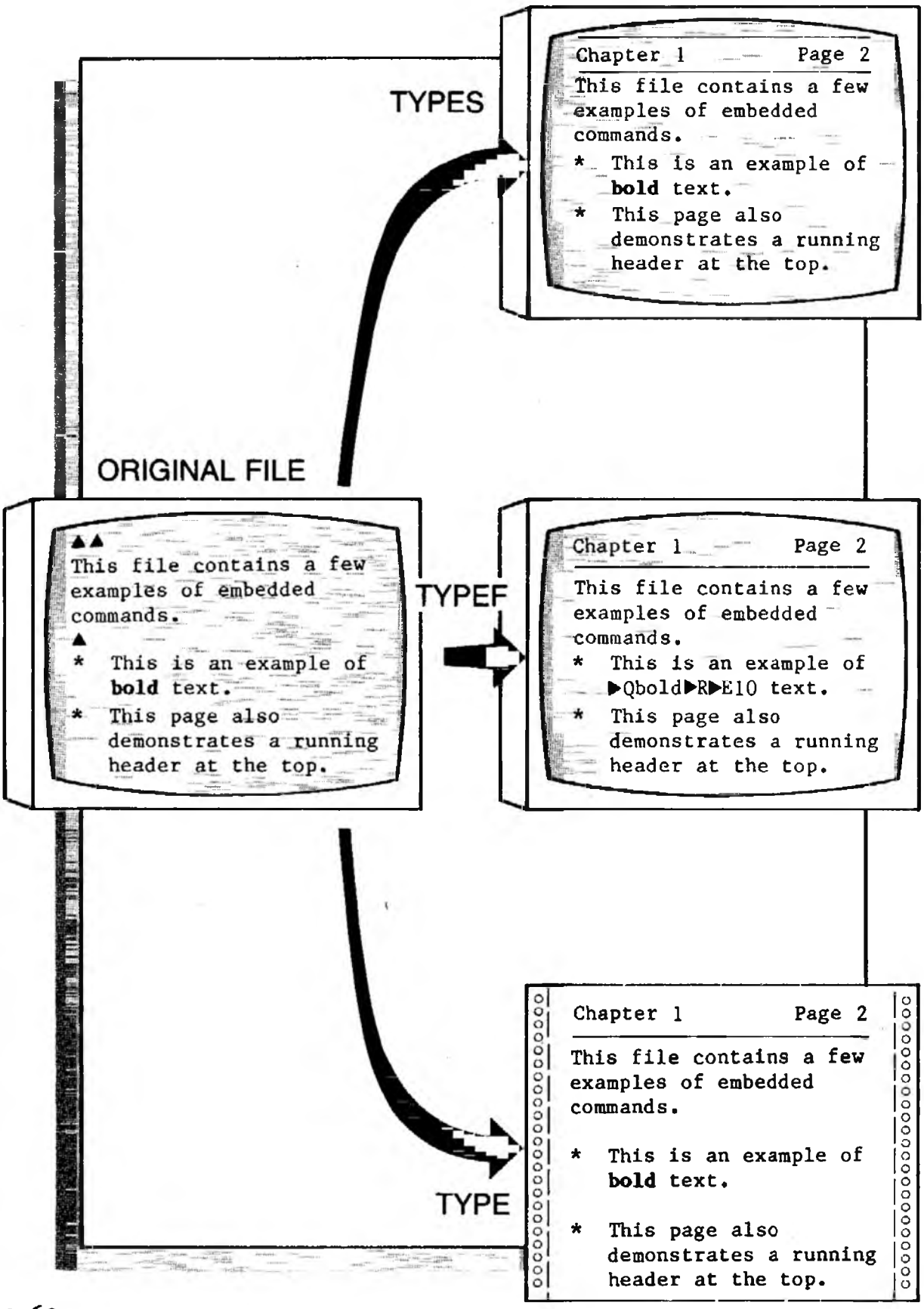
CM **TYPEF** *,d:targetfile,a-b* *(Options 2 & 3)*

- *filename* (optional) is the name of the file to be processed.
- *targetfile* (optional) is the file to which the output is to be sent. If this filename is omitted, the document is printed to FO.TMP.
- *d:* (optional) is the drive (A:, B:, or C:) you specify for the drive you want. If you omit the drive letter, the default drive is used.
- *a-b* (optional) is the range of pages to be printed to file. If omitted, all pages are printed. You can specify up to five ranges: *a-b/c-d/e-f/g-h/i-j*.
- TYPE is an immediate command (it is not embedded).
- This command was formerly called FORMAT, abbreviated FO.

EXAMPLE CM typef chapter.doc,target.doc,3-12 *(Option 1)*

ABBREV CM tyf chapter.doc,target.doc,3-12

Comparison of TYPES, TYPEF and TYPE. The differences between TYPES (Print to Screen), TYPEF (Print to File), and TYPE (Print to Printer) are clarified in the following illustration.



CHAPTER.DOC — Original File

- It contains 3 embedded commands displaying as triangles. In Expanded Display, the triangles expand to:

  «RM70»
  «RHA Chapter 1 Page «PN»»
  «IP0,4»

- Character modes are highlighted:

  BOLD

TYPES CHAPTER.DOC — Preview File

- No embedded triangles — they are incorporated as hard text. For example, the RH command is removed and Running Headers are inserted on every page.
- Character modes are still present. In Expanded Display they would appear as:

  «MDBO»
  «MDNM»

TYPEF CHAPTER.DOC — Printer-Coded File

- This document is formatted very much like the one directly above. In addition to the triangles being incorporated as hard text, the character modes are converted to the codes required by your printer:

  ▶Qbold ▶R ▶E10

- In addition, the character substitution tables in the Printer File take effect, substituting any specified characters.

TYPE CHAPTER.DOC — Printout

- A file equivalent to that shown above under TYPEF is sent to the printer. The printer separates the text from the printer codes — it prints the text and carries out the printer codes as instructions.

PURPOSE

TYPES (Type to Screen) displays a file on the screen as it would be printed. It shows you all running headers, footers, footnotes, page breaks and page numbers. This is a great way to preview a file to make sure it's correctly formatted before printing it. It's much faster than printing, and you can preview just a specific range of pages.

TYPES is often referred to as REVIEW (its former name). The file it produces is called REVIEW.TMP.

Option 1 allows you to preview a file stored on a disk drive. Option 2 lets you preview the file that is currently being displayed. The general, simplified forms are:

- Previewing a Stored Document *(Option 1)*
  CM**TYPES** *d:filename*

- Previewing the Displayed Document *(Option 2)*
  CM**TYPES**

ACTION *(Option 1)*

Previewing a Stored Document.
To preview a file which is stored on a disk:

1. Start with the display cleared of any document. You can open a new window—if you need to clear the display, use STORE or ABORT.

2. Let's say the name of the file you want to print to the screen is CHAPTER.DOC.

   Type: F5 types chapter.doc,2-5 ↵

Results: This prints to the screen pages 2 through 5 of the file CHAPTER.DOC. You can examine this file to see how CHAPTER.DOC would print out on paper.

ACTION
(Option 2)

Previewing the Displayed Document.
To preview a document which is currently displayed:

1. Start with the document in the display.

2. Type: F5 types ↵

Result: The displayed document is printed to another window. (If there is a long delay, refer to the note "Stopping TYPES" below.) You can examine this document to see how it would print out on paper.

TIP

Speeding up TYPES. When printing to the screen, a long document (over 20K) may take over a minute. To speed this up, specify only the range of pages you need, rather than the entire document. If you must print the entire document, perform Option 1 with only one window open, to free up memory. (See "Memory.")

NOTE #1

Stopping TYPES. The longer the document is, the longer it takes TYPES to complete its operation. If you find it taking too long, use Ctrl Break . This stops the process and displays what has been computed up to that point.

NOTE #2

What Specifically Does TYPES Do? The following changes are made to a file when it is processed by TYPES. All embedded commands are carried out by XyWrite as hard text and their triangles are removed. For example, the RH command is removed and Running Headers are inserted at the top of every page. Besides inserting the horizontal line between pages, the most noticeable changes made by TYPES are:

RH Running headers are incorporated into the text
RF Running footers are incorporated into the text
FN Footnotes are incorporated, with numbering
FD Soft page breaks are converted to hard breaks
PN Page numbers are inserted
DA Current date is inserted
TM Current time is inserted
RM Soft returns are converted to hard returns
TS Tabs are converted to series of spaces
TP Top margins are inserted as blank lines
BT Bottom margins are inserted as blank lines
OF Offset is incorporated as a shift in margins
LS Line spacing is incorporated as blank lines
WD Widow paragraph breaks take effect
OP Orphan paragraph breaks take effect
IX Index is extracted
TC Table of contents is extracted
~ Soft hyphens are converted to hard hyphens or are removed

TYPES does not process the file through the Printer File. As a result, the following statements hold true. (In contrast, the TYPEF command does process the file through the Printer File.)

- All character modes remain unchanged. For instance, a word which was originally underlined remains underlined.
- Justification is not apparent on the display. (Determined by value of MU in Printer File.)
- Printer codes are not embedded in the displayed file.

NOTE #3 **Range of Pages.** The same rules that apply to the TYPE command apply to TYPES for printing a specified range of pages. See the examples under the TYPE command.

NOTE #4 **Saving a Preview File.** You can save a file created with TYPES simply by using SAVE. It is saved under the filename REVIEW.TMP unless another name is specified.

NOTE #5 **Page Breaks and Page Numbers.** TYPES inserts horizontal lines across the width of the document to indicate page breaks. If you want to use the Page-Line indicator to correctly count lines and pages, you must insert a PL command at the top of the preview document. For a document with 66 lines per page, insert PL 67. (The extra number is needed in order to include the horizontal page-break line.)

FORMAT **CM TYPES** *d:filename,a-b* *(Option 1)*

CM TYPES *,a-b* *(Option 2)*

- *filename* is the name of the file to be printed to the screen.
- *d:* (optional) is the drive (A:, B:, or C:) you specify for the drive you want. If you omit the drive letter, the default drive is used.
- *a-b* (optional) is the range of pages to be printed. If omitted, all pages are printed. You can specify up to five ranges with *a-b/c-d/e-f/g-h/i-j*.
- TYPES is an immediate command.
- This command was formerly called REVIEW, abbreviated RV.

EXAMPLE **CM types chapter.doc,3-12** *(Option 1)*

ABBREV **CM tys chapter.doc,3-12** *(Option 1)*

PURPOSE

The **TYPE @** command prints a sequence of files to the printer, one after another, automatically. One benefit is that you may group files into **sets**. A set of files is treated as a single document — with a single run of page numbers, footnote numbers, chapter numbers, a single Table of Contents and a single Index accumulated from across all named files.

Similarly, TYPES @ and TYPEF @ print a sequence of files to the display and to a file, respectively.

- Chain Printing to the Printer
  CM **TYPE** *@parentfile,,m*
- Chain Printing to the Display
  CM **TYPES** *@parentfile*
- Chain Printing to a File
  CM **TYPEF** *@parentfile,targetfile*

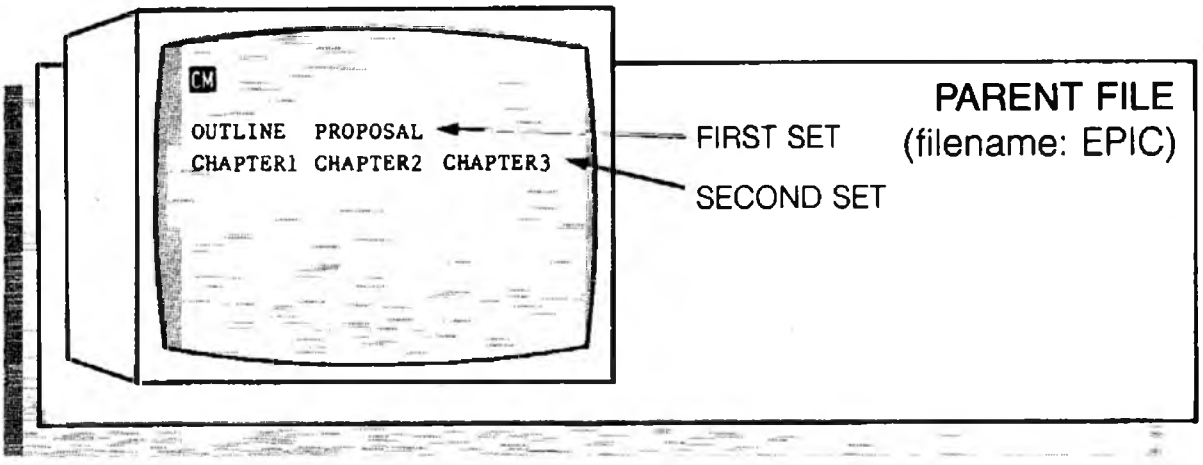
ACTION

Chain Printing to the Printer.
The following example illustrates how to print a sequence of files.

1. **Create the Parent File.** The parent file contains the names of the files to be printed, as the following example shows.

   a. Open a new file to be the parent file, giving it any name you wish (for example, EPIC):

   Type: F5 new epic ↵



b. List the names of all of the files you want printed as a set. Separate the names by spaces. At the end of the set press ↵. (You are not constrained as to the number of files in a set. The files may word wrap to subsequent lines and still be considered a set.)

   Type: outline proposal↵

c. List as many sets as you want. Make sure you end the last set with a carriage return.

   Type: chapter1 chapter2 chapter3↵

d. Store the file:

   Type: F5 store epic↵

2. **Insert Page Breaks.** XyWrite automatically inserts page breaks between sets — that is, the first document of each set will start at the top of a new page. However, if you want a document within a set to also start at the top of a new page, make sure it begins with a Page Break command (PG).

3. **Preview the Files.** (Optional) If you wish, preview the files on the display before you print them: (see the TYPES command for more details)

   Type: F5 types @epic↵

4. **Print the files.** Finally, print the files:

   Type: F5 type @epic↵

   Results: This single command prints all five files as shown in the following illustration. In our example, notice that the two lines in EPIC produce two sets of page numbers, as follows:

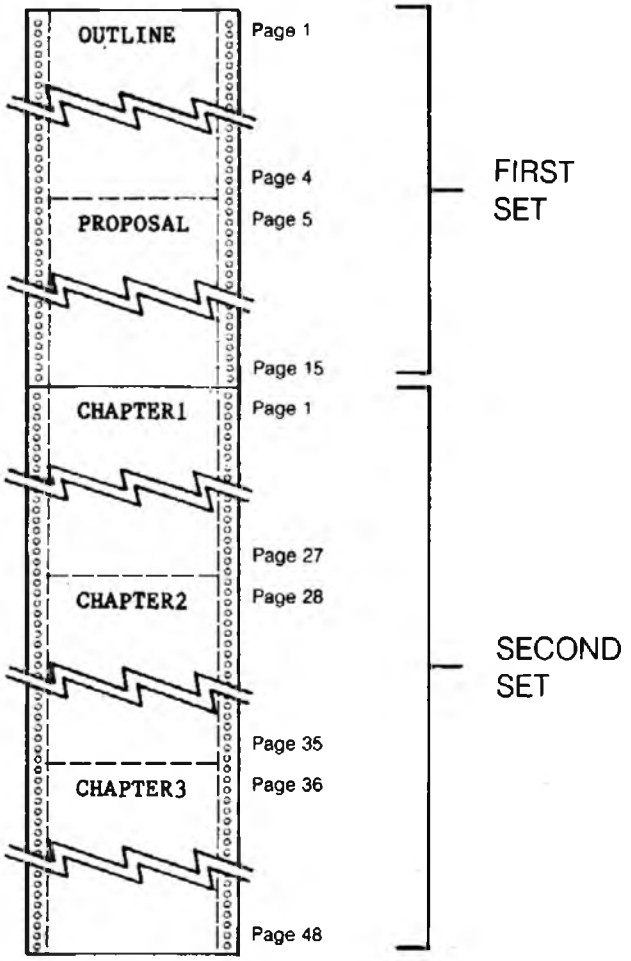
   - OUTLINE and PROPOSAL are printed as one document, with pages numbered from 1 to 15 in this example.

   - CHAPTER1 thru CHAPTER3 are printed as one large document, starting at page 1 (and footnote 1) with one continuous set of page numbers, footnotes, Table of Contents and Index.

NOTE #1 **Missing File.** When printing, if a filename in the parent file is not found on the disk, the file is ignored, and output continues uninterrupted with the next file.

NOTE #2 **Page Format.** The overall page format commands (such as Offset, Top Margin, Running Header) can be placed at the beginning of the first file. Their effects carry over into the following files of that set only, but not to following sets. (The DEFAULT conditions take over again at the start of each new set.)

CHAIN PRINTING



The same is true for numbering systems, including page, chapter and footnote numbering. You can put DC (Define Counter), FN (Footnote) and related commands in the first *file* of a set — the formats and sequences are maintained throughout that set only. This is a very powerful capability for printing a complete book, from several files. See Chapter 4, "Footnotes," "Page Numbering" and "Numbering" for details.

NOTE #3 **Background Printing.** Once printing begins, you may continue editing. But wait until a particular file is completely printed before saving to it.

NOTE #4 **Kill Printing.** One Kill Type (KT) command is all that's necessary to stop all the files from printing.

NOTE #5 **Related Commands.** The Set Page Number (SP) can be used to reset the page numbers anywhere in the text.

NOTE #6 **Chapter Numbers**. When you use TYPE @, you can automatically number your chapters by using the C0 counter. See "Numbering" in Chapter 4.

FORMAT

CM **TYPE** @*parentfile*,,*m* — Prints to the printer

CM **TYPES** @*parentfile* — Prints to the screen

CM **TYPEF** @*parentfile*,*targetfile* — Prints to a file

- *parentfile* is the parent file, containing just the names of the files to be processed consecutively.
- *targetfile* (optional) is the file to which the sequence of files is to be sent. If this filename is omitted, the documents are printed to FO.TMP.
- *m* is a modifier:
  - O (print odd pages only)
  - E (print even pages only)
  - P (stop printing after each page)

  P can be used with O or E in any combination
- Be sure to include the two commas.
- These are immediate commands.

EXAMPLE

```
CM type @epic,,ep
```

ABBREV

```
CM ty @epic,,ep
```

PURPOSE

The **WAIT** command causes printing to occur in the foreground instead of the background. Its effect is to make XyWrite wait until printing jobs are finished before the next keystroke is executed. You might use WAIT in a user program where you want to use the result of a TYPEF (Type to File) for further action. For example, if you were to use TYPEF REPORT, RESULT and wanted to next call up RESULT, it would be prudent to include a WAIT:

```
BC typef report,result
BC wait
BC call result
```

If you did not include WAIT, TYPEF would occur in the background and the CALL statement would attempt to call up RESULT before TYPEF was finished.

When you execute the WAIT command, the display freezes. XyWrite does not act on any further keyboard or keystroke input until the TYPE command is completed. Any keystrokes are saved and will be acted on when the current TYPE or TYPEF job is finished.

ACTION

Type to a File and Call the Result.

To demonstrate the WAIT command, we will type to a file with TYPEF and then call that file to the screen:

1. Type: F5 typef long,fo.tmp ↵
2. While the file LONG is being processed:

   Type: F5 wait ↵
3. Now immediately type:

   Type: F5 ca fo.tmp ↵

Result: Notice that your keystrokes in Step 3 are not immediately displayed — they are, however accepted by XyWrite. WAIT prevents the display from changing until TYPEF is finished. Then FO.TMP is called to the screen.

FORMAT

CM **WAIT**

- WAIT is an immediate command.

EXAMPLE

CM wait

PURPOSE

QUIT terminates the current editing session by clearing any open files from the display and returning control to DOS. Its general form is:

- CM **QUIT** *(Option 1)*
- Ctrl Alt Del *(Option 2)*

ACTION *(Option 1)*

Quitting XyWrite.
To quit XyWrite:

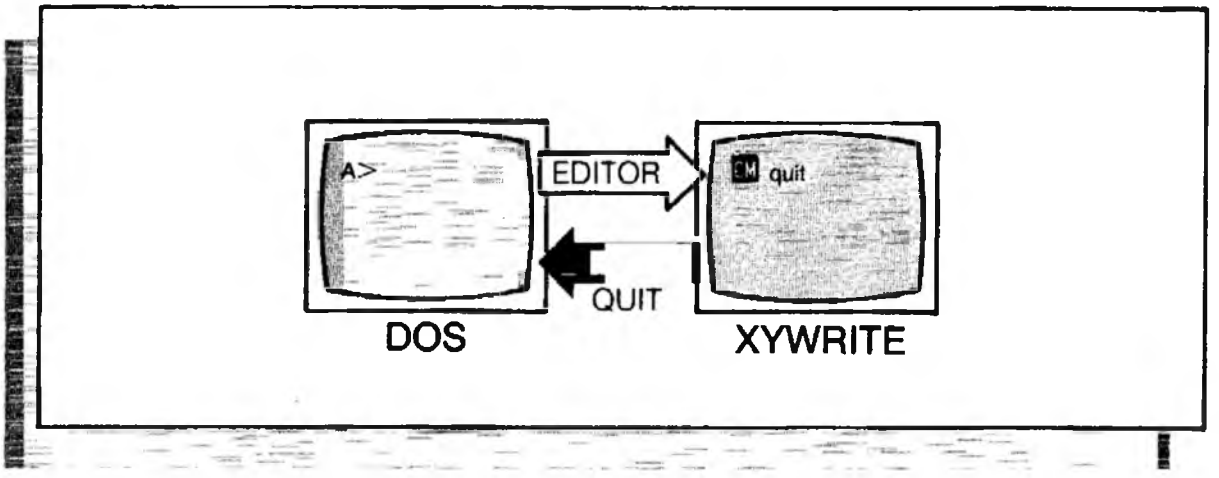
1. Clear documents from the display using STORE or ABORT. (Be sure to also clear any file from the second window, if present.)

2. Type: F5 quit ↵

Result: If any files are open, XyWrite warns you and asks if you want to quit anyway. (You answer Y to quit or N to continue in XyWrite.) XyWrite then returns control to DOS. All memory used by XyWrite is freed. All temporary (.TMP) files are automatically deleted except INDEX.TMP and PRINT.TMP.

If you wish to re-enter XyWrite after quitting, you must enter the following at the DOS prompt (A>, B> or C>):

Type: editor ↵



ACTION *(Option 2)*

Quitting XyWrite.
If your system becomes locked up for some reason and Option 1 doesn't work, use the following method:

Press: Ctrl Alt Del

XyWrite first asks if you want to quit. If you do:

Press: y

If you have files open, XyWrite asks if you want these files saved. If you do, press Y again. XyWrite then automatically saves all open files to your disk. In this way, XyWrite allows you to save files even if your system is locked up. However, it does not save these files using their original names. It saves the file in window #1 as QUIT1.TMP, the file in window #2 as QUIT2.TMP, and so on up to QUIT9.TMP.

ALSO SEE

The DOS Command. The DOS command also switches control to DOS, but without disturbing XyWrite. XyWrite is suspended as-is; you can return to any files left open with the EXIT command. In contrast, QUIT requires open files to be stored or aborted. (The DOS command can be found earlier in this section.)

FORMAT CM**QUIT**

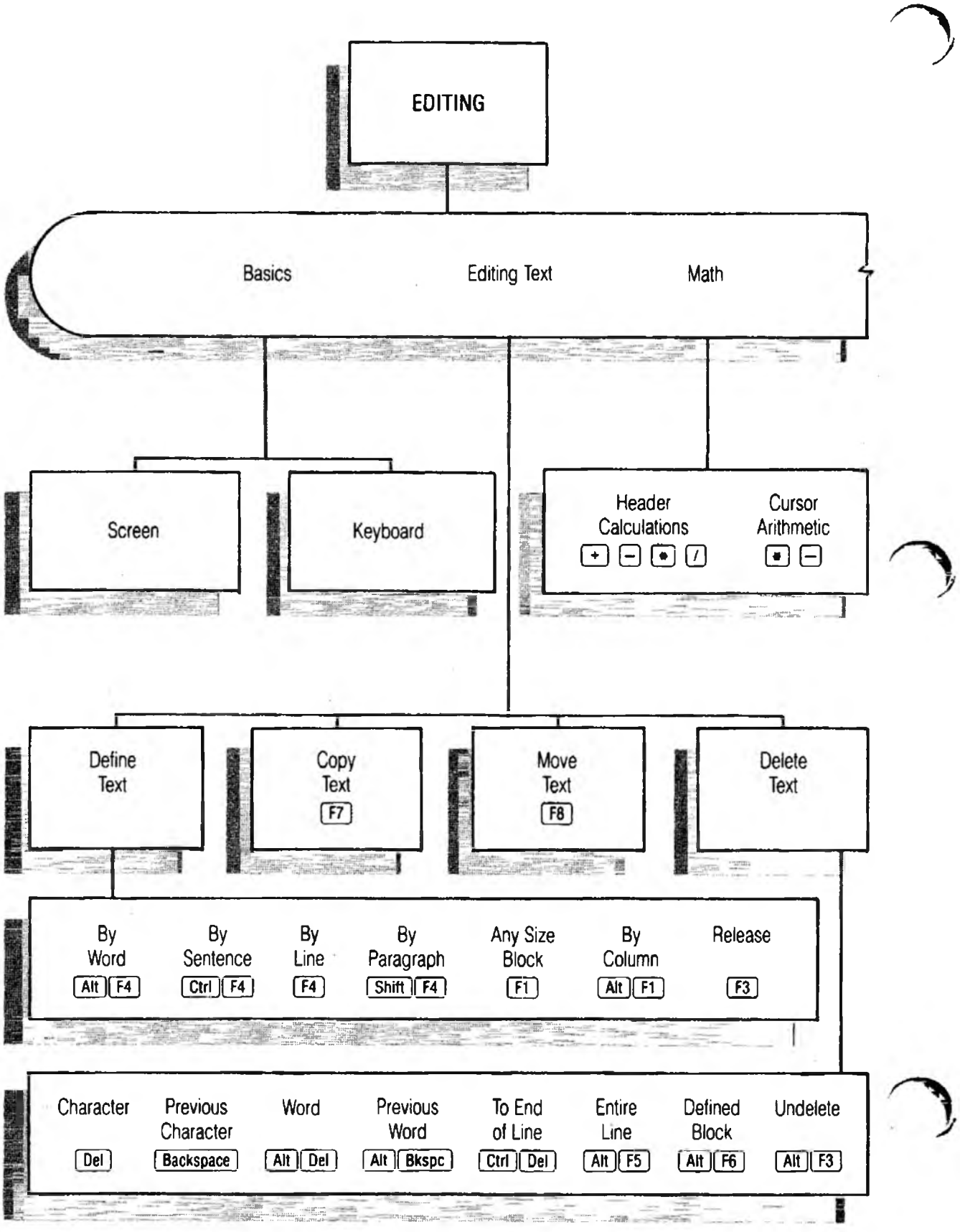
- QUIT is an immediate command.

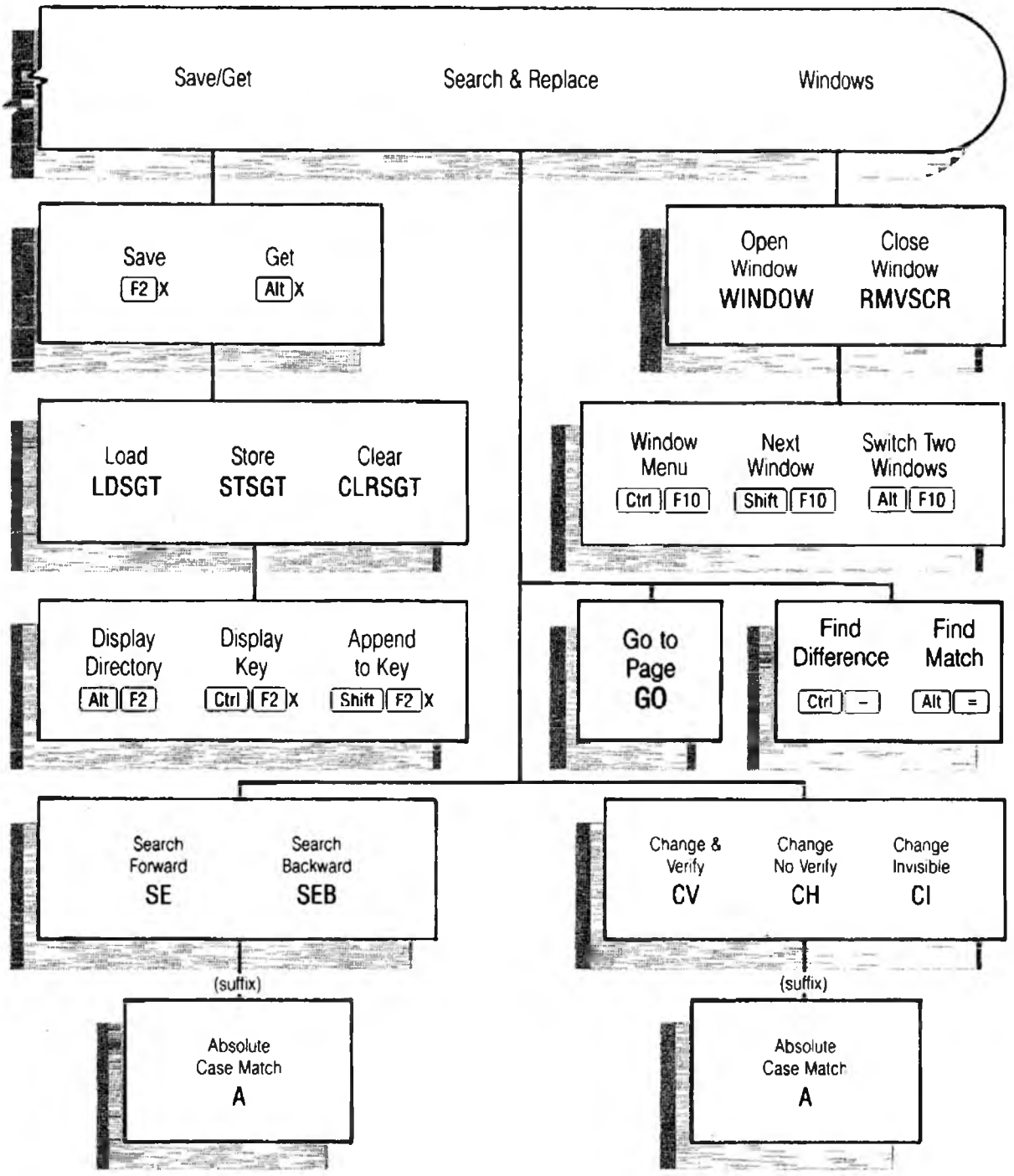
EXAMPLE CMquit

INTRO

This chapter covers basic editing. It begins with the screen and keyboard and progresses into methods you can use to modify text. These tools are the same whether you are writing a new document or modifying an existing one.

CONTENTS


EDITING
Basics
Editing Text
Math
Screen
Keyboard
Header Calculations
+ − * /
Cursor Arithmetic
* −
Define Text
Copy Text
F7
Move Text
F8
Delete Text
By Word
Alt F4
By Sentence
Ctrl F4
By Line
F4
By Paragraph
Shift F4
Any Size Block
F1
By Column
Alt F1
Release
F3
Character
Del
Previous Character
Backspace
Word
Alt Del
Previous Word
Alt Bkspc
To End of Line
Ctrl Del
Entire Line
Alt F5
Defined Block
Alt F6
Undelete
Alt F3


Save/Get
Search & Replace
Windows
Save
F2 X
Get
Alt X
Open Window
WINDOW
Close Window
RMVSCR
Load
LDSGT
Store
STSGT
Clear
CLRSGT
Window Menu
Ctrl F10
Next Window
Shift F10
Switch Two Windows
Alt F10
Display Directory
Alt F2
Display Key
Ctrl F2 X
Append to Key
Shift F2 X
Go to Page
GO
Find Difference
Ctrl -
Find Match
Alt =
Search Forward
SE
Search Backward
SEB
(suffix)
Absolute Case Match
A
Change & Verify
CV
Change No Verify
CH
Change Invisible
CI
(suffix)
Absolute Case Match
A

NOTES

INTRO

The keyboard and screen are useful topics for beginning a description of XyWrite. The Screen section is fairly elementary — you can cover it in one reading. You may find yourself referring to the Keyboard section several times until you've become familiar with it.

CONTENTS

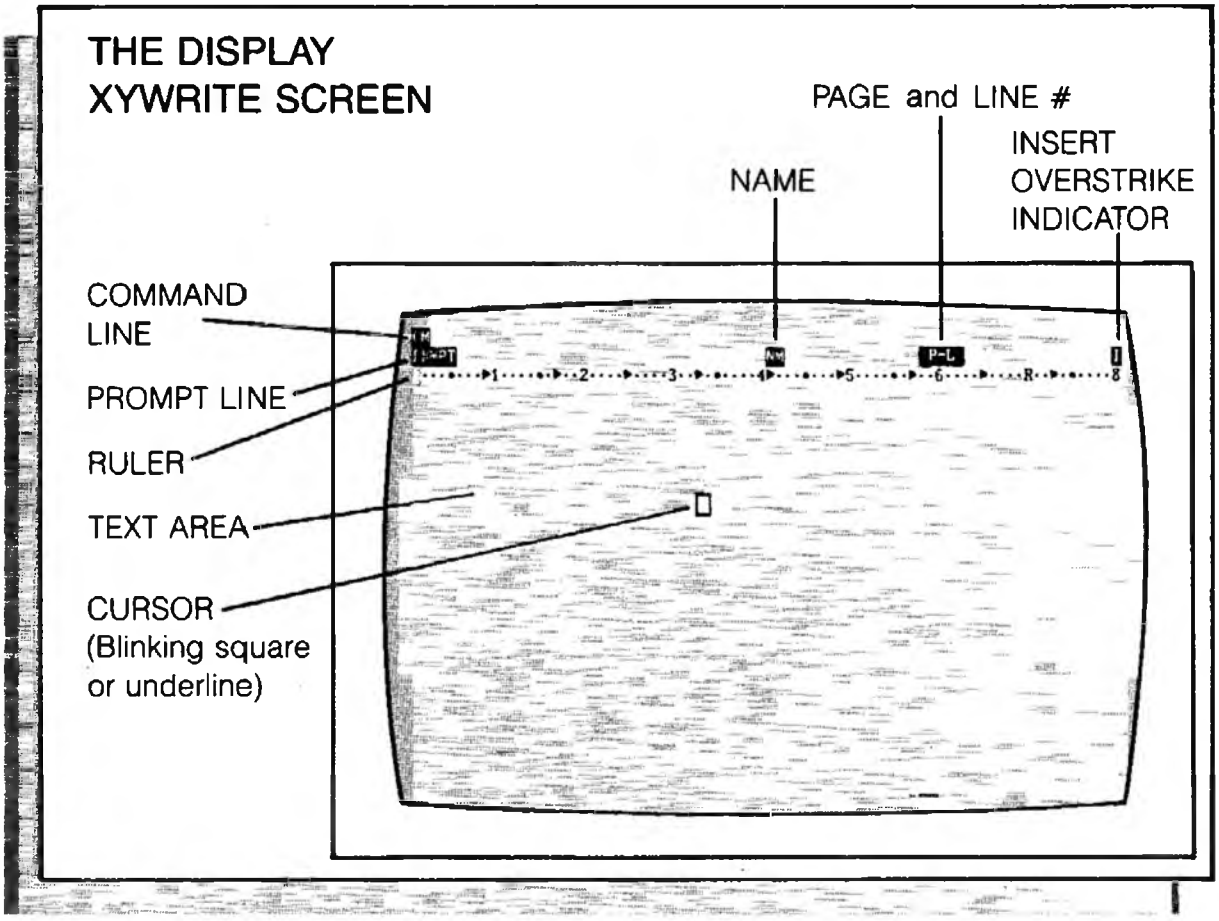
PURPOSE

The display is divided into two separate areas: the header, which is the top three lines, and the Text Area, which is the rest of the display. The features of XyWrite that are visible on the screen are described here.

The Cursor. The *cursor* is the blinking square or underline on the screen. The position of the cursor marks the "point of action" where text or commands are entered or deleted.

The shape of the cursor is a *square* in Insert Mode and an *underline* in Overstrike Mode. (You switch modes with the Ins key.)

Command Line. The Command Line is the entire top line of the screen. The CM (Command) indicates where you enter commands such as NEW, CALL, SEARCH, TYPE and QUIT. When you see the familiar mark CM you know you're in XyWrite. The CM is where you "talk" to XyWrite.



You can use the CM command to show the default drive in place of CM, such as C. (See Chapter 2 for information on the CM command.)

Many commands you type require as few as two letters, but some commands might use the entire line — for instance, PR (Prompt) with a message.

Prompt Line. This is the second line on the display — it has four parts:

- PRMPT (Prompt) - This is where XyWrite displays messages for you.

- NM (Name) - The filename of the currently open file appears here in whatever character mode is currently in effect. When you use SAVE or STORE (with no filename), XyWrite saves the document under this name. When you have several files open, the NM field tells you which document is currently active.

  You can enter the CM command to change NM to show the window number and define indicator, such as C= and change PRMPT to instead show the name of the current directory.

- P-L (Page-Line) - The page and line number of the cursor position appears here. There are three ways to turn on the Page-Line number (the second and third options also move the text):

  Press: Shift F9 *(Option 1)*

  Press: Alt PgDn or Alt PgUp *(Option 2)*

  Press: F5 go 1 ↵ *(Option 3)*

  To turn off the Page-Line number, switch to Expanded Display and back. Press the following key combination twice:

  Press: Ctrl F9

  If you want Page-Line to be displayed automatically when any file is called, change the Display Type (DT) setting in the Printer File. See "Normal/Expanded Display Type" in the Printer File section of Chapter 6.

- Toggle key indicators (described under Toggle Keys later in this section) include:

  C (Caps Lock)
  S (Scroll Lock)
  N (Numeric Lock)
  A (Automatic Uppercase)
  I (Insert/Overstrike)

Ruler. This is the third line on the display. The ruler has markings for margin, tab and indent settings. The Ruler is described in detail in the Formatting chapter.

The Text Area. This is the area where you view and revise documents. This viewing area is 22 lines long and 80 characters wide. You can scroll a document up and down in this area. You can also move it left and right — a document can be as wide as 256 characters.

You can also *split* this window for working on two or more files at the same time. You can view them side-by-side, one above the other, or on alternate full screens.

As you write and format your document, you might notice formatting markers — arrows and triangles — in the text area. If they're distracting, you can hide them with the NM (No Marker) function call. See Function Calls in the Keyboard File section of Chapter 6.

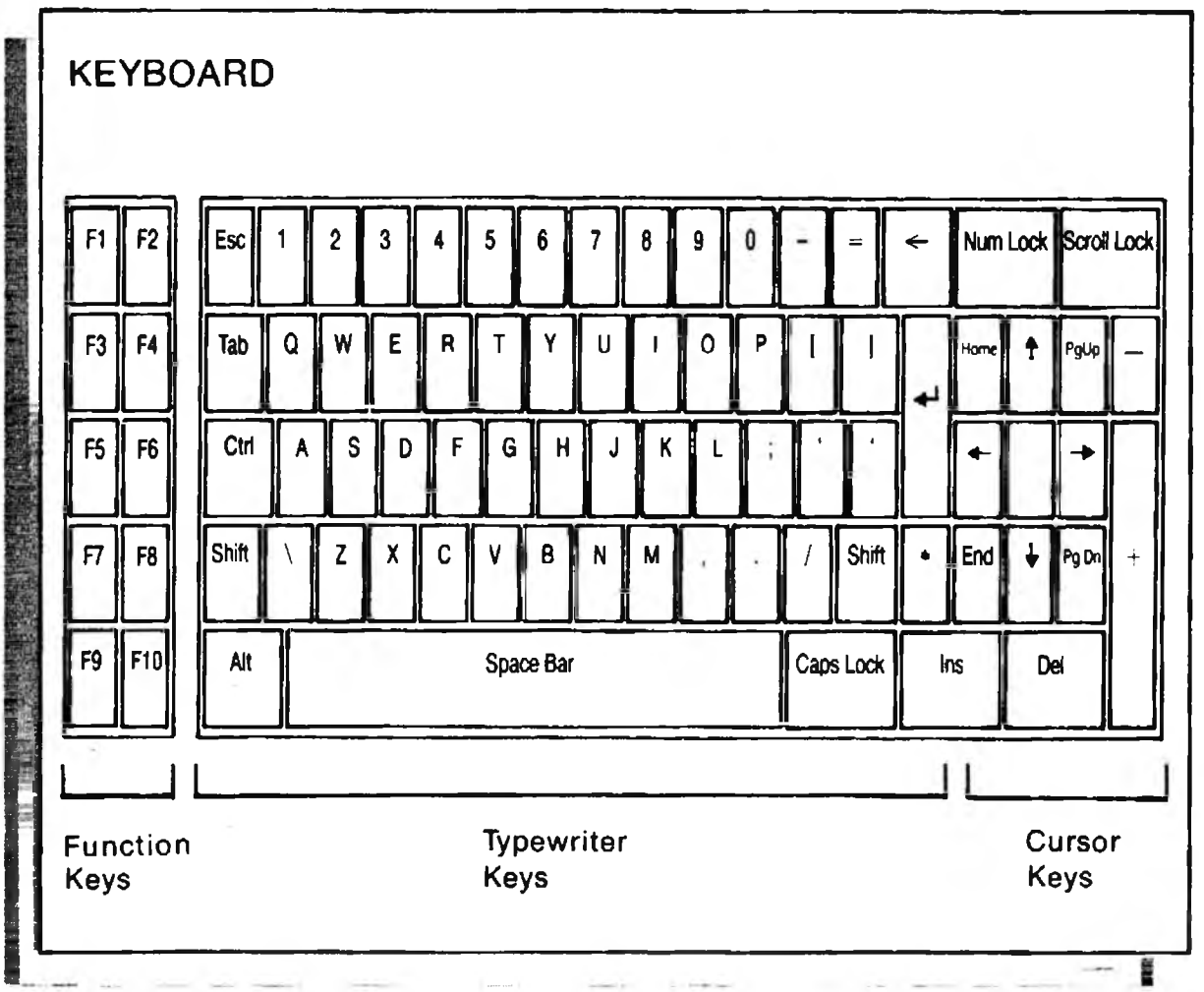
NOTE #1 **Header Field Names**. You can change the header field names to get information about where you are currently working in XyWrite. Refer to the CM command in Chapter 2 for information.

NOTE #2 **Non-Blinking Cursor.** If you prefer a non-blinking cursor, enter the CR command as a DEFAULT. The form of the command is DEFAULT CR=*e,h* where *e* is the character mode for the character under the cursor when editing and *h* is the character mode for the character under the cursor in help frames. For example, enter the command **default cr=112,7**. Use numbers between 1-127 for non-blinking and between 128-255 for blinking. To return to the standard blinking cursor, enter the command **default cr=0,0**.

PURPOSE

The keyboard, of course, is where you type text into your computer. The keyboard has basically three sections to it, as shown below. We will describe on the following pages those parts of the keyboard that are unique to XyWrite.

Typewriter Keys. The center section of the keyboard contains the typewriter keys — letters, numbers, standard punctuation and symbols. You use the keys Ctrl and Alt in combination with character keys to access other functions (Save/Gets and ASCII characters).



Command Line Keys. The Command Line is an essential part of XyWrite. It's important that you learn to use and become comfortable with the three keys associated with the Command Line: F8, F9 and F10.

F5 **Clear The Command Line.** Pressing this key clears the Command Line and positions the cursor next to the CM for you to enter a new command. You typically use F5 *prior* to typing in any command (such as NEW, CALL, SAVE, PRINT or SEARCH).

F9 **Execute.** This is the Execute key. Its sole purpose is to execute whatever command is currently on the Command Line. It works the same whether the cursor is on the Command Line *or* in the Text Area. You typically use F9 *after* typing in any command. For example:

F5 save F9

When you type a command on the Command Line, ↵ is an alternate way to execute the command — the following two lines are equivalent:

F5 save F9
F5 save ↵

The difference between F9 and ↵ is as follows:

- F9 executes whatever function is currently on the Command Line, whether the cursor is on the Command Line *or* in the Text Area.
- ↵ executes a command *only* when the cursor is on the Command Line. (It enters a *carriage return* when in the Text Area.)

There are times when you would want to execute a command from the text area. In these cases, ↵ would not work — for instance, when repeating the SEARCH command. To find "hello" in your text, for example, you'd type SEARCH /hello/ on the Command Line and execute it the first time with either ↵ or F9. Once the first instance is found in text, striking F9 continues the search right from where the cursor stands in the text. (↵ would enter a carriage return.)

F10 **Moving the Cursor Between Command Line and Text Area.** In effect, F10 is a cursor key — pressing this key moves the cursor between the Command Line and the Text Area (when a document is open). Unlike F5, it does *not* clear the Command Line.

There are two main uses for F10:

- To move the cursor from the Text Area to the Command Line in order to modify an existing command. For example, adding the letter "b" to the search command se /tomorrow/ to make it a backwards search:

  CM seb /tomorrow/

  This is quicker than pressing F5 and re-typing the entire command.
- To simply move the cursor from the Command Line to the Text Area. No commands are executed. Remember that F10 is just a cursor key — no more, no less.

The cursor maintains one location in the text and another on the Command Line. F10 switches between these points.

How are F5 and F10 different? F5 clears the Command Line and moves the cursor only one way — to the Command Line. F10 moves the cursor either way without clearing the Command Line.

PURPOSE

Cursor keys allow you to move the cursor or scroll the text. We have listed all cursor keys in the following tables. Note that with all of these functions, holding down the key will repeat the function.

Scrolling Text. These keys allow you to move up or down in a document, moving new text into view. Notice that as you move the cursor *down* in the document, new lines of text appear at the bottom of the screen and the text moves *up* the screen.

NOTE #1

Move by Character. There are two different ways to move the cursor by character:

Character Up, Down, Left, Right. The cursor moves to the next typed character or space, but will not move into the area to the right of a carriage return on any line. When the cursor hits the end of a line, it wraps to the next line.

Linear Up, Down, Left, Right. The cursor moves in a straight line (thus the word linear), regardless of the text or carriage returns present. Linear Left and Right stop at the ends of the lines; they do not wrap to the next line. Thus, linear cursor movements are well-suited for use with a mouse.

SCROLLING UP OR DOWN

Keys	Function	Scroll
Ctrl ↑	One line up	Line
Ctrl ↓	One line down	
PgUp	One screen up	Screen
PgDn	One screen down	
Alt PgUp	One printed page up	Printed Page
Alt PgDn	One printed page down	
Ctrl Home	To top of document	Document
Ctrl End	To bottom of document	

MOVING THE CURSOR

Keys	Function	Move
→	Cursor right	Character
←	Cursor left	*(Notes #1, 2)*
↑	Cursor up	
↓	Cursor down	
*	Cursor linear right	
*	Cursor linear left	
*	Cursor linear up	
*	Cursor linear down	
Alt →	To next word	Word
Alt ←	To previous word	
Ctrl Tab	To next tab	Tab
Shift Tab	To previous tab	
Ctrl →	To end of line (express right)	Line
Ctrl ←	To start of line (express left)	
*	To start of previous line	
*	To start of next line	
*	To start of previous sentence	Sentence
*	To start of next sentence	
*	To start of previous paragraph	Paragraph
*	To start of next paragraph	
Home	To top of screen (Home)	Screen
End	To bottom of screen	*(Note #3)*
Shift F10	Switch through all windows	Window
Alt F10	Switch between two windows	*(Note #4)*
F10	Switch between Command Line and text area	Command Line
F5	Move to and erase Command Line	

* These functions are not pre-assigned to any keys. Refer to the section in Chapter 6 on Keyboard File for instructions on how to assign these options to the keys of your choice.

NOTE #2 **Move by Line.** It is interesting to notice the difference between the following two items:

Character Up, Character Down. These move the cursor up or down a line without shifting the text currently visible on the screen.

Move by Line. This moves the text and cursor up or down a line on the screen, moving a new line into view. The cursor stays on the same character in the text.

NOTE #3 **Move by Screen.** The screen is another word for the display. When you move text by screen, you move the next screenful of text into view, without skipping over any text. In fact, there is one line of overlap.

For example, when moving down in a document, the bottom line moves to the top of the screen. When you move by screen, you can scroll quickly through a document, scanning all text.

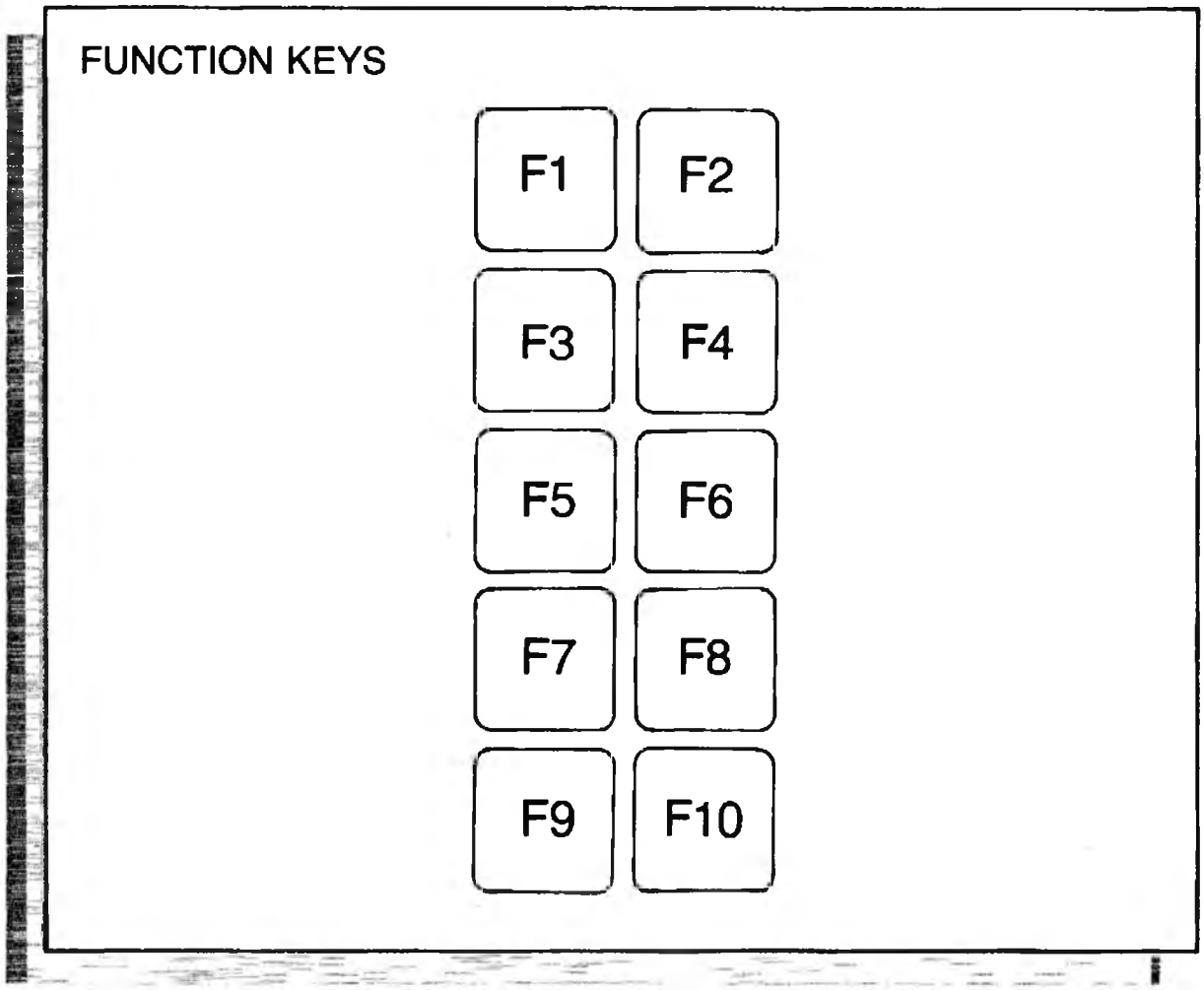
NOTE #4 **Move to Window.** You can have up to nine different windows open at the same time. With Crtl F10 (window menu) you can move to any specific window by pressing the number of that window. With Shift F10 you can move to the next active window in sequence. With Alt F10 you can return to the previously displayed window—use this to toggle between two windows. See the section on Windows for more information.

NOTE #5 **Move by Printed Page.** When you want to see where page breaks will occur, before you print a document, move the text by Printed Page (rather than by screen). The Page-Line indicator on the right end of line 2 turns on, and the start of the next page is displayed. The page breaks adjust for running headers, running footers, footnotes, top margin and bottom margin.

NOTE #6 **Moving Within Column Tables.** You can set up columns of text with word-wrap within the columns. For information on moving the cursor within the text columns, see the section on Columns in Chapter 5.

PURPOSE

The function keys are the ten keys F1 through F10 on the left side of the keyboard. (Refer to the illustration.) These keys provide frequently-used editing functions and commands. Once you know these functions you have a powerful set of tools for editing your text.



FUNCTION KEY SUMMARY

F1	**Begin/End Block Define.** Begin or end defining a block of text of any size.
Alt F1	**Begin Column Define.** Begin defining a *column* of text.
F2	**Save Defined Text.** Save the defined block of text to the Save/Get given by the next key struck.
Ctrl F2	**Save/Get Directory.** Show the contents of the single Save/Get given by the next letter or number struck.
Shift F2	**Append to Save/Get Key.** Add the defined text to the end of the text in the Save/Get given by the next key.
Alt F2	**Show Save/Get Directory.** Display the entire directory of 36 Save/Get keys.
F3	**Release Defined Text.** Release the block of text currently defined.
Alt F3	**Undelete.** Restores the last text deleted (except text deleted using Backspace or Del).
F4	**Define by Line.** Define the line the cursor is on.
Ctrl F4	**Define by Sentence.** Define the sentence the cursor is on.
Shift F4	**Define by Paragraph.** Define the paragraph the cursor is on.
Alt F4	**Define by Word.** Define the word the cursor is on.
F5	**Clear Command Line.** Clear the Command Line and move the cursor there.
Alt F5	**Delete Line.** Erases the line the cursor is on.

FUNCTION KEY SUMMARY (Continued)

F6	**Clear Command Line.** Same as F5.
Alt F6	**Delete Defined Block.** Erase the block of text currently defined.
F7	**Copy Defined Block.** Copy the currently defined block of text to the cursor position.
F8	**Move Defined Block.** Move the currently defined block of text to the cursor position.
F9	**Execute.** Execute the command currently on the Command Line.
Ctrl F9	**Toggle Between Expanded and Normal Display.** Switches the display between Normal and Expanded modes.
Shift F9	**Turn on Page Numbers.** Turns on the P-L indicator at the top of the screen.
Alt F9	**Help Screen.** Displays Help information.
F10	**Toggle Between Command Line and Text.** Move the cursor between the Command Line and text without clearing the Command Line.
Ctrl F10	**Select Window.** Display the menu for selecting windows.
Shift F10	**Switch Through All Windows.** Switch the cursor through all open windows in sequence.
Alt F10	**Switch Between Two Windows.** Switch the cursor between the current and previous windows.

PURPOSE

Toggle keys affect the way that you enter text. Each key has two states — on and off. Standard XyWrite provides five toggle keys:

- Insert/Overstrike [Ins]
- Numeric Lock [Num Lock]
- Scroll Lock [Scroll Lock]
- Caps Lock [Caps Lock]
- Automatic Uppercase [CM AU]

You can add more toggle keys or modify the existing ones (see "Keyboard File" in the Customizing chapter).

Automatic Uppercase is entered at the Command Line; all others are executed by pressing the key shown above. The [Shift] key is also described here.

ACTION

Switching a Toggle Key

All five toggle keys operate similarly. For example, to switch between Insert and Overstrike modes:

Press: [Ins]

Result: Insert mode is indicated by the letter "I" (for Insert) visible at the top right corner of the screen. Overstrike is indicated by the "I" turned off.

Cursor Shape. Another indicator of the Insert typing mode is the shape of the cursor. The cursor is a square in Insert mode and an underline in Overstrike mode.

TOGGLE KEYS

Ins

Insert Key. The Insert key switches between Insert and Overstrike. XyWrite is always in one mode or the other.

- Insert Mode. As you type on the keyboard, the characters are inserted into the text without destroying any of the text already present. The new text pushes the existing text out of its way. XyWrite starts up in Insert mode.
- Overstrike Mode. The characters you type *replace* the characters (or tabs) already in the text. This is useful when you want to modify text from one thing to another.

Num Lock

Numeric Lock Key. Press once to turn on, and again to turn off. (When *on*, the letter "N" is present at the top right of the screen.)

- Numeric Lock On. The numeric keypad shifts to *numbers keys* 0 through 9, for use as a calculator keypad.
- Off. The numeric keypad operates as *cursor keys*.

Scroll Lock

Scroll Lock Key. Press once to turn on, and again to turn off. This key is used in editing User Programs; refer to that section for details. When Scroll Lock is on, the letter "S" is present at the top right of the screen.

Caps Lock

Caps Lock Key. Press once to turn on, and again to turn off. When *on*, the letter "C" is present at the top right of the screen.

- Caps Lock On. All letters are locked in upper-case. No other keys are affected. This contrasts with the Shift key which shifts not only letter keys, but *all* keys, including number, punctuation and cursor keys.
- Off. All keys are un-shifted – letters are lower-case.

CM AU

Automatic Uppercase. This mode causes the first character of each sentence to automatically be entered uppercase. Unlike the other typing modes, this mode is entered (and exited) by executing the AU command from the Command Line. For more details, see the Automatic Uppercase command in the Formatting chapter.

Shift **Shift Key.** *(momentary)* Press and hold down to keep on. Release to turn off.

- Pressed In. All keys are shifted to uppercase, including letter, number, punctuation and cursor keys. If the Caps Lock key is on, depressing the shift key shifts the keyboard to lowercase.
- Released. All keys are un-shifted.

INTRO

Once you have written a document with XyWrite, you will be ready to revise it — that is, unless you are one of those who writes only final drafts. The following functions cover the essentials for making revisions.

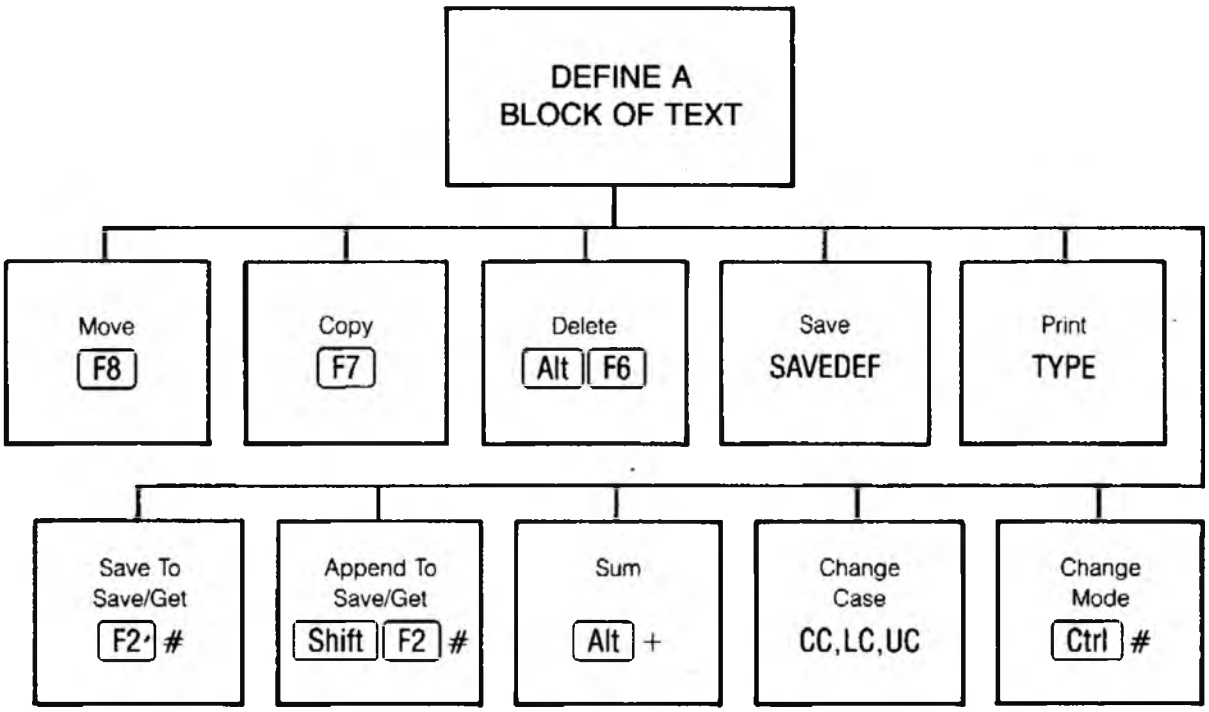
CONTENTS

PURPOSE

When you define text, you are selecting it to be moved, copied, deleted, saved or printed (just to mention a few things). The overall list of possibilities is shown in the diagram below.

Practical Uses. Here are some instances when the block editing features would be useful:

- Define a paragraph in order to move it to another part of your document, or to another document altogether.
- Define a word in order to underline it (MD UL).
- Define a heading in order to capitalize it (UC).
- Define a line of text in order to print out just that line (TYPE).
- Define a sentence in order to save it to a Save/Get Key (F2 #). You can later recall it at the push of a button.



Define Keys

Alt F4	**Define by Word**. This selects the *word* the cursor is on. It also selects the character (space or punctuation) immediately following the word. You can select successive words by holding down Alt and repeatedly striking F4.
Ctrl F4	**Define by Sentence.** This selects the *sentence* that the cursor is on. It selects all characters from the previous period or hard return, up to and including the next period or hard return. You can select successive sentences by repeating the keys. (↵ is the hard return.)
F4	**Define by Line.** This selects the entire line the cursor rests on, from left to right margin. You select successive lines by repeating F4.
Shift F4	**Define by Paragraph.** This selects the entire paragraph the cursor is in. It selects all characters from the previous hard return up to and including the next hard return. (↵ is the hard return.) You can select successive paragraphs by repeating the keys.
F1 ... F1	**Define Any Size Block of Text.** Follow the procedure "Defining a Block of Text." This procedure allows you to select any size block of text, from one character to the entire document.
Alt F1 ... F1	**Define a Column of Text.** Follow the procedure "Defining a Column of Text." This procedure allows you to select a column of text of any size.
F3	**Release Defined Text.** This releases any selected text, so that it can no longer be acted on as a block. Text is returned from bright to dim to indicate it is no longer selected. You are then free to select a new block of text. It is a good habit to release any defined text when you are through using it, as there are a number of functions that do not work while a block of text is defined.

ACTION **Defining Text by Fixed Size**

To define text by word, sentence, line or paragraph:

1. Press: F3 (*Optional*)

   Result: This ensures that no other text is still defined.

2. Move the cursor anywhere within the text you want defined.

3. Simply press the key(s) corresponding to the amount of text you want defined. For example, to define a sentence:

   Press: Ctrl F4.

Result: The defined text appears brighter than normal. You can now move it, copy it, delete it or save it as you wish.

ACTION **Defining A Block of Text**

To select a block of characters of virtually any size:

1. **Release any currently defined block.** (*Optional*)

   Press: F3

   Result: This ensures that no other text is still defined (so that Step 2 sets the *first* of the two end-points).

2. **Begin the block define.** Move the cursor to the first character of the text you want to define.

   Press: F1

3. **Define the size of the block.** Now move the cursor to the other end of the text you want to define. Notice that as you move, the area of text between the cursor and where you started is highlighted.

4. **End the block define.** To set the end of the defined block:

   Press: F1

Result: The block of text is now defined. Now you can move it, copy it, delete it or save it as you wish.

ACTION

Defining by Column

To define any rectangular column of text:

1. **Release any currently defined text.** (*Optional*)

   Press: F3

   Result: This ensures that any currently defined text is released.

2. **Position the cursor.** Move the cursor to the upper left (or lower right) corner of the column you want to define.

3. **Begin the column define.** To establish the first *corner point*:

   Press: Alt F1

4. **Define the size.** Now move to the opposite corner of the column. Notice that as you move, the area of text between the cursor and where you started is highlighted.

5. **End the column define.** When you reach the second corner, end the defined column:

   Press: F1

Result: The column of text is now defined. Now you can move it, copy it, delete it. save it or print it as you wish.

NOTE #1 **Releasing Defined Text.** It is a good habit to release any defined text with F3 when you are through using it.

NOTE #2 **Combining Defined Text.** Alt F4, Shift F4, and Ctrl F4 append text to any text which is already defined. Just press the key to add text to the block. For instance, you can define a word plus the next paragraph plus the next sentence. Notice that you can append text only *adjacent* to the defined text.

NOTE #3 **Define Status Indicator.** If you use the CM command to display the current window number (rather than NM), you will be able to track the status of the define process by referring to your header.

Normally the window field consists of the window number followed by a blank space. When you start defining a block of text by pressing F1, a hyphen appears after the number. When you complete the definition by pressing F1 again or when you press one of the define unit keys (e.g., F4), the hyphen becomes an equals sign. And when you release the defined block by pressing F3, the equals sign disappears and there is again a space after the window number.

NOTE #4 **Size Limitation.** There is a practical limit to the amount of text you can define (determined by the amount of available memory). When you reach this limit, the message "Can't Scroll Define or Display" appears. For more details, refer to the section on Memory.

NOTE #5 **Requirements for Column Define.** The column-defined text requires hard returns as follows: (A hard return is symbolized by a left-pointing arrow at the end of the line.)

- The column you define must have hard returns at the end of every line (wordwrapped lines are not allowed).
- The lines to which you are moving the column of text must end in hard returns.

You can define columns only for copying, moving or changing case. You *cannot* print just a defined column or change its character mode.

NOTE #6 **Storing or Aborting.** Whenever you STORE or ABORT a document, any defined block of text in that document is automatically released.

NOTE #7 **Defining Blocks in Text Tables.** To define a block of text within a Text (Column) Table, refer to the section on Columns in Chapter 5.

FORMAT F7 — Copying a block of text

This is an immediate commmand.

PURPOSE

F7 copies a defined block of text to another part of the document, or to another document altogether. It leaves the original text unchanged.

ACTION

Copying Text

To copy a block of text takes basically three steps:

1. **Define a Block.** Define the block of text you want to move. For details, see the section "Defining a Block of Text" earlier.
2. **Locate the Insertion Point.** Move the cursor to the point in the text where you want to insert the defined text. (The text will be inserted to the *left* of the cursor.)
3. **Copy the Text.** To copy the defined block of text:

   Press: F7

   Result: A copy is made of the text defined in Step 1, and that copy is inserted at the cursor location.
4. **Repeat Copy.** Repeat Steps 2 and 3 to insert more copies, if you wish.
5. **Release Defined Text:**

   Press: F3

   Result: This releases the defined text, completing the operation.

NOTE

Notes. The notes at the end of "Moving a Block of Text," which follows, also apply for copying text with F7.

FORMAT F8 Moving a Block of Text

This is an immediate command.

PURPOSE

F8 moves a defined block of text to another part of the document, or to another document altogether.

ACTION

Moving Text

To move a block of text takes basically three steps:

1. **Define the Text.** Define (select) the block of text you want to move. For details, see the section "Defining a Block of Text."
2. **Locate the Insertion Point.** Move the cursor to the point in the document where you want to insert the text. (The text will be inserted to the *left* of the cursor.)
3. **Move the Text.** To move the block of text:

   Press: F8

Result: The text defined in Step 1 is deleted from its original location and inserted at the cursor location. This completes the operation. F3 is not necessary after Step 3 since the text is automatically undefined when you *move* it (but not when you copy it).

NOTE #1

Moving or Copying Text Between Windows. When you press F8 (or F7 when copying), XyWrite first looks within the current document for the defined block of text to move. If it can't find any defined block there, it will look in the other window for a defined block to move.

NOTE #2

Size Limitation. When defining, moving or copying a long block, XyWrite might beep and give you the message "Can't Scroll Define or Display." This usually occurs when you are working on a large document. The solution is to save the defined block to a Save/Get key (say, F2 A) and release the block with F3. Then move to the point of insertion and press Alt A.

PURPOSE

XyWrite offers many different ways to delete text, as listed on the next page. There is also an undelete feature (included at the end of the list).

ACTION

Deleting Text

To delete text by character, word, sentence or paragraph:

1. Move the cursor onto (or next to, as appropriate) the text to be deleted.
2. Press the appropriate delete key(s) — for example, Del. If you wish, hold down the key(s) to repeat the delete.

Important: If you hold a key down too long, characters may continue to be deleted after you release the key. If that happens, use Ctrl Break to stop it.

ACTION

Deleting a Defined Block of Text

Use this procedure to delete any amount of text — a single character, paragraph or a column or the entire document. This action takes two steps:

1. Define the block of text you want deleted. (For more details refer to the section "Defining A Block of Text" in this chapter.)
2. Press: Alt F6

Result: The text defined in Step 1 is deleted.

There is a practical limit to the amount of text you can define and delete at once. Refer to the explanation in the earlier section "Defining a Block of Text."

Delete Keys

[Del] **Delete Character**. This key deletes the character at the cursor location. When held down, the cursor remains in one place and gobbles up the characters to the right. (When held: **Delete Characters to Right**.)

[Backspace] **Delete Character to Left**. (Backspace) Deletes characters to the left of the cursor.

[Alt] [Del] **Delete Word**. This deletes the word the cursor is on. If the cursor is not on a word, it deletes the word which follows. (When held: **Delete Words to Right**). (See Notes #1 & #2)

[Alt] [Backspace] **Delete Word to Left**. This deletes the word to the left of the word that the cursor is on. (See Notes #1 & #2)

[Ctrl] [Del] **Delete to End of Line**. This deletes from the cursor position to the right end of the line. (See Note #1)

[Alt] [F5] **Delete Entire Line.** This deletes the line the cursor rests on. (See Note #1)

[Alt] [F6] **Deleting a Defined Block.** This deletes whatever block of text is currently defined. Follow the procedure "Deleting a Defined Block of Text." This procedure allows you to delete any size block of text, from one character to the entire document.

[Alt] [F3] **Undelete.** Recovers the most recently deleted word, line or block of characters back into the text. It *cannot* recover characters which were deleted with [Del] or [Backspace]. Note that Undelete works only for the last thing deleted. (See Note #3)

Note: The following Delete functions are not pre-assigned to any keys. To assign them to keys of your choice, refer to the section on Function Calls in the Keyboard File section of Chapter 6. The functions are:

Delete by Sentence (RS)
Delete by Paragraph (RP)
Delete Spaces to the left of the cursor (UP)

NOTE #1

Error Beep. The functions noted earlier will beep if any text is already defined. To avoid the beep, press F3 (to release any defined text) prior to executing the function. For example, press F3 before Alt Del. (These functions do not work if any text is already defined, because they themselves must *use* the feature of defined text.)

NOTE #2

Word Separators. The two noted functions delete all text up to and including any word separator. The most common word separator is the single space. The other separators are:

@ # % & * + = ( ) [ ] { } < > / \ : ; ' " , . ! ? ^

The following four characters are treated as text, not as word separators: $ _ - '

NOTE #3

Undelete Size Limitation. Note that there is a practical limit to the amount of text that you can delete and still recover with Undelete. (You will get the message "Out of Memory" when you reach this limit.) This limit depends on the number and size of files open within XyWrite. You will have more memory available to you if you operate with only one window open.

If you have accidentally deleted a large block of text that cannot be undeleted, you can ABORT and restore the file to its original contents. (This may be practical only if you have *recently* saved your document.)

NOTES

INTRO

If you need to make calculations, you don't need to leave XyWrite for some other program — you can perform calculations right on the Command Line. And in the text area, you can add/subtract numbers one-by-one, total a defined block of numbers in your document, or evaluate a math expression with just a few keystrokes.

CONTENTS

FORMAT		
	+	Addition
	-	Subtraction
	*	Multiplication
	/	Division
	=	Equals

These are immediate commands.

PURPOSE

The math functions enable you to perform math calculations in two ways:

- **Header Calculations.** You can perform addition, subtraction, multiplication and division on the Command Line.
- **Cursor Arithmetic.** You can add or subtract numbers one at a time or in blocks. You can also evaluate any type of math expression by using block define.

The result of any calculation is two decimal positions greater than the number of decimal positions used by the most precise input number. For example, 2/3=.66 but 2.000/3=.66666.

ACTION

Header Calculations.

To perform calculations on the Command Line, you can use addition (+), subtraction (-), multiplication (*), and division (/). End the calculation with the equal sign (=) and ↵(Enter). For example:

Type: F5 3*4-1/2=↵

Result: The answer is 11.5. You may use parentheses to group parts of a calculation, such as (1+2)*3=.

ACTION

Cursor Arithmetic.

There are several methods for summing numbers which are already present in the text.

Adding/Subtracting Numbers One-By-One. *(Method 1)*

1. To add or subtract numbers one at a time, move the cursor onto a number and press Alt + (to add) or Alt - (to subtract).

   Result: The intermediate answer appears on the prompt line.

2. Then move the cursor to where you want the final answer placed, and press Alt = to place the result in the text. (This also clears the internal sum — sets it to zero.)

   You must use the numeric keypad + and - rather than the + and - along the top row of the keyboard.

Totaling a Block of Numbers in the Text. *(Method 2)*

1. Define a block of existing numbers (such as a row or column). To define the block use F1 or Alt F1 as you would define any text.

2. Press Alt + to add or Alt - to subtract. This sums the defined numbers and adds (or subtracts) the total to the internal sum.

3. Then move the cursor to where you want the final answer placed, and press Alt = to place the result in the text. This also clears the internal sum — sets it to zero. Be sure to release the defined numbers with F3.

Evaluating an Expression in the Text. *(Method 3)*

1. In the text area, define a block around any math expression (such as 3*4-1/2). There must be no spaces in the expression. You are allowed to use parentheses. (An equal sign is not required.)

2. Use Alt + to calculate the result and add it to the internal sum, or use Alt - to subtract it.

3. Then move the cursor to where you want the final answer placed, and press Alt = to place the result in the text. This also clears the internal sum — sets it to zero. Be sure to release the defined numbers by pressing F3.

TIP

Clearing Defined Text. Prior to using any math functions, it is a good idea to do two things:

1. Press F3 (to release any defined numbers or text)
2. Clear the internal sum with CLRSUM

This will ensure you are summing only the numbers you have explicitly defined. (See the following Note.)

NOTE

Clearing the Sum. To set the internal sum equal to zero:

Type: F5 clrsum ↵

Result: This clears the calculator to zero. The abbreviation for CLRSUM is CS.

INTRO

This section describes the capability to store and recall frequently-used text. The first section, Save/Get Procedure, covers the overall process; individual commands are described in the second part.

CONTENTS

PURPOSE

Save/Get keys allow you to save information for later recall. You do this by defining a block of text and saving it to any one of thirty-six [Alt] keys. We call these Save/Get keys because you can *save* text on them and later *get* that text back. You can recall the saved information as many times as you want, whenever you want.

Save/Get keys have many uses, including:

1. **Boilerplate Text**. Keep often-used blocks of text *at hand* to insert into a document whenever you wish.

2. **Cut and Paste.** Save blocks of text to insert at other locations in any window.

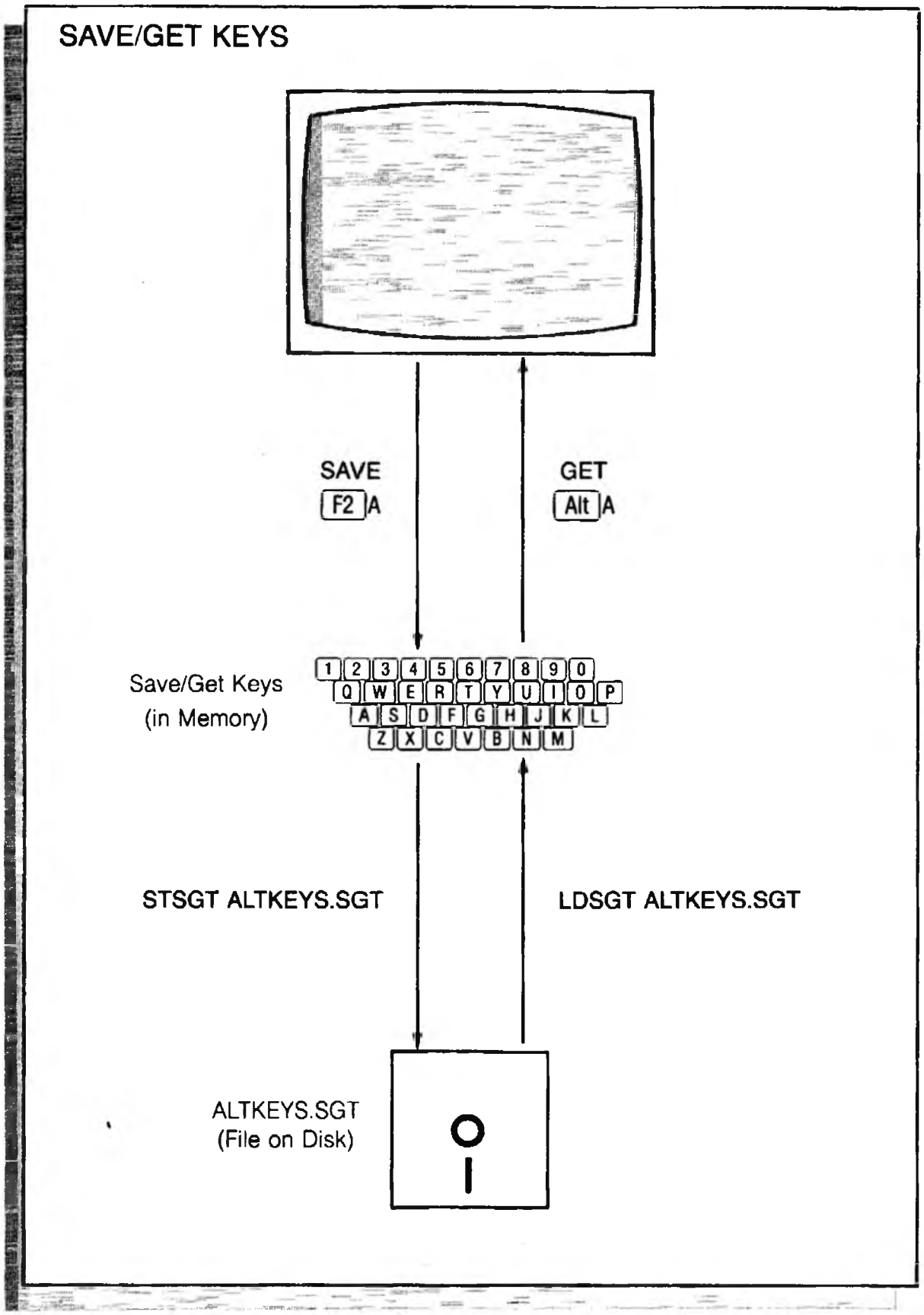
3. **Embedded Commands.** You can save embedded commands (such as LM, RM, TS, IP, MDBO) to a Save/Get key exactly the same way you save text. Then you can switch formats with a simple [Alt] keystroke.

The process is quite simple. For example, to save a sentence to the [Alt]A key, position the cursor within the sentence and define it using the [Ctrl][F4] (Define Sentence) function. Then strike [F2]A (Save Defined Block).

To recall the block you just saved, press [Alt]A. You can recall the block as many times as you desire (until the key is cleared or redefined).

The following procedure runs through the entire sequence of (1) saving text temporarily to Save/Get keys, and (2) saving a set of Save/Get keys to disk.

(Aside: There are 36 other Save/Gets reserved strictly for running programs. These correspond to Function Calls &A to &Z and &0 to &9 which are described at the LDPM command in User Programming in Chapter 5.)


SAVE/GET KEYS
SAVE
F2 A
GET
Alt A
Save/Get Keys
(in Memory)
1 2 3 4 5 6 7 8 9 0
Q W E R T Y U I O P
A S D F G H J K L
Z X C V B N M
STSGT ALTKEYS.SGT
LDSGT ALTKEYS.SGT
ALTKEYS.SGT
(File on Disk)

ACTION **Temporary Use of Save/Get Keys.**

This procedure saves text to memory and not *disk*. Refer to the illustration on the previous page for an overview of the following commands. With a document open, do the following:

1. **Save the Text.** Define the text you want to save. For example, to define a line of text, move the cursor to the line and press [F4].

   To save the text to one of the 36 possible Save/Get keys, press [F2] followed by any letter or number key. To save to letter X, for example:

   Press: [F2]X

   Press: [F3] (to release the defined block)

   If you want to save new text to key X, repeat this procedure. The new text replaces the previous text.

2. **Get the Text.** To insert the text from one of the keys (say, Save/Get X) into the file:

   Press: [Alt]X

   You can repeat Step 2 to insert the text in this or other documents as many times as you wish.

ACTION **Disk Use of Save/Get Keys.**

This procedure shows you (1) how to store the current *set* of Save/Get keys to disk, and (2) how to recall the *set* for use at a future editing session. (Refer to the previous illustration.)

1. **Store Save/Get Keys to Disk.** First view the current set of Save/Get keys (this is the set you will store):

   Press: [Alt][F2]

   To return to the document, after viewing, press [Space Bar]. To save the current set of Save/Get keys:

   Type: [F5]stsgt altkeys.sav[↵]

   You can save to any filename you wish. You can even store different sets of [Alt] keys under different filenames, and later recall whichever set you need.

2. **Loading Save/Get Keys from Disk.** Once the Save/Get keys ([Alt] keys) have been saved to disk, they can be loaded back into use at other editing sessions as follows.

   Enter LDSGT along with the name of the Save/Get set you stored in Step 1.

   Type: [F5]ldsgt altkeys.sav[↵]

   When this new set of keys is loaded, it overlays the previous set. Unassigned keys in the new set *do not disturb* previously assigned keys. (If you would prefer to clear the old [Alt] keys before loading a new set, enter CLRSGT at the command line.)

   You may include a LDSGT command in the STARTUP.INT file to automatically load a particular set of [Alt] keys every time XyWrite is started.

   (Optional) To check that the [Alt] keys were indeed loaded, view them by pressing [Alt][F2]. When done viewing the [Alt] keys, press [Space Bar].

NOTE #1 **Removing a Save/Get.** The process of storing new text to a Save/Get key clears the previous text from that key. You can also use the REMOVE command (described later in this section) to clear text from a key.

NOTE #2 **Attention!** Be aware that a Save/Get file is *not* an ordinary file. Just as you don't store it with the usual STORE command, you cannot call it for editing with CALL. If called up with CALL it is not readable; and if it is stored with STORE, the file will be ruined. ***This file cannot be edited directly.*** Instead, get the text to be edited into a regular file, (e.g., using [Alt]X), edit it and SAVE the new version to the same key (using [F2]X). Then store the set of Save/Gets back to disk with STSGT.

NOTE #3 **TYPE % and SAVE %.** You can print the contents of a single Save/Get using the TYPE % command — for example, use TYPE %A to print the Contents of Save/Get A. Similarly, use SAVE %A to save the Contents of Save/Get A to disk (it is saved as A.SAV).

FORMAT	F2 #	Save Text
	# is a letter (A-Z) or number (0-9) key. This is an immediate command.	

PURPOSE

F2 # copies the defined block of text to the Save/Get key you specify. (This is a short-term Save — that is, the text is saved until you QUIT XyWrite.) You can recall the text at any time by pressing Alt and that same letter or number key. You can save as much text as memory allows — typically up to 10K or 20K. For an overview of the Save/Get key procedure and what it's used for, see the previous section "Save/Get Procedure."

If text is already present in the Save/Get key, this command erases that text before saving the new text. If you wish to keep the text which is there and simply add text to the end of it, use the "Append to a Save/Get key" command.

ACTION

Temporary Save to a Save/Get Key

To save text until you quit XyWrite:

1. **Define the Text**. Define the text you want to save.
2. **Save the Text**. Choose which key you want to assign to the defined block: A-Z, or 0-9. To save to the X key, for example:

   Press: F2 X

   Result: Any text previously saved to the Alt X key (if any) is erased, and the text defined in Step 1 is saved to that key. The prompt line then says "DONE."

3. **View the Alt Key**. (Optional) To view the text saved to that one key:

   Press: Ctrl F2

   Press: X

   After viewing the text, press Space Bar (or any key) to return to the document.

FORMAT Alt * Get Text

* is a letter (A-Z) or number (0-9) key.
This command is an immediate command.

PURPOSE

Alt * copies text from the Save/Get key to the cursor location. You can recall the text at any time (as many times as you wish). This is the "Get" operation of the Save/Get keys.

You normally "get" text in order to copy it to another location. For an overview of the Save/Get key procedure, see the earlier section "Save/Get Procedure."

Alt * inserts text into the text area only — not on the Command Line (unless the Save/Get contains a program).

ACTION

Getting Text from a Save/Get key

Use this procedure to insert text which has been previously saved to a Save/Get key.

1. Move the cursor to the spot in the document where you want to insert the Save/Get text.
2. Press Alt along with the key you want. For example, to get the text from Save/Get key A:

   Press: Alt A

Result: This gets the text from the Alt A key and inserts it into the document at the cursor location.

FORMAT Alt F2 Display Save/Get Directory

This is an immediate command.

PURPOSE

Alt F2 displays the entire set of Save/Get keys. This enables you to check which Save/Get keys are currently available — to remind you what text is saved to each key.

ACTION

Displaying the Save/Get Directory

To display the currently loaded set of Save/Get keys:

1. Press: Alt F2

   Result: The Save/Get keys are listed on the display. Each entry begins with its identifying letter or number.

   There is one line per Save/Get — only the first 35 characters of each Save/Get are displayed. To view up to one screenful of text, use the "Display Save/Get Key" Command which follows.

2. When done viewing:

   Press: Space Bar (or any other key)

   Result: This returns you to your document.

FORMAT Ctrl F2 # — Display Save/Get Keys

is a letter (A-Z) or number (0-9).
This is an immediate command.

PURPOSE

Ctrl F2 # displays the text in the Save/Get key you specify. This enables you to read the text before inserting it into a document. If the text is more than one screenful, only the first screen is shown.

This command is similar to the previous command "Display the Save/Get Directory."

ACTION

Displaying a Save/Get Key

To display the contents of a Save/Get key:

1. Press Ctrl F2 together, then the letter or number you want to view. For example, to view key A:

   Press: Ctrl F2

   Press: a

   Result: The contents of Save/Get A is displayed. Up to a screenful of text is displayed, although the Save/Get key can hold more than that.

2. When done viewing:

   Press: Space Bar (or any other key)

   Result: This returns you to your document.

FORMAT Shift F2 # — Append to Save/Get Keys

is a letter (A-Z) or number (0-9).
This is an immediate command.

PURPOSE

Shift F2 # appends the currently defined text to the end of the Save/Get key you specify.

One use might be for re-arranging blocks of text; you could add blocks of text to a Save/Get key in the order you wish, and then recall the entire series of blocks.

ACTION

Appending to a Save/Get Key

To append text to a Save/Get key:

1. **Define the Text**. Define the text you want to append.
2. **Append the Text**. Choose which key you want to append to: A-Z, or 0-9. To append to the X key, for example:

   Press: Shift F2

   Press: X

   Result: Any text previously saved to the Alt X key (if any) is kept, and the text defined in Step 1 is added to the end. The prompt line then says "DONE."

3. **View the Alt Key**. (Optional) To view the text saved to that one key:

   Press: Ctrl F2

   Press: X

   After viewing the text, press Space Bar to return to the document.

FORMAT CM **STSGT** *filename* Store Save/Get Keys

filename identifies the file on disk to which the Save/Get keys will be stored.
This is an immediate command.

PURPOSE

STSGT (Store Save/Get keys) stores all of the currently active Save/Get keys to the specified file on disk. This enables you to reload the keys for use at a later editing session.

ACTION

Storing Save/Get Keys to Disk

This procedure saves Save/Get keys to disk. Refer to the illustration in the section "Save/Get Procedure".

1. **View the Save/Get Keys.** (Optional) To view the text which will be saved to disk:

   Press: Alt F2

   After viewing the text, return to the document by pressing ↵.

2. **Disk Save (Long-Term Save).** To save to disk all of the keys viewed in Step 1:

   Type: F5 `stsgt altkeys.sav` ↵

   In this case, ALTKEYS.SAV is the filename to which the keys are stored; you can use any filename you wish. You can even store different sets of Alt keys under different filenames and later recall whichever set you need.

   If you are using subdirectories, and plan to load the file with STARTUP.INT, be sure to save the keys to the same directory that has EDITOR.EXE.

FORMAT **CM LDSGT** *filename* Load Save/Get Keys

filename is the file from which the Save/Get keys will be recalled.
This is an immediate command.

PURPOSE

LDSGT (Load Save/Get Keys) loads all of the Save/Get keys from the specified file on disk. This enables you to use the keys saved in a previous editing session. To see how this complements the STSGT command, refer to the illustration in the earlier section "Save/Get Procedure."

ACTION

Loading Save/Get Keys from Disk

To load a set of the Save/Get keys from the disk and restore them to use, enter the LDSGT command along with the name of the Save/Get file you want to load. For example:

Type: F5 `ldsgt altkeys.sav` ↵

(Optional) To check that the keys were indeed loaded:

Press: Alt F2

When done viewing the Alt keys, press Space Bar. You may now insert the text from any of these keys into any file you call up.

NOTE #1

Overlaying Sets of Save/Get Keys. Loading a set of Save/Get keys replaces only those keys contained in the set you are loading. The contents of all other keys remains unchanged. For example, if keys A,B,C and D are originally defined, and you load a new set with keys C,D,E and F, you will end up with the old A, old B, new C, new D, new E and new F.

NOTE #2

Clearing Save/Gets. If you would prefer to clear the Alt keys before loading a new set, use CLRSGT.

TIP

Starting XyWrite with Save/Gets Loaded. You can include LDSGT in the STARTUP.INT file to automatically load the Alt keys every time XyWrite is started.

FORMAT		
	CM **CLRSGT**	Clear All Save/Gets
	CM **REMOVE #**	Clear Single Save/Get

is the Save/Get key — any single letter (A-Z), single number (0-9), &A-&Z or &0-&9.
CLRSGT and REMOVE are immediate commands.

PURPOSE

CLRSGT (Clear Save/Get Keys) clears *all* of the current Save/Get keys from memory. You might do this before loading in a new set of Save/Get keys.

REMOVE (Remove Save/Get Key) clears any *single* Save/Get key (or user program) from memory.

CLRSGT and REMOVE have no effect on any Save/Get files stored on disk.

ACTION

Clearing All Save/Get Keys.
To eliminate all current Save/Get keys from memory:

Type: F5 clrsgt ↵

Result: All Save/Get keys are now cleared from memory. This operation does not affect any Save/Get files on disk.

ACTION

Clearing a Single Save/Get Key.
To clear a single Save/Get key from memory, enter REMOVE followed by the name of the Save/Get (A-Z, 0-9, &A-&Z, or &0-&9). For example, to clear Save/Get X:

Type: F5 remove x ↵

Result: Save/Get X is now cleared from memory.

NOTE

Save/Gets Reserved for Programming. Save/Gets labeled &A to &Z and &0 to &9 are reserved strictly for user programming. These are described in the LDPM section of User Programming in Chapter 5.

FORMAT	CM IS #	Insert Save/Get Text
	# is any letter or number. IS is an embedded command.	

PURPOSE

The **IS** (Insert Save/Get) command allows you to insert any Save/Get block into the text at printout. IS performs the same function as [Alt]A except ▲IS:A is displayed on-screen instead of the actual text.

The IS command gives you the ability, for example, to make up a form letter using the text from various Save/Get keys. To do this, save each block you want inserted in the letter to a Save/Get, insert an IS command at each point in the letter where you want Save/Get text inserted, and then print the letter.

ACTION

Entering an Insert Save/Get Command.
To enter an Insert Save/Get command into your text:

1. Position the cursor where you want to insert the Save/Get text.

2. Enter the IS command along with the Save/Get letter or number key. For example, to insert Save/Get X:

   Type: [F5]is x[↵]

   Result: The IS command appears in the text as

   ▲IS:X

INTRO

Rather than scrolling screen-by-screen through the text hunting for a word, you can learn to make the computer work for you. The Search and Change commands can help you improve the speed at which you revise text. Some time spent learning these commands can pay off well.

CONTENTS

FORMAT

CM SEARCH */string/*	Search forward
CM SEARCHA */string/*	Search forward, absolute match
CM SEARCHB */string/*	Search backward
CM SEARCHBA */string/*	Search backward, absolute match

ABBREV

CM SE */string/*
CM SEA */string/*
CM SEB */string/*
CM SEBA */string/*

In place of the slash (/), you can use any character that does not appear in the string (See Note #4).

All of these are immediate commands.

PURPOSE

The **SEARCH** command and its variations allow you to search through a document to find the text you specify *(string)*. You have the four search choices listed above. You can search in either direction — the search continues until it finds the string or reaches the end of the document. To search through more than one file, refer to the next section "Searching Through Files."

Absolute Case Match. The **A** (absolute) at the end of SEARCHA and SEARCHBA means absolute case match: The search stops only for text that has exactly the same uppercase and lowercase letters that you specify — they must match letter-for-letter. See the examples that follow.

String. The *string* shown above in the format statements includes all characters that appear between the two slashes, including spaces, punctuation and symbols.

Search and Replace. If you want to search *and replace* the text, refer to the CHANGE commands in a later section.

Wildcard Search Characters. At the end of this section is a list of wildcard search characters which are very useful in searches.

ACTION

Search for Text

To search for text in a document:

1. Move the cursor to the point where you want the search to begin.
2. Enter the SEARCH command or variation, along with the text you're looking for. For example, to search forward for the word "orange":

   Type: F5 search /orange/ ↵

   Result: The search begins at the cursor location and searches forward, stopping at the first occurrence of "orange".
3. To search for the next occurrence of "orange":

   Press: F9

   Note: Continuing the search with F9 works only if the search command is still on the Command Line.

EXAMPLES

CM search /orange/	Begins at the cursor location in text and searches *forward*, stopping at the first occurrence of "orange" (or "Orange", or "ORANGE", accepting letters of either case).
CM searcha /orange/	Search stops only for "orange" (it would skip over "Orange" and "ORANGE").
CM searchb /orange/	Begins at the cursor location and searches *backward*, accepting either case.
CM seba /orange/	Searches backward only for "orange" (skipping over "Orange" and "ORANGE").

NOTE #1 **Shortcut.** To search for just *one word*, you can use:

```
CM se  orange
```

Note there are *two* spaces between "se" and "orange". This does not work when searching for more than one word. (See Note #4.)

NOTE #2 **Spaces are Characters, Too.** Spaces are counted in the search the same as any printable character. For example:

```
search /babysit/
```

would *not* stop at "baby sit". Because of the space, the two words do not match.

NOTE #3 **Special Characters**. All characters which appear between the slashes (/) are searched for, including the following special characters:

Character	Key
Tab	Tab
Space	Space Bar
Carriage return	Ctrl ↵
Left double-angle bracket («)	Ctrl <
Right double-angle bracket (»)	Ctrl >

NOTE #4 **The Slash (/) Separator.** You are not *required* to use the slash (/) to begin and end the string. You can use any character which does not appear in the string. In fact, you must *not* use a slash if it appears in the string (see the next Note). You can even use the single space as the separator as long as the space is not included in the string — this is why the shortcut in Note #1 works. (The Command Line is initially a row of spaces, so you get a space at the end of the word automatically).

NOTE #5 **Searching for a Slash (/).** If you are searching for text which contains a slash character (/), such as *miles/hour*, you must use a character other than the slash to enclose the text — choose any character which does not appear in the text. For example, you might choose the double quotation mark (").

```
CM search "miles/hour"
```

NOTE #6 **Searching for Embedded Text.** To search for embedded text, you must switch to Expanded Display. Embedded text is visible in Expanded Display within double-angle brackets, such as left margin «LM», right margin «RM», indent paragraph «IP0,10», page break «PG», and tab set «TS5,10,15».

This capability can be very useful. To search backward for the previous tab setting, you would type:

```
F5 seb /«TS/↵
```

Enter the left double-angle bracket by pressing Ctrl <. (If you did not include the double-angle bracket, the search would stop at any word containing ts, such as *itself*.)

NOTE #7 **Display Modes.** A search will not match text which is *partially* bold, underline, or reverse. For instance, SEARCH /fulltime/ will stop on "fulltime", but will *not* stop on "fulltime". This is because the latter has «MDUL» embedded in it, which is seen as characters in the search. In Expanded Display you would see:

```
full«MDUL»time
```

NOTE #8

Wild Card Characters. The following wildcard characters can be used in searches. We call these *wildcard* characters because (like joker cards in poker) they can represent other values. They provide you with powerful search capabilities.

Any Single Letter (A-Z): L Press: Alt Shift L
You can read this wildcard as any letter. It allows any of the 26 letters of the alphabet to take its position in the text.

Any Single Number (0-9): N Press: Alt Shift N
You can read this wildcard as any number. It allows any of the 10 numbers to take its position in the text.

Any Single Number or Letter: A Press: Alt Shift A
You can read this wildcard as any number or letter. It represents any of the 26 letters or 10 numbers when the search is executed.

Any Single Character: X Press: Alt Shift X
You can read this wildcard as any character. It represents any character, letter, number, punctuation, space, symbol, or other graphic mark of the 255 characters of the ASCII set when the search is executed.

Any Single Separator: S Press: Alt Shift S
You can read this wildcard as any separator character. The most common word separator is the single space. The other separators are:

@ # % & * + = ( ) [ ] { } < > : ; ' " , . ! ? $ _ – / ~

Any String of Characters: W Press: Alt Shift W
You can read this wildcard as any string up to 80 characters in length. It can include any characters from the 255 characters of the ASCII set. This wildcard must be used with at least one other character.

EXAMPLES

CM search /computeL/	Finds "computer" and "computed"
CM search /chapter N/	Finds "Chapter 1" and "Chapter 2"
CM search /AA6-8964/	Finds "386-8964" and"EU6-8964"
CM search /8X23X86	Finds "8-23-86" and "8/23/86"
CM search /SroseS/	Finds "rose" when it stands alone — it would not stop at "primrose" or "rosemary".
CM search /AlexW Bell/	Finds "Alex Bell", "Alexander Bell" and "Alexander Graham Bell".

NOTE #9 **Searching for Just a Word.** The single separator wild card Alt Shift S deserves special mention. It allows you to search for just a word, as shown in the example above. It skips over occurrences of the string as part of a larger word.

NOTE #10 **Related Commands.** If you want to search for text, and *replace* it with other text, refer to the CHANGE commands which follow.

NOTE #11 **Stopping a Long Search.** Use the Ctrl Break key to stop a lengthy search that is in process.

FORMAT	CM **SEARCH** *range/string/*	Search files
	CM **SEARCHA** *range/string/*	Search files, absolute match
ABBREV	CM **SE** *range/string/*	
	CM **SEA** *range/string/*	
	range is *globalname1,globalname2,globalname3, . . .*	
	globalname is described below.	
	These are immediate commands.	
EXAMPLE	CM se *.*/orange/	

PURPOSE

When you specify a *range* along with the **SEARCH** command, you can search across multiple files to find the string of text you want. You would use this procedure when you are looking for text but are not sure what file it may be in. The other forms of the command (SEARCHB, SEARCHBA, CHANGE) do not work across multiple files.

Unlike the search procedure of the previous section, when you search through more than one file, *you must start with an empty window*. XyWrite will allow you to browse quickly through the files, one at a time.

Range. The *range* is the series of filenames you want to search through, separated by commas (but no spaces following the commas).

Globalname. The *globalname* can be any filename, such as B:CHAPTER.DOC. It can also be any global filename using * or ?, such as A:*.* or CHAPTER?.DOC or B:*.TXT. You can include a drive letter and path. Globalnames are described further under the DIR command in Chapter 2.

String. The *string* is the same as defined earlier for the SEARCH command. It can include any wildcard characters, as described in the previous pages.

ACTION **Search for Text**

To search across multiple files for text:

1. Move to an empty screen (where no file is open). For example, press Ctrl F10 3.
2. Enter the SEARCH or SEARCHA command followed by the filenames. Follow this with the string you want to search for. For example:

   Type: F5 search a:*.*,b:*.doc/orange/ ↵

   Result: The search looks for the first occurrence of "orange", first searching through the files on drive A and then through the files with the extension DOC on drive B.

3. Select your response. Type C, O, S or N:

   C Continue searching for the next occurrence.

   O Open the file that is shown on the screen.

   S Stop the search and clear the screen.

   N Next file — skip to the next file and continue the search.

Result: If you type C or N, the search continues until "orange" next occurs. If you type O or S, the search stops.

NOTE #1 **Stopping a Long Search.** Use the Ctrl Break key if you want to stop a lengthy search that is in process.

NOTE #2 **Searching for a Slash.** If a slash (/) appears in the string you are searching for, then use another character, such as the quote ("), to set off the string.

FORMAT

CM CV */string1/string2/*	Change and verify
CM CVA */string1/string2/*	Change and verify, absolute
CM CH */string1/string2/*	Change (no verify)
CM CHA */string1/string2/*	Change (no verify), absolute
CM CI */string1/string2/*	Change Invisible
CM CIA */string1/string2/*	Change Invisible, absolute

A means absolute case match (described below).
string1 is the text being searched for.
string2 is the text which is inserted into the text, replacing *string1*.
All of these are immediate commands.

PURPOSE

Each of the CHANGE commands searches forward through the document to find the text you specify, in order to replace it. The search ends at the bottom of the document. To search *without* replacing, see the SEARCH commands.

Verify means that when XyWrite finds the string, it asks you whether or not to make the change. You must respond before it will look for the next occurrence.

Absolute Case Match. The "A" (absolute) at the end of CVA, CHA, and CIA means absolute case match: The search looks only for text that has exactly the same uppercase and lowercase letters that you specified in *string1*. They must match letter-for-letter.

Replacement Text. *string2* is always inserted into the text "as is", with its letters uppercase or lowercase *exactly as you typed them*.

Change Invisible. CI and CIA do not refresh the display while making changes, and so execute quicker than the other commands.

EXAMPLES

CM cv /orange/grape/	Changes every instance of "orange", "Orange" and "ORANGE" to "grape", stopping each time to allow you to verify each change.
CM cva /orange/grape/	The same as CV, but skips over "Orange" and "ORANGE".
CM ch /orange/grape/	Changes every instance of "orange", "Orange" and "ORANGE" to "grape". Runs non-stop, *without verifying*. Each change is visible on the display.
CM cha /orange/grape/	The same as CH, but skips over "Orange" and "ORANGE".
CM ci /orange/grape/	Changes every instance of "orange", "Orange" and "ORANGE" to "grape". The changes are *not* displayed until they are all done. This command is much faster than CH.
CM cia /orange/grape/	The same as CI, but skips over "Orange" and "ORANGE". This command is much faster than CHA.

ACTION **Changing Text with Verifying.**

CV and CVA: To search for text and have XyWrite stop to ask you to verify each change:

1. Move the cursor to the point in text where you want to begin the search.

2. Enter CV or CVA. For example, to search for the word "orange" and replace with "grape":

   Type: F5 cv /orange/grape/↵

   Result: The search begins at the cursor location and continues forward, stops at the first occurrence of "orange" and asks you to verify the change. Since we specified CV (and not CVA, the search stops for "Orange" or "ORANGE" or any other combination of upper and lowercase letters.

3. Verify the Change. Type A, Q, N, S, or Y:

   A Abandon the search (without replacing the text) and return cursor to initial starting point.
   Q Abandon the search (without replacing the text) and leave the cursor at the current point.
   N No, do not replace the text; continue the search.
   S Stop after replacing the text.
   Y Yes, change the text and continue the search.

Result: If you type Y or N, the search continues for the next occurrence of "orange". If you type A, Q or S, the search stops.

ACTION **Changing Text Without Verifying.**

CH, CHA, CI and CIA: To search for text and change it *without* it stopping for verification:

1. **Save the Document**. As a precaution, before making changes, it is a good idea to SAVE your document, (especially when using CI or CIA). This provides a copy of the document on disk, allowing you to recover the original should you mistakenly change text you did not intend to change.

2. Enter CH, CHA, CI or CIA. For example, let's use CH to search for the word "orange" and replace with "grape":

   Type: F5 ch /orange/grape/ ↵

   Result: The search begins at the cursor location and continues forward; at each occurrence of "orange" it removes the word and replaces it with "grape". Since we specified CH (and not CHA), the search stops for "Orange" or "ORANGE" or any other combination of lowercase and uppercase letters. The changes continue non-stop until the end of the document is reached, at which point the prompt line says DONE.

3. Emergency Stop. If you need to stop a search before it reaches the end, press:

   Ctrl Break

NOTE

Deleting Text. You can use the CHANGE commands to delete text. You simply omit *string2* from the command (but keep the three slashes). For example, to delete the word "orange" from your document, use:

```
ch /orange//
```

ALSO SEE

Related Commands. Refer to the SEARCH commands, to search *without* replacing text. The notes in that section also apply for these CHANGE commands with two exceptions:

- All CHANGE commands search in a *forward* direction — you cannot search and replace backwards.
- Unlike SEARCH commands, the CHANGE commands do not allow the use of wildcard characters.

FORMAT CM GO *m-n* Go to Page and Line Number

m is the page number
(If *m* is omitted, GO uses the current page).
n is the line number
(If *n* is omitted, GO uses line 1).
GO is an immediate command.

PURPOSE The **GO** command allows you to go directly to the page and line number you specify. This page and line number corresponds to the PG-LN appearing in the upper right corner of the screen.

ACTION

Moving to a Page and Line Number
To move to certain page and line number in your document:

Enter the GO command with the page and line number. For example, to go to page 4, line 28,

Type: F5 go 4-28 Enter

Result: The page-line number turns on (if not already on) and the cursor moves to the first character position of line 28 on page 4.

NOTE #1 **Options.** You can move to a specific line on the *current* page by omitting the page number. You must precede the line number with a hyphen. For example, GO -3 moves the cursor to line 3 of the current page.

To move to the first line of any page, as a shortcut, specify only the page number. For example, GO 11 moves the cursor to line 1 of page 11.

NOTE #2 **JUMP Command.** The JMP (Jump) command allows you to jump to a specific character within the current file. Use the form: JMP *n* where *n* is the number of characters from the start of the file. For example, JMP 9885 positions the cursor on the 9885th character of the file. Each ↵ (Enter) counts as two characters: Carriage Return / Line Feed. The characters within embedded commands (that appear in the Expanded Display) also count — thus, «RM70» counts as 6 characters.

FORMAT		
	Ctrl -	Find Difference
	Ctrl =	Find Match

PURPOSE

The **Compare** function allows you to examine two similar files character-by-character to find likenesses and differences. You might use this function to compare an edited version of a file against the original.

Compare uses two commands: Find Difference Ctrl - and Find Match Ctrl =. You may start the comparison with either command, but you must use them *alternately* to find matches and differences between the two files.

XyWrite defines a match as 80 consecutive matching characters. This means that Compare does not stop at insignificant matches such as the word "the." A single character defines a file difference.

ACTION

Comparing Two Files.

To compare two files, say, your draft version of a document with an edited version:

1. Call the first file to the screen.

   Type: F5 call draft ↵

2. Open a second window and call the second file to the screen.

   Press: Alt F10

   Type: F5 call chapter ↵

3. Search for the first *difference* between the files:

   Press: Ctrl -

   Result: Compare searches through both files until it finds a difference. The cursor stops at that point in both files. Use Alt F10 to toggle between the two files to see exactly what the difference is. You can make an edit in either file or proceed to the next step.

4. Search for the next *match* between the files.

   Press: Ctrl =

   Result: Compare searches through both files until it finds a match. The cursor stops at that point in both files.

5. Continue alternating the Find Difference (Step 3) and Find Match (Step 4) commands until you have finished the comparison.

NOTE #1 **Comparing Embedded Commands.** The Compare function works better at comparing embedded commands (such as RM, TS and MDBO) if you switch to Expanded Display in both files before starting the comparison. (Use Ctrl F9.)

NOTE #2 **Cursor Location.** The Compare function starts its search at the current cursor locations in both files. Be sure that the cursor starts at the same point in both files or Compare will not find where the files match.

INTRO

XyWrite gives you access to more than one document at a time. In fact you can view as many as *nine* documents at once. XyWrite does this by displaying each document in a separate window. These rectangular windows can be any size you want and can overlap. You control how the windows are displayed.

CONTENTS

PURPOSE

Quite often it's handy to be working on two files at the same time, or maybe even three or more. The windows feature allows you to do just that. With windows you can switch between documents with one or two keystrokes. You can access as many as *nine* documents this way. You can arrange these files inside windows that overlap or are side by side. For example, you can refer to several different note files quickly and easily as you are working on a chapter of your novel.

A **window** is a rectangular area on the screen through which you can view a document, as in the illustration below. Each document requires its own window. XyWrite starts with a single full-screen window. You can *open* more windows in order to view other documents without storing the documents that are already open. You can move or copy text directly from one window to another.

Open Window. When you are using more than one window, then the windows that you are using are called *open* windows. These windows can have documents in them or not. A window which is not open is labeled NOT IN USE in the window menu. Once you open a window, it remains open until you close it (although it may be hidden behind another window).

Active Window. When you have more than one window open, only one is active at a time; the others are *suspended*. The filename at the top of the screen tells you which file is active. You can type into the active document, modify it, scroll it and SAVE it.

The keys Shift F10 and Alt F10 control the movement of the cursor from window to window. The cursor remembers its position in each document, so that it can return to where you last left it.

Window Numbers. Each window is numbered 1-9. You can display the window number and filename for each window by moving the top border down one or more lines. (This is apart from displaying the window number and filename in the header at the top of the screen.) Use the procedure "Changing the Size of a Window."

Quick Windows. You can call up windows quickly — for example, strike Alt 9 to get window 9. You would do this by assigning function call #9 to the Alt-9 key in the Keyboard File. See #1 to #9 Function Calls in Chapter 6. Also see Super Keyboard, Appendix E.

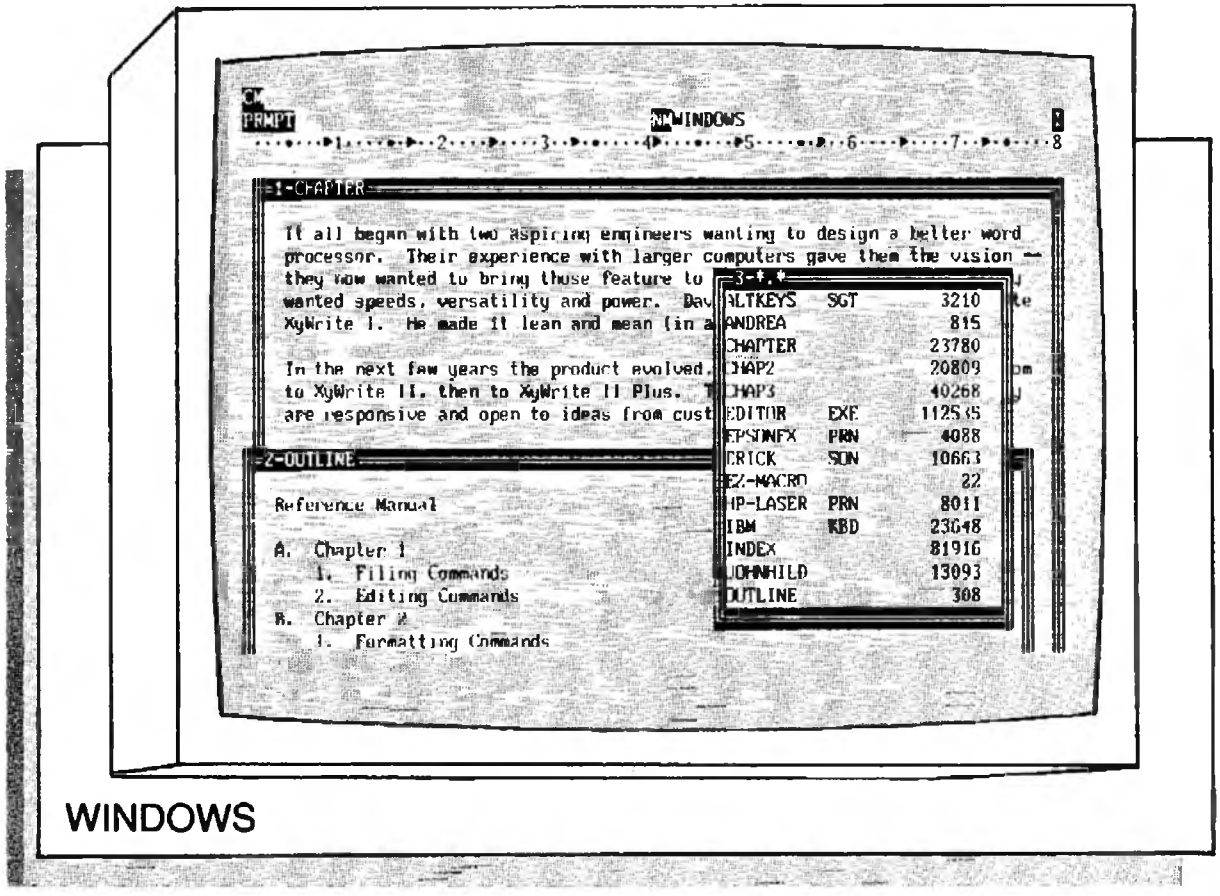
Procedures. Refer to the sections that follow:

Window Menu Ctrl F10
- Displaying the Window
- Opening a Window
- Changing the Size of a Window
- Changing a Window to Full Screen Size
- Moving to Another Window by Window Number
- Removing the Borders from All Windows

Switching Windows Alt F10, Shift F10
- Switching Between Two Windows
- Moving to the Next Window

WINDOW Command
- Opening a Window with the Window Command

RMVSCR Command
- Closing a Window



FORMAT Ctrl F10 — Display the Window Menu

This is an immediate command.

PURPOSE

Ctrl F10 provides a selection menu to handle all of the options that are available for controlling windows within XyWrite. You can:

- Open and close windows.
- Change the size of windows.
- View the contents of the nine windows.
- Conceal or display the window borders.

ACTION

Displaying the Window Menu

When you want to open another window, move to another window or re-size the current window:

1. Press: Ctrl F10

   Result: This gives you a full screen menu that explains the options that are available.

2. If you have no selection to make, press A to abandon the menu.

ACTION

Opening a New Window

When you open a new window, the next available window number is used (unless you specify a number 1-9 to open it). To open a new window:

1. Press: Ctrl F10

2. Select how you want the new window to appear. Type H, V, N or a window number, 1, 2, 3, 4, 5, 6, 7, 8 or 9.

   H Splits the active window in half horizontally and opens a new window in the *lower* half of the screen.

   V Splits the active window in half vertically and opens a new window in the *right* half of the screen.

   N Opens the next available window as a full screen.

If you want to open a particular window, such as window 3, simply type that number, as follows. (However, it is usually more convenient to use N above, since it automatically chooses the next available window for you.) Select a window number (1-9) from any of those not in use. For example:

3 Opens window 3 as a full screen.

Automatic New Window. As a convenience, you can set XyWrite to open a new window *automatically* — if you enter CALL, DIR or NEW but already have a file open on screen. To set this up, use DEFAULT NW=1. (See the Default commands in Chapter 6 for more information.)

ACTION

Changing the Size of a Window

To change the size of a window:

1. Move the cursor to the window whose size you want to change.
2. Press: Ctrl F10
3. Select T, B, L, or R:

   T to move the top border.
   B to move the bottom border.
   L to move the left border.
   R to move the right border.

4. Use the cursor keys to move the selected border. Move the Top and Bottom borders with the cursor up or cursor down keys. Move the Left and Right borders with the cursor left or right keys. You can move a border beyond the edge of the screen, to make more room for the text (just as you might expect).
5. Continue selecting the borders and moving them with the cursor keys until you are satisfied. Notice that *you can set all four borders without returning to the Window Menu*, by selecting T, B, L and R one after another.
6. Press ↵ to complete the selection.

You can accomplish the same thing with the WINDOW command. See the description later in this section.

ACTION **Changing a Window to Full Screen Size**

To restore a window to a full screen:

1. Move the cursor to the window you want to be full screen.
2. Press: Ctrl F10
3. Press: F

ACTION **Moving to Another Window by Window Number**

When you want to move to another window:

1. Press: Ctrl F10
2. In the list at the bottom of the menu, find the file (or the unused window) you want to move to, then type the number of that window. To move to window 3:

   Press: 3

Result: The cursor moves to the selected window.

ACTION **Removing the Borders from All Windows**

The borders are the double lines that define the window boundaries. You can hide the borders at any time:

1. Press: Ctrl F10
2. Press: C

Result: The borders are concealed. By repeating the procedure you can display them again.

NOTE #2 **Viewing One File in Two Windows.** In XyWrite, each window displays a separate file. Thus, if you call the *same* file into two (or more) windows, you are viewing more than one copy of the same file. Changes you make in one copy are not automatically made on the other. Therefore:

CAUTION: When viewing the same file in two windows, make changes to only one copy. STORE that copy back on the disk; ABORT the other.

FORMAT

Alt F10	Switch Between Last Two Windows
Shift F10	Switch to Next Open Window

These are immediate commands.

PURPOSE

The main use for Alt F10 is to move back and forth between two windows — that is, between the currently displayed window and the one displayed previously. (If a second window is not open, Alt F10 automatically opens the next window that is NOT IN USE.)

You use Shift F10 to cycle through all of the open windows (up to nine).

ACTION

Switching Between Two Windows

To move the cursor back to the window it was in prior current window:

Press: Alt F10

To return to the window you just now left, press the same keys again:

Press: Alt F10

Result: By successively pressing Alt F10, you can switch back and forth between the same two windows. To select a new pair of windows, select them one at a time from the Window Menu.

ACTION

Moving to the Next Window

To move the cursor to the next open window (of the nine windows):

Press: Shift F10

Result: The cursor moves to the next window. By successively pressing Shift F10, you can cycle through all the open windows.

FORMAT CM WINDOW *#,left,top,width,length*
is the window number you are defining (1-9)
left is the column number of the border (0-80)
top is the line number of the top border (0-22)
width is the number of columns wide for text (1-80)
length is the number of lines of text (1-22)

This is an immediate command.

EXAMPLE CM window 3,40,1,35,10

PURPOSE

The **WINDOW** command lets you define a window from the Command Line without going through the window menu. It defines the size of window and makes that window active.

ACTION

Opening a Window with the Window Command

To open another window, enter WINDOW with the parameters as defined above in Format. For example:

Type: F5 window 3,40,1,35,10 ↵

Result: This opens window 3 (if it was not already open) in the top right section of the screen — starting at column 40, line 1, with a width of 35 columns and length of 10 rows.

NOTE #1 **Setting the Windows at Startup.** By adding a WINDOW command as a line in your STARTUP.INT file, you can have XyWrite automatically set up your windows when it loads.

NOTE #2 **Window Size.** It is interesting to note that the parameters corresponding to a *full* display are: WINDOW 2,0,0,80,22

FORMAT	CM **RMVSCR**	Remove Screen
ABBREV	CM **RS**	
	This is an immediate command.	

PURPOSE

RS (Remove Screen) closes the window that the cursor is located in and returns the display to the previously displayed window. (This command was formerly called Reset Screen.) In general, we use the terms *window* and *screen* interchangeably.

ACTION

Closing a Window.
To close a window:

1. Move the cursor to the window you want to close (see Note #1 below). To do this:

   Press: Shift F10

2. Clear the window you want to close of any document:

   Type: F5 store ↵ or F5 abort ↵

3. Reset the window:

   Type: F5 rmvscr ↵

Result: The window closes and the display returns to the previously displayed windows (if any).

NOTE #1

Identifying the Active Document. In the case of a split screen, there are two ways to tell which window is currently active:

- The filename of the active document appears at the top of the screen.
- The cursor is located in the active window. (If necessary, press F10 to move cursor off Command Line and into the window.)

NOTE #2

Automatic Window Closing. If you set the New Window setting to 1 (DEFAULT NW=1), then the active window is automatically closed whenever the displayed file is aborted.

NOTES

INTRO

XyWrite's Spelling Checker program "proofreads" your documents for you, saving you from the embarrassment of distributing material with typographical errors. You can check the spelling of a word, a defined block, a file, or a series of files. You can even have the Spelling Checker correct your errors automatically, and you can extend the Auto-Correct feature to give you a very powerful "shorthand" phrase expansion. This section describes the procedures for running and modifying the Spelling Checker program.

Two additional utilities are provided for word counting and for sorting lines of text. You can count either forward or backward from your position in the file. You can sort lines of text alphabetically within a block or within a whole file.

A thesaurus is also available. For help in finding just the right word, you can quickly reference words that have a similar meaning. XyWrite gives you word power as well as powerful word processing!

CONTENTS

PURPOSE

The Spelling Checker compares the words in your document with a standard dictionary and any personal dictionaries you have loaded into memory. If it does not find a match, the Spelling Checker notifies you that it has found a questionable word. Since all XyWrite dictionaries are memory resident, checking is done quickly.

There are three ways to *check* words with the Spelling Checker:

- **Auto-Check.** Check each word automatically as you type it in. A beep notifies you that a word is questionable.
- **Word Check.** Select a word with the cursor and check it with a single keystroke.
- **File Check.** Check a block of defined text, a file, or a list of files with the SPELL command.

There are three ways to *correct* words with the Spelling Checker:

- **Auto-Correct.** Correct your misspellings as you type with the automatic replacement option. (You can also use this feature to expand an abbreviation into the word or phrase it represents.)
- **Word Correct.** Correct by selecting from a list of alternates chosen from the dictionary and given in the spelling menu.
- **File Correct.** Correct by supplying the alternates to the list of questionable words found by the SPELL command. You update the file(s) automatically in one pass by giving the CORRECT command.

Before using these features, you need to understand something about the different dictionaries and the menus that the XyWrite Spelling Checker uses.

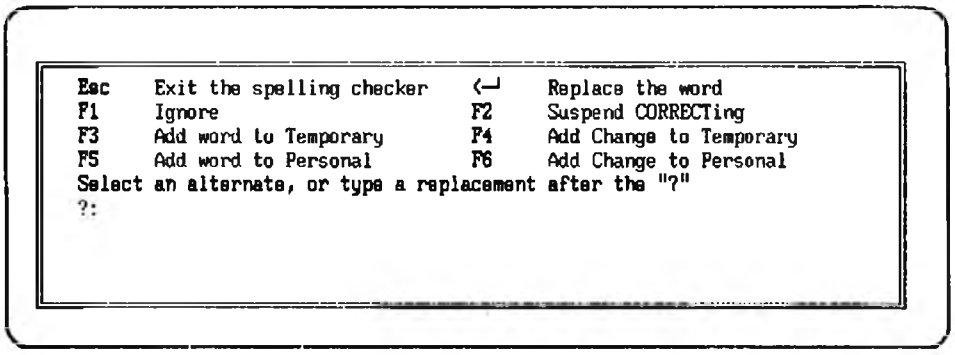
The Dictionaries. The three types of dictionaries associated with the Spelling Checker program are: the *main* dictionary (DICT.SPL), the *personal* dictionaries, and the *temporary* dictionary. DICT.SPL is a large (approximately 100,000 words), built-in dictionary included on Supplemental Disk 1. It is in binary format and cannot be directly edited.

The optional personal dictionaries contain specialized words that you use often. They include such things as proper names, product names, technical terms relating to your line of work, etc. You can have several different personal dictionaries. Your program disk contains some examples: LEGAL.SPL, which contains common Latin and other legal terms; BUSINESS.SPL, which contains standard business and political terms; and PERS.SPL, which contains general supplemental words as well as some very common misspellings and their corrections to support automatic correction of your work.

The temporary dictionary, which is created as you use the Spelling Checker, contains words that you do not wish to save. Because they are stored only temporarily, these words are ones you expect to use at your current editing session only. You enter words into this dictionary by using the main spelling menu, which is described below.

The Menus. The main spelling menu appears whenever the program encounters a questionable word – one it cannot find in any of the current dictionaries. The menu may appear on the top or bottom half of the the screen, depending on where the questionable word appears in the file. Because the menu uses only half of the screen, you can see the questionable word in context on the other half of the screen.

The menu looks like this:



The menu shows you three pieces of information:

- The *actions* you can take in response to what the Spelling Checker has found. This is the first part of the menu you see.

- The *questionable word,* which is preceded by a question mark.

- A list of *alternate words* – suggested replacements for the word in question. The message "Searching" appears until the first alternate is found. When the list is complete, the first alternate is highlighted.

  Sometimes the Spelling Checker encounters a word for which it has no alternate spelling. When this happens, the message "No alternates" appears in the menu.

The Spelling Checker also uses a second menu, which appears whenever you add a word that contains one or more capitalized letters to the dictionary. It asks you to verify that the capitalization you are using is the standard.

You will learn how to use each of these menus in the following procedure.

ACTION **Using the Spelling Checker.**
To check spelling in an existing document:

1. Call an existing file to the screen.

2. Enter the SPELL command.

   Type: F5 spell ↵

   Result: XyWrite checks to see if the main spelling dictionary, DICT.SPL, is loaded into memory. If it is not, XyWrite automatically loads it. The cursor moves to the first questionable word in your file and displays the spelling menu. You now select the option you want from the menu.

   - **Exit from the Spelling Checker.** Turn off the Spelling Checker program by pressing Esc. Select this option if you are using the SPELL command to check the spelling in a long file and decide to stop and store your corrections to disk before finishing.

   - **Replace Word.** When you press ↵, the selected word in the menu replaces the word under the text cursor. The word in the menu could be one of the alternate words listed by the Spelling Checker or it could be a replacement word that you typed into the menu. When you press ↵, neither the temporary nor personal dictionary is affected.

   - **Skip Questionable Word.** Press F1 to tell the Spelling Checker program to ignore the questionable word. Select this option if the word is correctly spelled but is not a word that you expect to use again - perhaps the name of an obscure author you are citing.

- **Edit While Correcting.** Press F2 to return to the text file to make an edit. This is similar to Esc, but XyWrite leaves the SPELL command in the header so you can restart the checking process by pressing F9. This option is useful if you notice a grammatical error when the Spelling Checker stops in your file. (For more information on this option, see "File Correct.")

- **Add Word to Temporary Dictionary.** Press F3 to add the questionable word to the temporary dictionary. Make this selection if the word in question is spelled correctly in your file but is not a word that you use regularly.

- **Add Word to Personal Dictionary.** Press F5 to add the questionable word to your *personal* dictionary. Make this selection if the word in question is spelled correctly in your file *and* is a word that your use regularly (like your company's name or products).

If none of the alternates presented is appropriate, you can type in your own replacement word after the question mark in the menu. To enter that replacement word into your file as an alternate spelling for the questionable word *and* into one of the dictionaries, use one of the two remaining options.

- **Add Change to Temporary Dictionary.** Press F4 to replace the questionable word with the replacement word you typed into the menu *and* add the replacement word to the temporary dictionary as an alternate spelling.

- **Add Change to Personal Dictionary.** Press F6 to replace the questionable word with the replacement word you typed into the menu *and* add the replacement word to the personal dictionary as an alternate spelling.

3. Let's assume the Spelling Checker stopped on the word "Pronut," a product name you are using frequently today but do not expect to refer to again. Add the word to the temporary dictionary.

   Press: F4

   Result: A new menu appears on the screen, asking you to verify the capitalization of the word. This menu appears whenever you tell XyWrite to add a word that contains uppercase letters to one of the dictionaries.

4. "Pronut" is a proper name, so enter it in the dictionary with the first letter capitalized.

   Press: F2

   Result: The Spelling Checker moves to the next questionable word in the file. It will no longer stop on the word "Pronut" unless it appears in all lowercase.

5. Select the appropriate option from the spelling menu.

6. Repeat step 5 until the Spelling Checker reaches the end of the file.

NOTE #1 **Memory Requirements.** DICT.SPL and its associated program SPELL.OVR require a minimum of 135K of memory. This is in addition to the memory required to run XyWrite with DOS.

NOTE #2 **Multiple Dictionaries.** If you have more than one personal dictionary loaded into memory, F5 and F6 add words only to the first personal dictionary you loaded.

NOTE #3 **Upper- and Lowercase.** When you use the spelling menu to add a word containing one or more uppercase letters to the dictionary, the Spelling Checker displays a second menu that asks you to verify the capitalization of the word. For information on how the Spelling Checker handles upper- and lowercase entries in the dictionary, see "Editing a Personal Dictionary."

NOTE #4 **Duplicate Alternates.** Sometimes, the Spelling Checker lists the same alternate word twice. That's because the Spelling Checker searches the dictionaries twice: once for words that *sound like* the questionable word and once for words that are *spelled like* the questionable word. If an alternate word satisfies both conditions, it appears twice.

NOTE #5 **Numbers and Punctuation.** The Spelling Checker program ignores punctuation marks, embedded commands, and symbols. It also ignores numbers when they are the only characters in a word (e.g., 1987). If you want the Spelling Checker to ignore numbers when they start a word (e.g., 12th, 1920s), enter the command DEFAULT CK=1 in the printer file, startup file, or on the command line. (The initial setting is CK=0.)

NOTE #6 **Reassigning Keys.** If you want to assign the eight actions associated with the Spelling Checker menu to keys other than F1 through F6, Esc, and ↵, you can use the function calls Q1 through Q8, respectively. Changing the keyboard table definitions does not change the menu display, nor does it affect the operation of the keys when in normal text operation.

To reassign the Spelling Checker functions, call your keyboard file and insert the function calls Q1 through Q8, followed by a comma, at the beginning of the key code definitions you want to change.

For example, suppose you want to move the Spelling Checker function "Edit in context" from F2 to the 2 key on the number keys along the top of the keyboard. The standard IBM PC keyboard definition for 2 is 3=2. To add the Spelling Checker function to that key, change the definition to 3=Q2,2.

Remember that every time you change the keyboard file, you must use the LDKBD command to load the changes into memory.

NOTE #7 **Spell Checking Footnotes.** The SPELL command does NOT check the spelling of words in footnotes when text is displayed in the Normal mode. Either go to Expanded display or give the SPELL command and the filename (File Check).

ACTION

Editing a Personal Dictionary.

You can use the personal dictionaries just as they are provided on the program disks, you can add words to them using the spelling menu, or, since they are just text files, you can edit them directly. You can also create your own personal dictionary to handle the spelling of names, cities, and other words that you use every day.

1. Select a name for your dictionary. Let's call it MINE.SPL.

2. Enter the NEW command, just as you would for any text file.

   Type: F5 new mine.spl ↵

   Result: The new (empty) file called MINE.SPL appears on the screen.

3. On the first line, enter the label that tells XyWrite that this file is a personal spelling dictionary file.

   Type: ;SP;←

   Be sure to type the label in uppercase letters.

4. Enter the information you want into the dictionary.

   a. **Adding Words.** Enter the first word you want to store in the dictionary. If the word is a proper name, use the correct combination of upper- and lowercase letters (see Note #7). At the end of the word, press ↵. For example, enter a name:

      Type: Babar←

      Result: Once you load the dictionary, you can type the name Babar without having the Spelling Checker program flag it as an error. Repeat this step for every word that you want to enter manually into your personal dictionary.

b. **Setting Up Automatic Replacement.** If there are words that you frequently misspell in a certain way, you can enter those misspellings along with their corrections into a personal dictionary. Then, whenever you are using Auto-Check/Correct or Auto-Replace, the Spelling Checker corrects the word for you. When you use the File Check or Word Check functions, the Spelling Checker lists the correction you enter in the dictionary as the first alternate spelling.

For example, let's say that you have trouble with the word "receive," and often type it as "recieve." To have Auto-Check/Correct or Auto-Replace automatically correct it for you:

Type: `recieve receive←`

Result: When you run Auto-Check/Correct or Auto-Replace, it automatically replaces the misspelled version of the word with the alternate spelling you specify. When you run File Check or Word Check, "receive" is the first alternate spelling of "recieve" in the spelling menu.

You can also use the Spelling Checker's automatic replacement feature to create your own shorthand typing. Type the following line into your personal dictionary.

Type: `xyw XyWrite←`

Result: When you have either Auto-Check/Correct or Auto-Replace on and type "xyw" into your text file, the Spelling Checker automatically changes it to "XyWrite." You can use the same method to change one word into a phrase. For example, type the following line into your personal dictionary:

Type: `p1 party of the first part←`

Every time you type the word "p1," Auto-Check/Correct or Auto-Replace substitutes the phrase "party of the first part."

If you want to use a multi-line phrase as the replacement, end each line with an ASCII 13 (♪).

c. **Ignoring Entries in DICT.SPL.** Occasionally, you may want to have the Spelling Checker ignore a word that is in DICT.SPL. For example, you may frequently write about a company that use a non-standard spelling of a common word, such as WITTE. It is easy to misspell this word by typing WITTY. If you want the Spelling Checker to let you know that WITTY may be a misspelling of the company name, you can tell the program to ignore the entry in DICT.SPL. Enter the standard spelling followed by a space, a question mark, and a carriage return. For example:

Type: `witty ?←`

Result: When you type the word "witty," the Spelling Checker program will flag it as a misspelled word.

The personal dictionary MINE.SPL now looks like this:

```
;SP;←
Babar←
recieve receive←
p1 party of the first part←
witty ?←
```

5. Load the dictionary into memory (see "Loading the Dictionary" in this section).

NOTE #7 **Upper- and Lowercase**. In addition to verifying the spelling of a word, XyWrite's Spelling Checker also checks that you have used the right combination of upper- and lowercase letters, according to the way you enter words into the dictionaries. The following rules apply:

- If the word in the dictionary is all lowercase, the Spelling Checker will accept any of the following combinations of the word as proper spellings: all lowercase, all uppercase, or with the first letter capitalized. Any other combinations will be flagged as questionable. For example, *startup*, *STARTUP*, *and Startup* are all acceptable, but if you type StartUp, the Spelling Checker will beep.

- If the word in the dictionary is all uppercase, you must always type it in all uppercase to avoid getting beeped.

- If the word in the dictionary is a combination of upper- and lowercase, you must type that word exactly as it appears in the dictionary or in all uppercase. Any other combination is beeped.

ACTION **Loading the Dictionary.**
The Spelling Checker program uses dictionaries loaded in memory to verify the spelling of your document. When you use the File Check, Word Check, or Auto-Check/Correct functions, XyWrite automatically loads DICT.SPL if it is not already in memory. If you want the Spelling Checker to also be aware of the specialized words you use, use the LOAD command to load one or more personal dictionaries.

Let's load PERS.SPL and LEGAL.SPL.

Type: F5 `load pers.spl+legal.spl` ↵

Result: Both the personal dictionaries are loaded into memory from this single command. You can now use any of the Spelling Checker options. For more information about the LOAD command, refer to "Loading Customization Files" in Chapter 5.

NOTE #8 **Location**. To load DICT.SPL, the Spelling Checker program requires access to the program SPELL.OVR and to DICT.SPL. To find them, the program looks first in the current subdirectory; if the required files are not there, it searches the subdirectories defined in the DOS *path* statement.

FORMAT [Ctrl] S — Spell Check Single Word

This is an immediate command.

PURPOSE

[Ctrl] S gives you a quick way to check the spelling of a single word. When you see a word that doesn't look quite right, [Ctrl] S lets you spot check and, if necessary, correct it. After you check the word, the cursor moves to the start of the next word, so you can quickly continue checking additional words if you want.

[Ctrl] S checks to see if DICT.SPL is loaded into memory. If it is not, it automatically loads it for you.

ACTION

Checking the Spelling of a Single Word.
To check the spelling of a single word:

1. Move the cursor to the word in question.

2. Press: [Ctrl] S

Result: If the word is correctly spelled, the message OK appears in the PRMPT field and the cursor moves to the next word in the file. Repeat step 2 if you wish. If Word Check does not find the word in one of the spelling dictionaries in memory, it displays the spelling menu. For an explanation of the menu, see "Using the Spelling Checker" at the beginning of this section.

NOTE

Reassigning Word Check. You can reassign this function to another key in your keyboard file. (See "Keyboard File" in Chapter 6.) The function call for Word Check is SO (Spell One Word).

FORMAT

CM **SPELL** (Option 1)
CM **SPELL** *filename,targetfile* (Option 2)
CM **SPELL** *@parentfile,targetfile* (Option 3)

filename is the name of the file being searched.
targetfile (optional) is the name of the file where you want questionable words stored. If you omit this name, XyWrite assigns the name SPELL.TMP.
parentfile contains the names of the files to be searched.

These are immediate commands.

PURPOSE

The **SPELL** command lets you search existing text for words with questionable spelling (i.e., words that do not appear in one of the dictionaries that you loaded into memory). SPELL also automatically loads DICT.SPL into memory if it is not already there.

There are four ways you can use the SPELL command:

- On a defined block
- On a displayed file
- On a single stored file
- On a list of stored files

If you use SPELL on a defined block or a displayed file, it displays the spelling menu when it finds a questionable word. If this word is in the temporary or a personal dictionary with a replacement spelling, the replacement is listed as the first alternate. For example, if you have the entry "untill until" in your personal dictionary, "until" becomes the first alternate in the list that the Spelling Checker displays.

Additionally, this use of the SPELL command gives a count of the total number of words in the file and the total number of "bad" (questionable) words found.

The SPELL command can also search one or more stored files and make a list of questionable words in a separate file. You might use this option if you have a long document to search and want to make some phone calls while the search is taking place. When SPELL finishes, you can review the list at your convenience and make corrections. The CORRECT command can then be used to update your file(s). (See "File Correct.")

ACTION
(Option 1)

Using the SPELL Command with a Displayed File.
To check a displayed file (or a defined block) for questionably spelled words:

1. Move the cursor to the point in the file where you want the check to begin (or define the block of text you want to check).

2. If you want the check to include the text in running headers, footers, footnotes, and index entries, switch to expanded mode.

   Press: Ctrl F9

3. Enter the SPELL command.

   Type: F5 spell ↵

Result: If there are no questionable words, the cursor moves to the end of the file or defined block, and the message "*n* words, 0 bad" appears on the PRMPT line (*n* is the total number of words in the file or defined block). If there is a questionable word, the cursor moves to that word and the spelling menu appears on the screen. When the Spelling Checker finds a replacement spelling for the questionable word in the personal or temporary dictionary, the program displays the replacement spelling as the first alternate word in the menu. (You can add replacement words "on the fly" by typing them into the spelling menu.)

This process continues until the SPELL program reaches the end of the file or defined block. At that point, the message "*n* words, *p* bad" appears on the PRMPT line (*n* is the total number of words found during the search, and *p* is the number of questionable words found).

ACTION *(Option 2)*

Using the SPELL Command with a Stored File.
To check a file that is stored on disk and list the questionable words in a separate file:

Type: F5spell chapter.doc,spell.err↵

Result: The file CHAPTER.DOC is checked for spelling accuracy, and any questionable words are listed in the file SPELL.ERR. (If no target file is named, XyWrite assigns the name SPELL.TMP.) When the search is complete, the message "Done" appears on the PRMPT line. You can call the target file right away or wait until later to review SPELL's findings. (See "File Correct" later in this section.)

ACTION *(Option 3)*

Using the SPELL Command with Multiple Files.
To check a group of files for spelling accuracy:

1. Create the parent file.

   Type: F5ne chapters.all↵

2. List the names of all the files that you want XyWrite to search for words with questionable spellings. Type each filename on a line by itself.

   ```
   chapter1←
   chapter2←
   chapter3←
   .
   .
   .
   ```

3. Store the parent file and enter the SPELL command.

   Type: F5st↵

   Type: F5spell @chapters.all↵

   Result: XyWrite checks the words in each file listed in CHAPTERS.ALL and compiles a list of those words that it does not find in one of the spelling dictionaries. Because we did not specify a target file, the questionable words are stored in the file SPELL.TMP.

FORMAT

CM **CORRECT** *filename,string*

filename (optional) is the name of the file created by the SPELL command. (If the name is omitted, SPELL.TMP is assumed.)
string (optional) is text to be appended. (See Note #1.)

This is an immediate command.

PURPOSE

The **CORRECT** command is a companion command to the SPELL command. When you use the SPELL command to check the spelling in a stored file or files, the program creates a separate file that contains a list of questionable words. You can then use the CORRECT command and this list to fix the misspellings in your original document.

Before running the CORRECT command, you can edit the list of questionable words created by the SPELL command. If you provide replacement words for all of the questionable words, the CORRECT command will update your document by inserting the corrections in one pass.

If you are not sure about the spelling of a word when you are editing the list, you can leave it alone. When you run the CORRECT command, it will stop at that word and display the spelling menu. This lets you see the word in context and gives you a list of alternates at the same time.

If you know a word in the list is correct as is, then simply delete it from the list and it will be ignored by CORRECT.

CORRECT can also be used to append an optional string of text to all occurrences of the words listed in the correction file. You can use this option to help you create a *concordance* (an alphabetical index of keywords (see Note #1).

ACTION **Editing the Correction File.**

If you want to, you can edit the correction file created by the SPELL command before you run CORRECT.

1. Call the file created by the SPELL command. For example:

   Type: F5 ca spell.tmp ↵

   Result: The file appears on the screen. The first line in this file shows, in reverse mode, the name of the file SPELL reviewed for typographical errors. The following lines show, in normal mode, words in the named file that do not appear in one of the spelling dictionaries. If you asked SPELL to review more than one file, each filename is listed in reverse mode, followed by a list of questionable words. For example:

   ```
   CHAPTER1
   mispell
   occurance
   committment
   CHAPTER2
   recieve
   ```

2. Move the cursor to the first questionable word.

3. If the word is correctly spelled and you do not want to add it to a dictionary, delete it from the list. If it is incorrectly spelled and you know how to fix it, type a space and the correct spelling of the word. For example:

   ```
   mispell misspell
   ```

   If you are not sure of the spelling of a word, leave it alone.

4. Repeat steps 2 and 3 for each word on the list.

5. Store the file.

   Type: F5 st ↵

Result: The correction list is ready to use with the CORRECT command.

ACTION **Running the CORRECT Command.**

To run the CORRECT command, perform these steps.

1. Type: F5 correct ↵

   Result: The CORRECT command uses the information in SPELL.TMP to revise your document. (If your file has any other name, be sure to specify it when you type the CORRECT command.) If you deleted a word from SPELL.TMP, the CORRECT command skips over that word in your document. If you typed a replacement word for the questionable word, it inserts the replacement for you. For all other words in the list, CORRECT displays the spelling menu on one half of the screen and the part of the document that contains the questionable word on the other half of the screen.

2. Select the option you want from the spelling menu. For example, to add the word you originally typed to the temporary dictionary:

   Type: F3

   Result: That word is added to the temporary dictionary, so all other occurrences of it are ignored. If SPELL.TMP contains another word without a replacement, a new menu and a new part of your document appear on the screen. Repeat this step for each such word in the list.

3. When CORRECT reaches the end of your document, it displays the message "File corrected, save it? (Y/N)."

   To save the changes and store the document to disk:

   Press: Y

   To discard the changes:

   Press: N

NOTE #1

Automatic Index Creation. A special option of the CORRECT command allows you to compile a list of words and append a string to each occurrence of those words in your document. For example, you could compile a list of words you want to index and have the CORRECT command append an index marker every time it finds a word from your list in the document.

To use this function to create an index, you must create a new file (let's say its name is INDEX), enter (on the first line) the filename of the document in reverse mode followed by a carriage return in normal mode, and then enter the single-word index entries in normal mode, ending each entry with a carriage return. Store the file and then issue the command:

F5 correct index,«x1» ↵

Result: The CORRECT command goes into your document and appends an index marker to the words that you listed in the file "INDEX." When you extract the index with the IX command, all occurrences of these words will be included in the index. (For more information on indexes, see "Table of Contents and Index" in Chapter 5.)

NOTE #2

Translations. You can use the CORRECT command as part of the process of translating a document from one language to another. Simply create a correction file and enter the name of the document you want to translate in reverse mode on the first line, followed by a carriage return in normal mode. Then, still in normal mode, type in the word you want to translate, a space, and the word's equivalent in the other language. Repeat this step for each word you want to translate. For example:

```
CHAPTER.DOC
here ici
you vous
.
.
.
```

Only single-word word entries can be translated in this way, although you can replace the single words with a phrase.

FORMAT

Ctrl A — Turn Auto-Check/Correct On and Off
Ctrl F — Return and Fix Spelling Error

These are immediate commands.

PURPOSE

Auto-Check/Correct verifies your spelling as you type. If a replacement word has been given in one of the active personal dictionaries, it will automatically *correct* as well.

With Auto-Check/Correct ON, whenever you type a tab, space, or carriage return, XyWrite checks the spelling of the word immediately preceding the cursor. It looks in DICT.SPL, which it automatically loads into memory if it is not already there, and in any personal dictionaries that you loaded. If the word does not appear in one of these dictionaries, you will hear an *error beep.*

If the word appears in the personal or temporary dictionary with a replacement spelling, you will hear a *correction beep*, which is easily distinguishable from the error beep. The correction beep indicates that Auto-Check/Correct has automatically corrected the error.

When you hear the error beep, you have several choices:

- You can keep on typing to complete your train of thought, and then return to the word in question.
- You can stop immediately to correct the word by using either XyWrite's normal editing functions or the spelling menu.
- You can ignore the beep entirely. (You might choose this option if you have correctly typed a name or specialized term that does not appear in one of the dictionaries *and* if you don't wish to add it to your personal dictionary.)

ACTION **Using Auto-Check/Correct.**

To use Auto-Check/Correct to check your work as you type:

1. Turn on Auto-Check/Correct.

   Press: Ctrl A

   Result: The letter "c" appears in reverse mode at the top right corner of the screen. Auto-Check/Correct is now active.

2. Call your file to the screen or create a new one.

3. Type in the text of your document until you hear an error beep. Let's say you just typed the word *micro-justification* which does not appear in any of the dictionaries you loaded into memory, but which you use frequently. To avoid hearing a beep every time you type *microjustification*:

   Press: Ctrl F

   Result: The cursor moves to the questionable word and the spelling menu is displayed.

4. From the menu, choose the option you want. In this case, the word *microjustification* is correctly spelled, and you want to add it to your personal dictionary.

   Press: F5

   Result: The menu disappears and the cursor returns to your document, following the word *microjustification.* The word is now in your personal dictionary so you will no longer hear a beep when you type it.

5. Continue creating or editing your document. Each time Auto-Check/Correct corrects an error for you, you will hear the correction beep, and each time you type a word that does not appear in one of XyWrite's dictionaries, you will hear an error beep. Repeat step 3 whenever you want to see a list of alternate spellings or add a word to the dictionary.

6. When you are finished, you can store your file and call another one. Auto-Check/Correct remains on until you turn it off. To turn it off:

   Press: Ctrl A

NOTE #1 **Using Multiple Windows.** When you turn Auto-Check/Correct on, it is on in all nine windows and remains on until you cancel it.

NOTE #2 **Previous Error.** Pressing Ctrl F returns the cursor to the last word that Auto-Check/Correct detected as questionable. If you correct that error and then press Ctrl F again, the cursor will not move to another word. Auto-Check/Correct remembers only the one position.

NOTE #3 **Reassigning Keys.** You can reassign these functions to other keys in your keyboard file. (See "Keyboard File" in Chapter 6.) The function call for Auto-Check/Correct is AC, and the function call to move the cursor to the previous error is FS (Fix Spelling).

NOTE #4 **Automatic Correction.** For information on how to set up your personal dictionary to automatically correct your misspelled words or to expand abbreviations that you type into your file, refer to the section "Editing a Personal Dictionary."

NOTE #5 **Changing the Beeps.** You can change the tone of or even completely turn off the *error* and *correction* beeps by using the DEFAULT command to change the EB and CB settings, respectively. (See"Default Settings" in Chapter 6 for more information.)

NOTE #6 **Replacement without Correction.** You can use Auto-Replace to insert replacement words or phrases without invoking the Spelling Checker. (See the following description.)

FORMAT Ctrl R Turn Auto-Replace On and Off

This is an immediate command.

PURPOSE

Auto-Replace lets you use the automatic correction feature of the Spelling Checker without having to load the main spelling dictionary.

Whenever you type a tab, space, or carriage return, XyWrite checks the personal dictionaries in memory to see if the word you typed appears with a replacement word or phrase. If it is, XyWrite makes the replacement and beeps to indicate that a change has been made.

ACTION

Using Auto-Replace.
To use the Auto-Replace command:

1. Load the personal dictionaries that contain the replacement words and phrases you want inserted into your text. For example:

   Type: F5 `load pers.spl` ↵

2. Turn on Auto-Replace.

   Press: Ctrl R

   Result: A lowercase "r" appears in reverse mode at the top right corner of the screen. Auto-Replace is now active.

3. Call your file to the screen or create a new one.

4. Type in the text of your document.

   Result: Every time you hear a correction beep, it means XyWrite has deleted the word you just typed and inserted the replacement word or phrase associated with it in your personal dictionary.

5. When you are finished, you can store your file and call another one. Auto-Replace remains on until you turn it off. To turn it off:

   Press: Ctrl R

NOTE #1 **Using Multiple Windows.** When you turn Auto-Replace on, it is on in all nine windows and remains on until you cancel it.

NOTE #2 **Reassigning the Key.** You can reassign this function to another key in your keyboard file. (See "Keyboard File" in Chapter 6.) The function call for Auto-Replace is AR.

NOTE #3 **Building a Replacement Dictionary.** To help you build your replacement list, there is an IR (Insert Replacement) function call that can be assigned to a key on your keyboard. This function lets you add replacement words to the personal or temporary dictionary more easily as you type. IR displays the spelling menu with one keystroke so you can type in your replacement spelling and add it to the temporary or personal dictionary. (See "Using the Spelling Checker." See also "Keyboard File" in Chapter 6.)

NOTE #4 **Automatic Correction.** For information on how to set up your personal dictionary to automatically correct your misspelled words or to expand abbreviations that you type into your file, refer to the section "Editing a Personal Dictionary."

NOTE #5 **Audible Signals.** You can change the tone of or even completely turn off the correction beep by using the DEFAULT command to change the CB setting. Refer to "Default Settings" in Chapter 6 for more information.

Storing the Temporary Dictionary

FORMAT **CM STSPELL** *filename*

This is an immediate command.

PURPOSE The **STSPELL** command lets you save the words in the temporary dictionary in a file so you can reload it later. The temporary dictionary contains all of the words from the personal dictionaries that you loaded into memory, plus any words you added through the spelling menu. Unless you use STSPELL, the temporary dictionary is lost whenever you quit XyWrite or unload the Spelling Checker feature from memory.

ACTION **Storing the Temporary Dictionary.**
Suppose you are working on a document which contains many specialized words that you don't expect to use again. You added those words to the temporary dictionary as you developed the document, but now you have to quit XyWrite before finishing it. Rather than lose all those dictionary entries, you can store them in a file and reload them when you return to XyWrite.

1. Decide on a name for the temporary dictionary. Let's call it TEMP.SPL.

2. Enter the STSPELL command.

   Type: F5 `stspell temp.spl` ↵

   Result: The contents of the temporary dictionary are copied from memory into the file named TEMP.SPL. The temporary dictionary remains in memory as well until you quit XyWrite or unload the Spelling Checker.

3. When you are ready to reload the temporary dictionary:

   Type: F5 `load temp.spl` ↵

   Result: You can return to work on your document and use the Spelling Checker without having to reload the personal dictionary and without having to reenter the specialized words that you put in the temporary dictionary.

FORMAT [Ctrl] T Display a List of Synonyms

This is an immediate command.

PURPOSE

XyWrite's thesaurus displays a list of synonyms. To use it, you just put the cursor on the word for which you want a synonym and press [Ctrl] T. XyWrite looks in its thesaurus for words with a similar meaning. When it finds them, it displays a list organized by parts of speech and by meaning. You can review the list and take one of three actions.

- Select one of the synonyms listed and insert it in your document ([↵]) with the same punctuation and capitalization and in the same display mode as the original word
- Leave your original word intact ([Esc])
- Build a new list of synonyms based on one of the words in the original list ([Ctrl] [PgDn])

To use the thesaurus, you must have the files WORD.OVR and WFBG.SYN either in the current subdirectory or in one of the subdirectories defined in your PATH statement. WORD.OVR is a program file that XyWrite needs to execute the thesaurus function and WFBG.SYN is the dictionary of synonyms. These files are not loaded into memory, so XyWrite goes to disk each time you use the thesaurus.

ACTION

Using the Thesaurus to Select a Synonym.

To use XyWrite's Thesaurus:

1. Make sure the files WORD.OVR and WFBG.SYN are in the current directory or in one of the subdirectories defined in your PATH statement.

2. Call your file to the screen.

3. Put the cursor on the word for which you want a synonym. As an example, let's find a synonym for the word "tool."

4. Press: Ctrl T

   Result: A menu appears, indicating XyWrite is looking for synonyms of the word you indicated. After a few seconds, that menu is replaced by:

```
tool:
noun · chessman, front, instrument, pawn, peon, puppet, stooge;
     · apparatus, appliance, contraption, contrivance, device, doodad,
       doohickey, gadget, gimmick, gizmo, implement, instrument, invention,
       machine, mechanism, thingumajig, utensil, widget.
verb · auger, bore, drill, pierce, ream.

     ↑↓↔:point      RTN:replace      ESC:exit      CTL-PGDN:look up
```

The first thing in the menu is the word you are looking up (in this case, "tool.") Next are all the synonyms for the word "tool" if you are using it as a noun. Notice that there are two sets of noun synonyms, which means there are two different definitions for the word "tool." After the nouns, there are several verb synonyms. (Depending on the word you are looking up, you may see a list of adverbs and adjectives in addition to - or instead of - the nouns and verbs.)

5. Let's replace the word "tool" with the word "utensil." Use the cursor arrow keys to select the word "utensil."

   Result: The word "utensil" is highlighted.

6. Press: [↵]

   Result: The word "utensil" appears in your file, replacing the word "tool," and the menu clears from the screen.

ACTION **Building Another Level of Synonyms.**

Suppose you are reviewing a list of synonyms and see a word that is close to being what you want. You can select that word and, instead of inserting it in your document file, have XyWrite build a new list of synonyms. Let's try it.

1. Repeat steps 1-4 of the above procedure.

2. Move the cursor to the word "peon."

3. Press: [Ctrl] [PgDn]

   Result: A new menu appears on the screen. This menu displays a list of synonyms for the word "peon." It also gives you a new option: if you press [Ctrl] [PgUp], XyWrite redisplays the previous menu.

4. Select the option you want from this menu. Let's exit from the thesaurus.

   Press: [Esc]

   Result: The menu disappears from the screen and your file is unchanged.

NOTE #1 **When There Are No Synonyms**. Occasionally, you may ask XyWrite to give you a synonym for a word that doesn't have one. When that happens, XyWrite tells you that the word was not found in the thesaurus, and gives you a list of words that immediately precede and follow it in the thesaurus dictionary. The cursor is on the word that is the closest in spelling to the word you requested a synonym for. You can request a list of synonyms for one of the words in the list or you can exit from the thesaurus.

NOTE #2 **Long Lists.** If the list of synonyms does not fit in one menu, you will see a message telling you to press PgDn to display the next part of the list.

NOTE #3 **Supplying a Word.** If you have the thesaurus menu on the screen and want to look up a word that does not appear in the displayed list, just type in the word you want to look up. As soon as you press the first character of the word, a window opens within the menu. Finish typing the word and press Ctrl PgDn to display a new list of synonyms or Esc to close the window.

NOTE #4 **Reassigning the Key.** You can assign this function to another key in your keyboard file. (See "Keyboard File" in Chapter 6.) The function call for the thesaurus is SY.

NOTE #5 **Word Forms.** The thesaurus contains only the root form of many words. For example, when you look up the word "dictionaries," XyWrite displays a list of synonyms for the word "dictionary." If you press ↵ to replace the word "dictionaries" with a synomym from the list, you have to edit the replacement to make it plural. You have to make similar adjustments for different tenses and parts of speech.

FORMAT

CM **WC** Word Count
CM **WCB** Word Count Backward

These are immediate commands.

PURPOSE

The **WC** and **WCB** commands count the number of words in your file or the number of words in a defined block. They then display the exact number of words in the PRMPT field of the header.

A word is a string of characters followed by a word separator (space, comma, colon, semicolon, period, question mark, exclamation point, tab, or carriage return).

ACTION

Counting Words from the Current Cursor Position to End.

To count the number of words in a displayed file (or a defined block):

1. Move the cursor to the point in the file where you want the count to begin (or define the block of text you want to search).

2. If you want the Word Count to include the text in running headers and footers, footnotes, and index markers, switch to expanded mode.

   Press: Ctrl F9

3. Enter the Word Count command.

   Type: F5 wc ↵

   Result: XyWrite counts the number of words from the current cursor position to the end of the file (or in the displayed block) and displays the total number in the PRMPT field.

ACTION

Counting Words from the Current Cursor Position to Start of File.

You can determine exactly how many words you have written up to a certain point in your file by counting the number of words from the current cursor position to the beginning of the file:

1. Be sure the cursor is at the point where you want the count to begin.

2. If you want the Word Count to include the text in running headers and footers, footnotes, and index markers, switch to expanded mode.

   Press: Ctrl F9

3. Enter the Word Count Backward command.

   Type: F5 wcb ↵

Result: XyWrite counts the number of words from the current cursor position to the beginning of the file (or in the displayed block) and displays the total number in the PRMPT field.

NOTE #1

VA Settings. You can obtain the current value of the total number of words counted by using the VA command with $WC. If you place a VA $WC format command in your file, it will read and print out the last value calculated by the WC or SPELL commands. You can use this value in keystroke programs as well. (See "User Programming" in Chapter 5.)

FORMAT

CM **SORT** *filename,targetfile* (Option 1)
CM **SORT** (Option 2)

filename is the file you want to sort.
targetfile is the file where sorted text is stored.

This is an immediate command.

PURPOSE

The **SORT** command allows you to rearrange the entries in a file or in a defined block into alphabetical order. (An *entry* can be a single character or a group of words. Each entry ends with a carriage return.)

ACTION
(Option 1)

Sorting a File.
To sort the entries in a file, say PERS.SPL, into alphabetical order:

1. Call the file to the screen and check that each entry ends with a carriage return.

2. Store the file and enter the SORT command.

   Type: F5 sort pers.spl,pers.srt ↵

Result: A new file called PERS.SRT contains entries from PERS.SPL rearranged into alphabetical order. The file PERS.SPL remains unchanged.

ACTION
(Option 2)

Sorting a Defined Block.
To sort the entries in a defined block:

1. Call the file to the screen and define the section you want to sort into alphabetical order.

2. Enter the SORT command.

   Type: F5 sort ↵

Result: The defined block disappears from the screen for an instant. When it reappears, the entries in the defined block are in alphabetical order.

NOTE

Default Sort Table. XyWrite has a built-in set of sorting rules for arranging entries. For example, XyWrite treats the upper- and lowercase versions of letters as equal. Perhaps you want uppercase entries to be sorted before lowercase entries. You can do that by creating your own sort table. (See "Sort File" in Chapter 6 for more information.)

INTRO

Formatting is the manner in which text is arranged on a page. Common examples include margins, indents, justification, use of running headers, footers, footnotes, and character modes. The power of XyWrite is that all of these controls are embedded as hidden characters in the text (and abbreviated on the display), so that you can go back and change them at will.

With XyWrite's system of embedded commands, you have tremendous versatility at revising text. For example, you can change a single indent command to alter the way all paragraphs are indented. This versatility extends to every formatting command represented by a triangle (▲) on the display — for margins, tabs, indents, page length, and so on.

The formatting commands are listed in the table of contents on the following pages. The chart which then follows shows graphically how the commands in this section relate to each other.

CONTENTS

This chapter is divided into eleven major sections arranged alphabetically. Each section stands on its own, making it easy for you to read only the sections of interest. Thus, if you are interested in how to control the margins and tabs, you would do well to read the Page Width section from start to finish.

The Table of Contents appears on the next two pages.

INTRO

Formatting is the manner in which text is arranged on a page. Common examples include margins, indents, justification, use of running headers, footers, footnotes, and character modes. The power of XyWrite is that all of these controls are embedded as hidden characters in the text (and abbreviated on the display), so that you can go back and change them at will.

With XyWrite's system of embedded commands, you have tremendous versatility at revising text. For example, you can change a single indent command to alter the way all paragraphs are indented. This versatility extends to every formatting command represented by a triangle (▲) on the display — for margins, tabs, indents, page length, and so on.

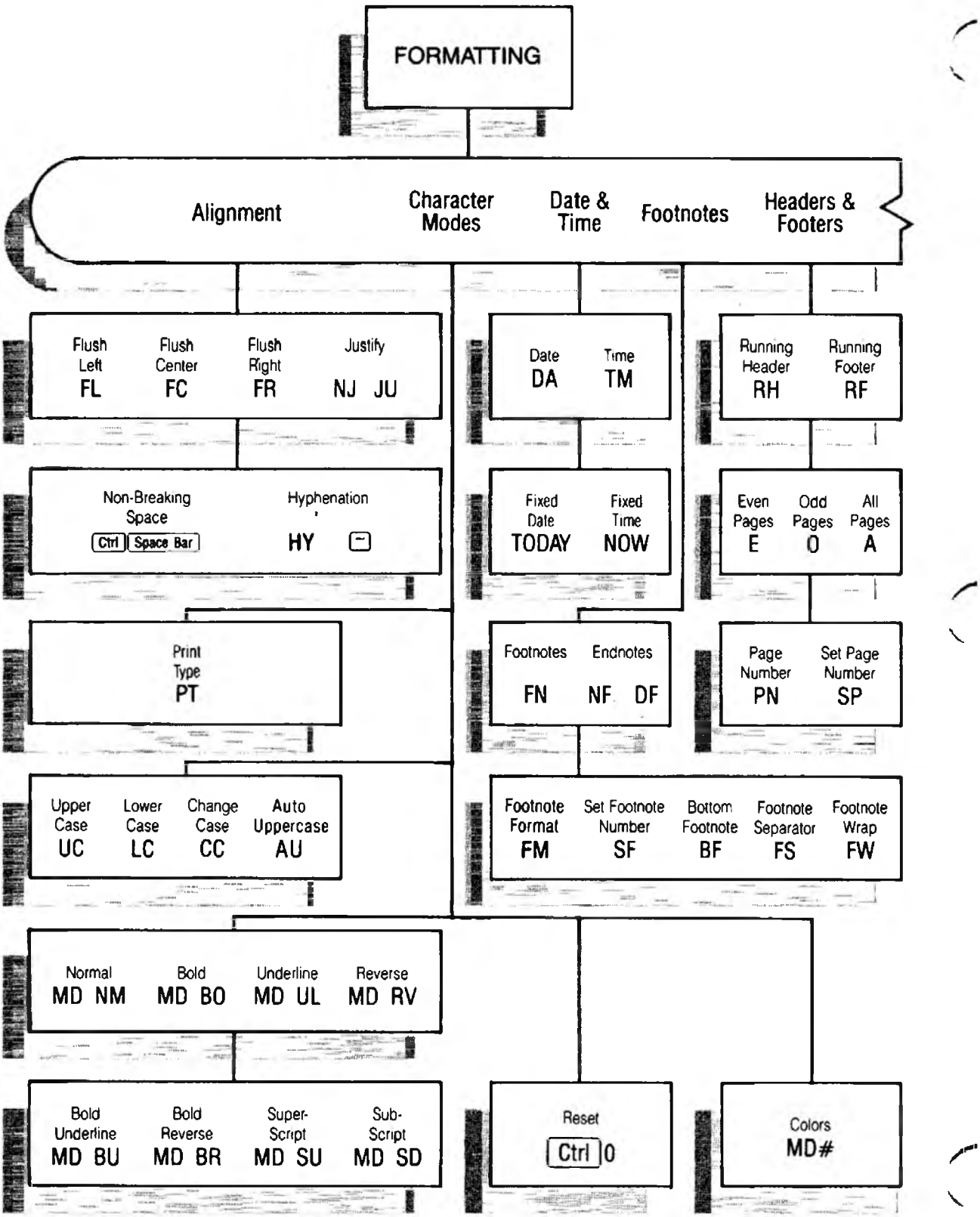
The formatting commands are listed in the table of contents on the following pages. The chart which then follows shows graphically how the commands in this section relate to each other.

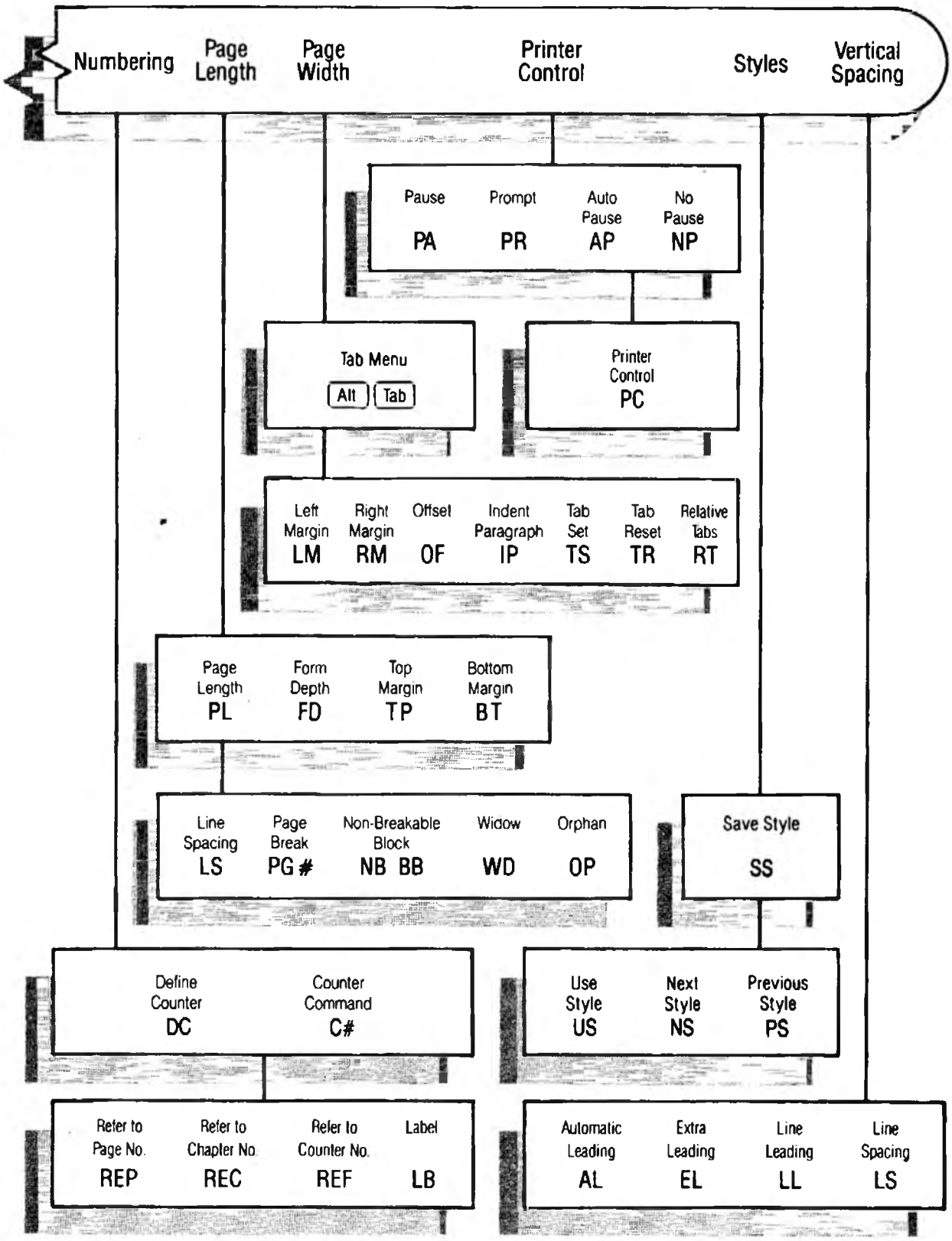
CONTENTS

This chapter is divided into eleven major sections arranged alphabetically. Each section stands on its own, making it easy for you to read only the sections of interest. Thus, if you are interested in how to control the margins and tabs, you would do well to read the Page Width section from start to finish.

The Table of Contents appears on the next two pages.

CONTENTS


FORMATTING
Alignment
Character Modes
Date & Time
Footnotes
Headers & Footers
Flush Left FL
Flush Center FC
Flush Right FR
Justify NJ JU
Non-Breaking Space Ctrl Space Bar
Hyphenation HY -
Print Type PT
Upper Case UC
Lower Case LC
Change Case CC
Auto Uppercase AU
Normal MD NM
Bold MD BO
Underline MD UL
Reverse MD RV
Bold Underline MD BU
Bold Reverse MD BR
Super-Script MD SU
Sub-Script MD SD
Date DA
Time TM
Fixed Date TODAY
Fixed Time NOW
Footnotes FN
Endnotes NF DF
Footnote Format FM
Set Footnote Number SF
Bottom Footnote BF
Footnote Separator FS
Footnote Wrap FW
Reset Ctrl 0
Running Header RH
Running Footer RF
Even Pages E
Odd Pages O
All Pages A
Page Number PN
Set Page Number SP
Colors MD#


Numbering
Page Length
Page Width
Printer Control
Styles
Vertical Spacing
Pause PA
Prompt PR
Auto Pause AP
No Pause NP
Tab Menu
Alt Tab
Printer Control PC
Left Margin LM
Right Margin RM
Offset OF
Indent Paragraph IP
Tab Set TS
Tab Reset TR
Relative Tabs RT
Page Length PL
Form Depth FD
Top Margin TP
Bottom Margin BT
Line Spacing LS
Page Break PG #
Non-Breakable Block NB BB
Widow WD
Orphan OP
Save Style SS
Define Counter DC
Counter Command C#
Use Style US
Next Style NS
Previous Style PS
Refer to Page No. REP
Refer to Chapter No. REC
Refer to Counter No. REF
Label LB
Automatic Leading AL
Extra Leading EL
Line Leading LL
Line Spacing LS

Embedded Commands. An embedded command is a command which is inserted into the text. It is normally hidden so as not to interfere with the text. Examples are:

Mode Bold	«MDBO»
Right Margin	«RM70»
Tab Set	«TS5,10,15»

Generally, embedded commands affect the *format* of a document and not its content. Although hidden, an embedded command is actually present in the file — for example, you could search for «MDBO». In Expanded Display you can edit embedded commands.

Characteristics. Let's cover some of the characteristics common to all embedded commands.

1. **How Embedded Commands Are Normally Displayed.** Embedded commands appear in the Normal Display either as a triangle, as a character mode (bold, underline, reverse or a combination), or as a bold footnote number.

   Moreover, if the cursor is placed on the triangle, the command itself appears on the PRMPT (Prompt) line of the header.

2. **Where Embedded Commands Take Effect.** Embedded commands take effect starting at the point they are placed in the text and *continuing* until the end of the document or until they are overridden (whichever occurs first). An embedded command never affects any text located on any lines preceding the command.

   For example, FC centers text from its triangle to the end of the document — or until a flush left, flush right, or justify command is encountered.

   The following commands have a repeated effect only if placed in a running header or footer: TM (Time), DA (Date) and PN (Page Number).

3. **How Embedded Commands Can Be Expanded for Viewing.** Embedded commands can be viewed by pressing Ctrl F9 to switch to the Expanded Display. They can be edited in this mode just like any other text.

4. **How Embedded Commands Can Be Edited.** You can switch to Expanded Display to edit commands, or in Normal Display simply delete and re-enter them. You can also define and copy or move them as you would any text.

5. **How Embedded Commands Affect Printing.** While an embedded triangle occupies a space on the screen, it occupies no such space on the printout. Instead, it is stripped out, and its effect is printed (FC would center the text).

   For example, the first of the following two lines shows how an embedded triangle appears on the screen, while the second line shows how that line is printed — the triangle is removed and the sentence is shifted to the left.

   ▲Embedded triangles are displayed but not printed.
   Embedded triangles are displayed but not printed.

6. **Hiding Embedded Commands from View.** If you prefer not to have the embedded commands visible on the screen, you can hide them with the NM (No Markers) command. Go to the command line and type:

   F5 func nm ↵

   The markers re-appear upon striking Ctrl F9 twice (to switch to Expanded Display and back). You can customize XyWrite to display files without markers by including the DT=3 setting in your Printer File. See Printer File in Chapter 6, Customizing.

PURPOSE

Normal Display and Expanded Display are two different ways to view a document, as shown in the illustration on the following page.

- **Normal Display** shows the embedded commands as triangles, keeping the display uncluttered. Text is shown as it will be printed. As the name suggests, this is the mode you normally use.
- **Expanded Display** reveals all embedded commands within double-angle brackets — that is, they are expanded for viewing. You can then move the cursor into the embedded commands and change them. Because the embedded commands may be lengthy, line endings are not shown as they will be printed.

ACTION

Toggling Between Normal and Expanded Display.
To switch from Normal to Expanded Display:

Press: Ctrl F9

To switch back to Normal Display, perform the same action again. This action toggles between the two modes.

NOTE #1

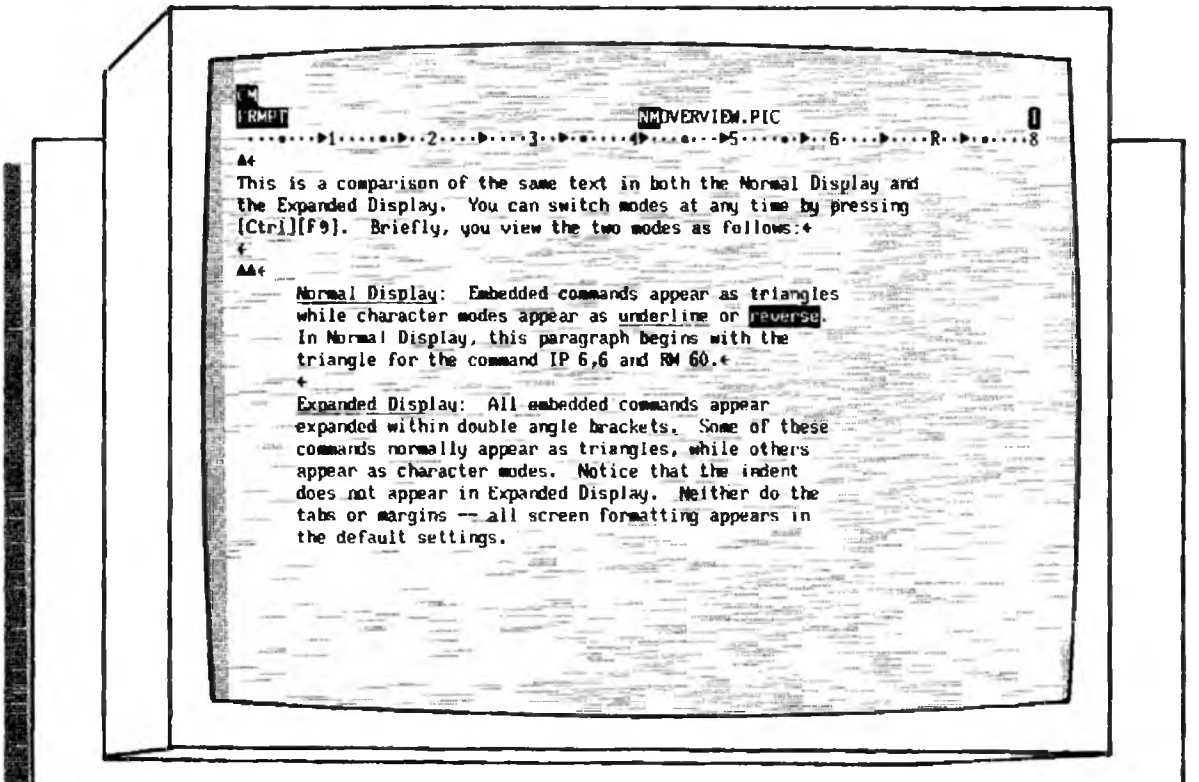
P-L Indicator. The Page-Line indicator (P-L) at the top of your screen turns off when you are in the Expanded Display. P-L operates only in the Normal Display. (Consequently, if you wish to turn off the P-L indicator, press Ctrl F9 twice.)

NOTE #2

Default Ruler Settings. In Expanded Display the *default* tabs are used in place of the document's own tab stops. In order to make the Expanded Display appear more like the Normal Display, use the DEFAULT command to set the tab stops. For example, if your document uses tab settings TS 3,6,9 then type:

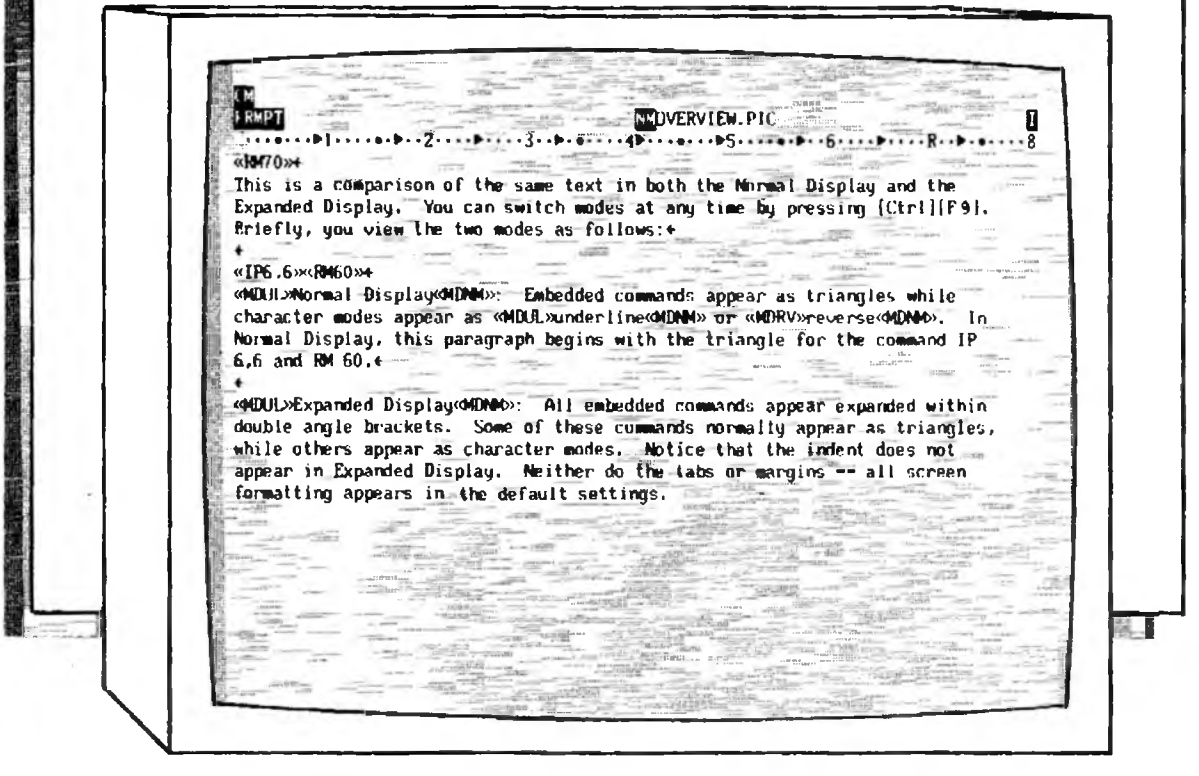
```
CM default ts=3,6,9
```

Now whenever you switch to Expanded Display, these tab settings will be in effect.



NORMAL DISPLAY

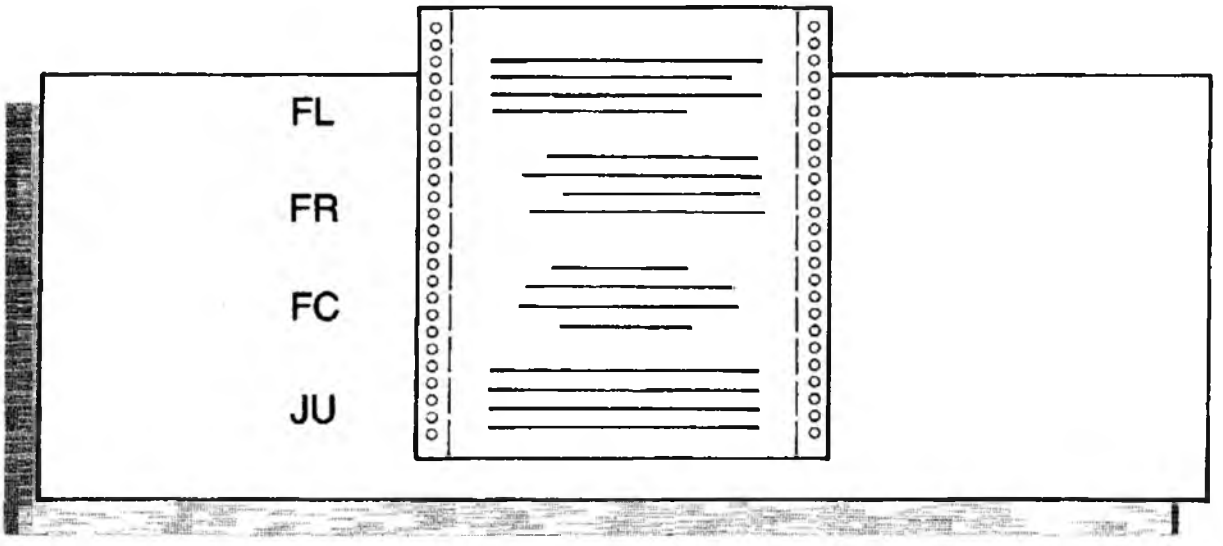
EXPANDED DISPLAY



INTRO

You may want to modify the appearance of text by changing its alignment. The four alignments available are shown in the figure below. This section covers these alignments plus two related topics — Non-Breaking Space and Hyphenation. Microjustification is described in Chapter 6.

CONTENTS



FORMAT		
	CM FL	Flush Left
	CM FC	Flush Center
	CM FR	Flush Right

PURPOSE

These three commands control the horizontal alignment of text. Each command is entered at a point in text and takes effect from that line forward. You can freely switch from one alignment to another simply by entering the command for the new alignment.

- Flush Left (FL) pushes the lines of text against the left margin (LM). This is the most common alignment.
- Flush Center (FC) centers the text between the left and right margins. Both the left and right edges are ragged.
- Flush Right (FR) pushes the lines of text against the right margin (RM).

ACTION

Setting the Text Alignment

To set the text to one of the three alignments:

1. Move the cursor to the start of the line whose alignment you want to change (or to a preceding line).
2. Enter FL, FC, or FR. To center the text, for instance:

   Type: F5 fc ↵

Result: The text following the embedded triangle is centered. All lines are centered up to the point where a counter-acting command (either FL, FR, or JU) is encountered. Being an embedded command, FC would appear in Expanded Display as «FC».

ALSO SEE

Flush Tabs. If you want to have both flush left text and flush right text on the same line, use flush tabs. Refer to "Tab Settings" elsewhere in this chapter.

FORMAT	CM JU	Justify On
	CM NJ	No Justify

PURPOSE

The **JU** (Justify on) command causes the text to be printed fully justified — that is, the text is even along both its left and right edges. JU and NJ are complementary — you use JU to turn justification on and NJ to turn it off.

XyWrite offers two kinds of justified text:

- Whole-Space Justification (described below)
- Microjustification (described in Chapter 6)

You choose one or the other by the settings in your Printer File (DF WS=1 at the beginning of your Printer File designates Whole-Space Justification). XyWrite printer files are initially set up for the highest quality printing — Microjustification. While Microjustification has a better appearance than Whole-Space Justification, it is very slow on some printers.

ACTION

Selecting Text to be Justified

To turn on justification:

1. Move the cursor to the start of the line you want justified, or to a preceding line.
2. Type: F5 ju ↵

Result: The text following the JU embedded triangle will be justified when it is printed. The text does not automatically appear justified on-screen. Justification stays in effect throughout the remainder of the document unless NJ is encountered to turn it off. To turn off justification:

1. Move the cursor to the line whose alignment you want to change.
2. Type: F5 nj ↵

Result: The text following the NJ is flush left (or flush center or flush right if it was last in effect.)

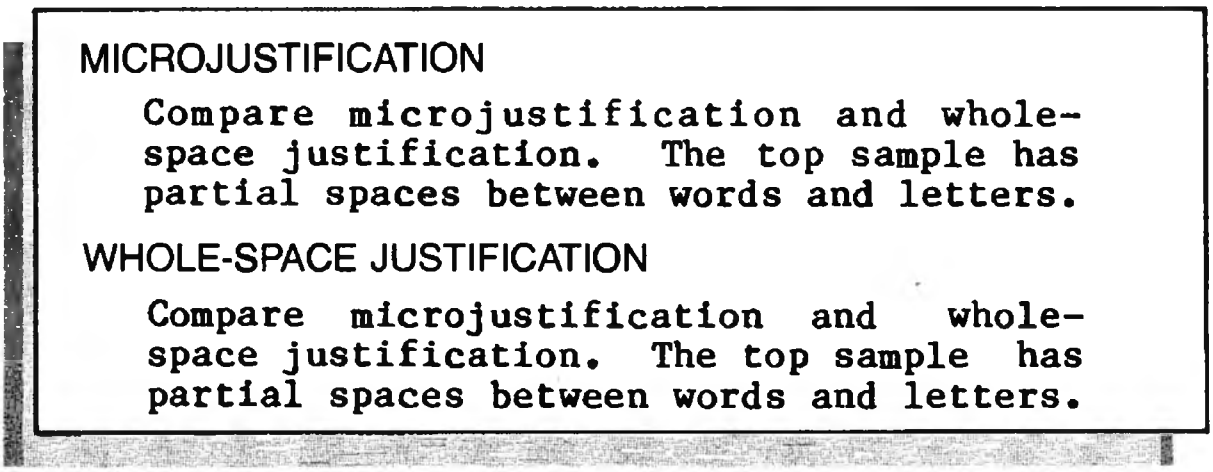
NOTE #1 **Hard Returns/Soft Returns.** Justification operates only on lines ending with a soft return (word wrap) — it does not operate on lines that end with a carriage return.

NOTE #2 **Comparison of Justification Types.** In short, the two types of justification work as follows:

- Whole-space justification — XyWrite calculates the number of spaces by which a line would be short of the right margin, and then distributes those spaces equally between words (but not between characters of a word).

- Microjustification — This method is based on inserting partial spaces not only between words but also between characters. The effect is smoother spacing than with Whole-Space Justification.

Microjustification is described further in Chapter 6, "Customizing."



NOTE #3 **Overriding Commands.** JU and NJ override the flush commands FL, FC and FR. To return to FL from JU, for instance, you type NJ (not FL).

NOTE #4 **Combinations of Alignments.** If you follow FL or FR with JU, the body of text is justified; however, short lines — lines which are not justified — are pushed flush left or right according to whether FL or FR preceded JU.

NOTE #5 **Embedded Commands.** JU and NJ are embedded commands — they are embedded in text, visible as triangles in Normal Display. In Expanded Display they are shown expanded — for example: «JU».

NOTE #6 **Display of Justification.** XyWrite does not try to show the text justified on-screen; the text is justified when printed.

FORMAT Ctrl Space Bar

PURPOSE The Non-Breaking Space inserts a space that will not break at the end of a line. This is useful when you want to prevent two words from ever being separated.

ACTION **Inserting a Non-Breaking Space**
To insert a Non-Breaking Space between the words *Route* and *66*:

Press: Route Ctrl Space Bar 66

Result: Now "Route 66" will always appear on one line.

NOTE **ASCII 32.** The Non-Breaking Space is really the character ASCII 32. You can search for it with the SEARCH command by typing Ctrl Space Bar on the command line as you would any other character (it appears on the command line as the three digits: space, 2, 0).

TIP **Making the Non-Breaking Space Visible.** If you want to distinguish the non-breaking space in the display from the normal space:

1. Select a character (perhaps the center dot, ASCII 250).
2. Assign it to Ctrl Space Bar in the Keyboard File:

   ```
   TABLE=CTRL

   57= · ←
   ```

3. Assign the *space* character for printout in a Character Substitution File:

   ```
   · = ←
   ```

PURPOSE

Hyphenation can be automatic or manual. In automatic hyphenation, XyWrite breaks the words as you enter them. It uses an internal set of hyphenation rules; exceptions to the rules are handled in an exception dictionary. You can turn automatic hyphenation on and off in different areas of text with the embedded command HY.

ACTION

Turning on Automatic Hyphenation.
To turn automatic hyphenation on, load a hyphen exception dictionary. DICTION is the file that is supplied with XyWrite:

Type: F5 `lddict diction` ↵

Result: This loads the hyphen dictionary and turns hyphenation on for all documents.

NOTE #1

Altering the Automatic Hyphens. If you want to make changes to the way XyWrite hyphenates specific words, change the hyphen exception dictionary. For information on how to do this, see the LDDICT command in Chapter 6.

NOTE #2

Creating Other Hyphenation Dictionaries. You can create and use other hyphenation exception dictionaries for special purposes. You do not need to use the file DICTION that we provide.

NOTE #3

Using LDDICT to Turn Off Hyphenation. You can disable hyphenation by giving the LDDICT command with no filename. This clears the current dictionary from memory and turns off the automatic hyphenation. Or use the HY command described in the next section.

NOTE #4

Manual Hyphenation. You can override the automatic hyphenation of a word by inserting a soft hyphen within the word in your document. If you place the soft hyphen in front of the first letter of the word, the word will not be broken.

FORMAT		
	CM HY ON	Turns hyphenation on
	CM HY OFF	Turns hyphenation off
	HY is an embedded command.	

PURPOSE

The **HY** command lets you turn automatic hyphenation on and off within a document.

For automatic hyphenation to work, it is necessary that (1) you load a hyphenation dictionary with LDDICT, and (2) that HY be *on*. By default, HY is *on* at startup (see Note #2).

ACTION

Switching Hyphenation On and Off
Load the hyphenation dictionary (or make sure it has already been loaded):

Type: F5 lddict diction ↵

Result: Automatic hyphenation is now turned on.

To turn off automatic hyphenation:

1. Position the cursor where you want to disable hyphenation.
2. Type: F5 hy off ↵

Result: The embedded command tells XyWrite not to automatically hyphenate any text that follows.

To again turn on automatic hyphenation:

1. Position the cursor where you want to enable automatic hyphenation.
2. Type: F5 hy on ↵

NOTE #1

Loading LDDICT. If you have not loaded an exception dictionary with LDDICT, HY has no effect on the file.

NOTE #2

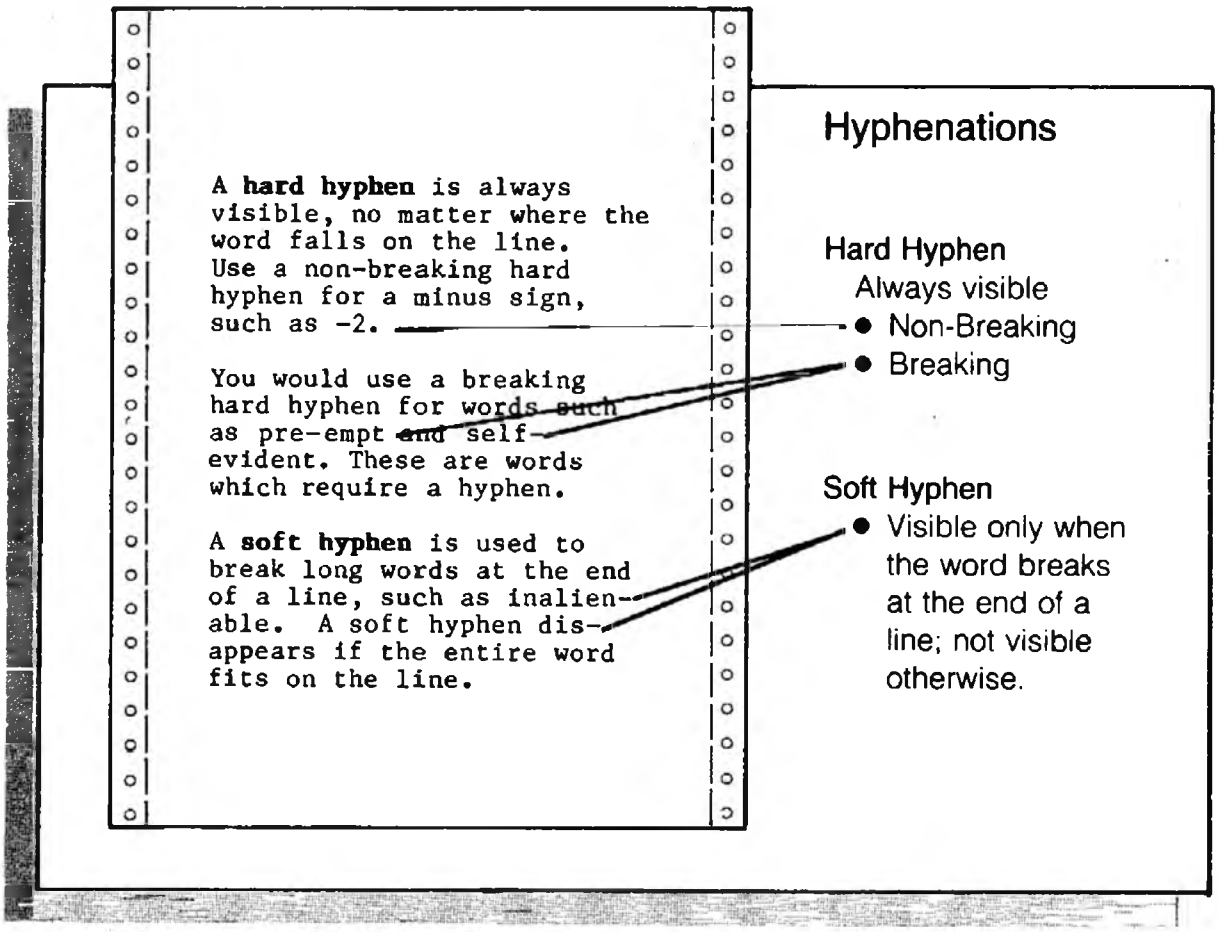
Default Hyphenation Setting. The default at startup is HY ON unless you use the DEFAULT command to change it.

NOTE #3

Hyphenation Parameters. You can control the minimum size of hyphenated words and the number of letters before and after a hyphen. See the LDDICT command, Chapter 6.

PURPOSE

XyWrite has three different hyphens: a non-breaking hard hyphen, a breaking hard hyphen and a soft hyphen. Examples of each are shown in the illustration below.



ACTION

Inserting Hyphens

To insert the three different hyphens, refer to the following chart.

Type of Hyphen	Action
• Non-Breaking Hard Hyphen	
Alt Shift 45	Press and hold Alt Shift, type the number 45, then release Alt Shift.
• Breaking Hard Hyphen	
−	Use either the hyphen located in the top row next to the "0", or use the minus sign on the numeric keypad.
• Soft Hyphen	
Shift ~	The tilde key is located immediately left of the Enter key (on a standard IBM keyboard).

ACTION

Deleting a Soft Hyphen

If a soft hyphen is visible at the end of a line, you can delete it as you would any other character. However, if that word is moved to the middle of the line, the soft hyphen will not be visible. To delete a soft hyphen (in either case), do the following:

1. Switch to Expanded Display with Ctrl F9. The soft hyphen will show as a tilde (~).
2. Type: F5 search /~/↵
3. Type: Backspace

Result: Step 3 deletes the tilde that was found in Step 2.

ACTION

Deleting a Hard Hyphen

Because a hard hyphen is always visible, it's easier to find than a soft hyphen. You search for a Non-Breaking Hard Hyphen differently than you do for a Breaking Hard Hyphen.

- To search for a Non-Breaking Hard Hyphen:

  Type: F5 search /

  Type: Alt Shift 45

  Type: /↵

  Result: This appears on the Command Line as:

  CM search / 2D/

- To search for a Breaking Hard Hyphen:

  Type: F5 search /-/↵

  Use either hyphen on the keyboard.

NOTE #1

Hard Hyphen. A Hard Hyphen is *always visible*, whether the word appears in the middle or at the end of a line. XyWrite has two kinds of Hard Hyphens.

- **Breaking Hard Hyphen.** Use this hyphen in words where the hyphen is a normal part of the word (words such as self-evident, pre-empt), and where you want to allow the word to break at the hyphen at the end of a line.
- **Non-Breaking Hard Hyphen.** Use this hyphen for the minus sign (–2), and for hyphenated words which you don't want to break at line endings — company or product names, like Lotus 1-2-3.

NOTE #2

Soft Hyphen. Use a soft hyphen to break a long word between syllables when the word falls at the end of a line. Use the soft hyphen only for words which would otherwise not be hyphenated. Then the soft hyphen is visible only when the word breaks at the end of a line, and is not visible otherwise (except in Expanded Display).

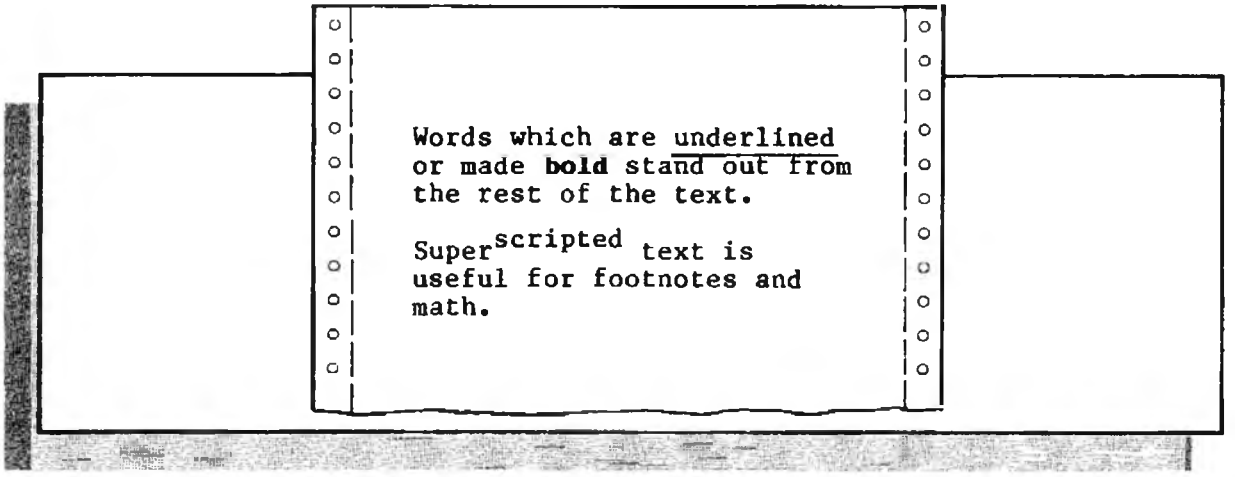
TIP

Positioning the Cursor to Insert a Soft Hyphen. If you hadn't realized, throughout XyWrite it is the *left edge* of the cursor rectangle that indicates where characters are inserted (in Insert mode). Thus you place the cursor on the character to the *right* of where you want to insert the soft hyphen.

INTRO

In order to add emphasis to text, you can use the MD commands to modify it with underlining, bold, reverse, superscript, subscript, and various combinations. (You can also make up your own modes, such as strikethrough or double-underlining — see the "Printer File" section in Chapter 6.) How these modes are actually displayed and printed depends on the capabilities of your hardware. You can also change the typeface with the PT command—for Pica, Elita or Proportional type.

CONTENTS



FORMAT CM MD *nn* or Ctrl #

(Mode Command) (Control Key)

nn is a two-letter mode command or a number
is a number from 0 to 8 at the top row of the keyboard
These are embedded commands

PURPOSE

Character modes allow you to highlight text as listed below. In the Normal Display these modes are displayed as they would be printed. Using the Ctrl key with a number is a convenient method for allowing you to enter text in a new mode without affecting existing text. When you press Ctrl 2, anything you type will be in bold.

The value *nn* in the format statement above can be any of the 13 two-letter commands shown below, or any number 0 to 255 from the Color Table. See Note #5 for further information on the Color Table.

Mode Command	Control Key	Character Mode
—	Ctrl 0	Reset (Default)
MD NM	Ctrl 1	Normal
MD BO	Ctrl 2	Bold
MD UL	Ctrl 3	Underline
MD RV	Ctrl 4	Reverse
MD BU	Ctrl 5	Bold Underline
MD BR	Ctrl 6	Bold Reverse
MD SU	Ctrl 7	Superscript
MD SD	Ctrl 8	Subscript
MD FN	—	Footnote
MD FL	—	Flashing
MD FU	—	Flashing Underline
MD FR	—	Flashing Reverse
MD SO	—	Standout (Flashing Bold)
MD *nn*	—	Color modes (*nn* = 1-255)

When using MD UL, you can control how tabs and spaces are underlined with the UL command. See Note #4.

The procedure for new text is different than that for existing text. These two procedures follow.

- Changing the Mode of Existing Text. (*Option 1*)
- Selecting a Mode for New Text. (*Option 2*)

ACTION
(*Option 1*)

Changing the Mode of Existing Text

There are two ways to change the character mode of existing text. We will use bold as an example.

Using the Ctrl Key. You define the block of text (steps 1-4) and then change the character mode of the entire block.

1. Move the cursor where you want the bold to start.
2. Press: F1
3. Move the cursor where you want the bold to end.
4. Press: F1
5. Press: Ctrl 2 to select bold
6. Press: F3 to release the defined text

Result: The defined text switches to bold. You can use any of the several methods available to define the text — define by word, by sentence or by paragraph.

Using the MD Command. You change the character mode at the start (Step 2) and then again at the end (Step 4).

1. Move the cursor where you want the bold to start.
2. Type: F5 md bo ↵

   Result: The text from this point to the end of the document (or to the next mode command) changes to bold.

3. Move the cursor where you want the bold to end.
4. Type: F5 md nm ↵

   Result: Only the text between the start (MD BO) and end (MD NM) is now bold. Being an embedded command, MD BO appears in the Expanded Display as «MDBO».

ACTION *(Option 2)*

Entering Text in a New Mode.
To enter text in new mode, use the Ctrl key. Let's use bold as an example.

1. Move the cursor to where you want to type in bold text.
2. Press: Ctrl 2 (to select bold)
3. Begin typing the bold text.
4. When done typing, reset the mode with Ctrl-zero:
   Press: Ctrl 0

NOTE #1 **How Modes Affect Hyphens.** Hyphens inserted by Automatic Hyphenation or as soft hyphens print out in Normal Mode rather than in the current mode. To print the hyphens in the current mode, enter the setting FO=4 in the Font Table of the Printer File (see Chapter 6).

NOTE #2 **Use Ctrl 0 for Reset.** When you first start up XyWrite, it is in the Ctrl 0 (Control-zero) mode. This is an adaptive mode; in it you can move the cursor about and enter text in whatever mode exists at the new cursor location. If you move the cursor to bold text, the new text is automatically entered in bold. After changing modes, you can return to the adaptive mode with Ctrl 0.

NOTE #3 **Underline on Color Monitors.** If you have a color monitor, the underline does not appear on your screen; it appears instead as blue text.

NOTE #4 **Controlling Underlining.** The UL setting lets you specify how tabs and spaces in the underline mode (MD UL) print out. The choices are as follows:

ul 0	Underline everything
ul 1	Underline everything but tabs
ul 2	Underline everything but tabs and spaces
ul 3	Underline only letters and numbers

UL 1 is the default. To enter the UL command:

1. Move the cursor to the top line of the file (or ahead of the text whose underlining you want to control).

2. Enter the UL command on the Command Line. For example, to underline everything in the MD UL mode except tabs:

   Type: F5 ul 1 ↵

NOTE #5

Colors. You can use color modes even if you don't have a color monitor. Although the colors will not be visible on a monochrome (black & white) monitor, they will produce the same printing styles that a color monitor would.

You use MD 1 to MD 255 to specify colors modes. (You can view these on-screen with the Help File by selecting BY_KEYWORD and typing in COLOR TABLE.) For example:

MD 31	Bright white characters on blue background
MD 4	Red characters on black background
MD 116	Red characters on white background
MD 208	Black characters on magenta background

You select the colors as follows (using the example of bright white characters on blue background):

1. Pick a color for the *characters* from the Color Table. Bright white characters is number 15.
2. Pick a color for the *background*. Blue background is number 16.
3. Add the two numbers together: 15+16=31
4. At the command line, type MD 31:

   Type: F5 md 31 ↵

NOTE #6

Printing of Character Modes. XyWrite provides a great deal of flexibility in how character modes are printed. You can specify in the Printer Table exactly how a mode is to be printed, depending on your printer's capabilities. For example, you can specify the bold mode, MD BO, to be printed as either doublestrike (exact overstrike) or as shadowprint (slight offset to overstrike). (In fact, you are free to use MD BO to represent italic or whatever printing mode you want.) Refer to "Printer File" in Chapter 6.

FORMAT		
	CM UC	Uppercase
	CM LC	Lowercase
	CM CC	Change Case

These are immediate commands (not embedded).

PURPOSE

The case commands allow you to change letters to uppercase or lowercase. You can change the character at the cursor location or change an entire block of defined text. Three different commands are at your disposal:

- **UC** (Upper Case) — Changes text to uppercase (capital letters)
- **LC** (Lower Case) — Changes text to lowercase
- **CC** (Change Case) — Changes uppercase to lower and lowercase to upper.

The following procedures provide two different ways to change the case of text:

- Changing the Case of Individual Letters. (*Option 1*)
- Changing the Case of Defined Text. (*Option 2*)

ACTION
(*Option 1*)

Changing the Case of Individual Letters

To change the case of individual letters:

1. Move the cursor to the character whose case you want to change.
2. Enter UC, LC or CC — for example, to change the character to uppercase:

   Press: F5 UC F9

3. Press F9 once for each character you want to make upper case.

Result: The lowercase characters are changed to uppercase. Uppercase letters remain uppercase.

ACTION
(Option 2)

Changing the Case of Defined Text

To change the case of a block of defined text:

1. Move the cursor to the *start* of the block of text you want to change.
2. Press: F1
3. Move the cursor to the *end* of the block you want to change.
4. Press: F1
5. Enter UC, LC or CC. For example, to change the case:

   Type: F5 cc F9
6. Press: F3 to release the defined block of text

Result: The defined text changes case in Step 5. Notice that by repeating F9, you can toggle the case back and forth, from upper to lower to upper case. (If this procedure doesn't work, press F3 and try again.)

FORMAT	CM AU	Automatic Uppercase
	AU is an immediate command.	

PURPOSE

The **AU** (Automatic Uppercase) command automatically capitalizes the first letter of each sentence, reducing your need to use the Shift key.

ACTION

Typing with Automatic Uppercase. Automatic Uppercase command is a toggle — you issue it once to turn it on and again to turn it off. To use the Automatic Uppercase feature:

1. Move the cursor where you want to start using Automatic Uppercase.

2. Press: F5 au ↵ (to turn on AU)

   Result: The letter **A** appears at the top right corner of the screen, to indicate Automatic Uppercase is turned on.

3. Press: F10 (to move cursor to text area)

4. Begin typing. The first letter following a period (.), question mark (?), exclamation mark (!) or ↵ (Enter) is automatically typed uppercase without your having to use the Shift key.

5. When finished typing:

   Press: F5 au ↵ (to turn off AU)

NOTE

How AU Works. The AU command capitalizes the first letter following a period, regardless of what the period is used for — including abbreviations and numbers (as a decimal point). If you have a lot of abbreviations or numbers, this mode may not be for you. However, if you are typing straight text, this mode can save you time.

FORMAT	CM**PT** *n*	Print Type
	n is 1, 2 or 3 You can add fonts to your Printer File that use values of *n* from 1 to 255. PT is an immediate command.	
EXAMPLE	CM**pt 1**	

PURPOSE

The **PT** (Print Type) command specifies the typeface (font) in which your text is printed. For most printers, XyWrite is set up as follows:

- PT 1 — 10-pitch (Pica or Courier)
  All characters are the same width, at 10 characters per inch.
- PT 2 — 12-pitch (Elite)
  All characters are the same width, at 12 characters per inch.
- PT 3 — Proportional type
  Characters have different widths — for instance, an "i" is narrower than a "w." (For some printers, PT 3 is 15-pitch instead.)

Since the PT 2 characters are narrower than the PT 1 characters, you can fit more text on a page if you use PT 2. PT 1 characters are wider, and therefore easier to read.

Proportional type has a classier appearance, and is often preferred. However, proportional type is not available on all printers, and is less useful in tables or columnar data.

For a few printers, including laser printers, PT 2 and PT 3 are not as described above. The best way to see what fonts your printer has is to print out the test sheet shown in the illustration "Testing Your Printer's Fonts."

Typically, you choose a default value for PT you like for your body copy (such as PT2). Then you might change the font (to PT 1) for titles to make them stand out. You are free to change fonts back and forth as many times as you wish. You can use PT to change entire documents or to change a single word or character.

ACTION **Setting a Type Font.**
To set text to be printed in a particular type font:

1. Move the cursor to the start of the text where you want the new font to begin.
2. Enter the PT command you want. For example, if you want Elite:

   Type: F5 pt 2 ↵

Result: The PT is embedded in the text as a triangle. From this point forward, the text will be printed in Elite (12 characters per inch).

NOTE #1 **More Fonts.** With a little work, you can add fonts beyond the standard PT 1, PT 2, and PT 3, if your printer is capable. For instance, an Epson FX printer is capable of:

PICA/EXPANDED
COMPRESSED
ELITE/EXPANDED
COMPRESSED/EXPANDED
PROPORTIONAL/ITALIC
PROPORTIONAL/EXPANDED
PROPORTIONAL/ITALIC/EXPANDED.

Thus, you could assign ELITE/EXPANDED font to PT 4. Call up your Printer File (CALL 3EPSONFX.PRN) to see what capabilities are listed at the top of the file. Read the Printer File section in Chapter 6 for information on how to assign those fonts to PT values.

NOTE #2 **Default Setting.** The default value is PT 1. This causes your printer to print 10 characters per inch. This is a convenient setting — it means the ruler at the top of the screen measures *inches* (and each dot on the ruler corresponds to 1/10", or one character).

If you consistently use a font other than PT 1 then make that the default instead. To make PT 2 the default, add DEFAULT PT 2 to your STARTUP.INT file (see Chapter 6). You can also set the default in your Printer File with DF PT 2.

TESTING YOUR PRINTER'S FONTS

Simply create a file with the following text and PT commands, and then print it. The resulting printout is shown below.

SCREEN

«PT1»→ ▲
This paragraph is PT 1. For an Epson FX, this produces PICA type (10 characters per inch).

«PT2»→ ▲
This paragraph is PT 2. For an Epson FX, this produces ELITE type (12 characters per inch).

«PT3»→ ▲
This paragraph is PT 3. For an Epson FX, this produces PROPORTIONAL type (about 11 characters per inch on the average).

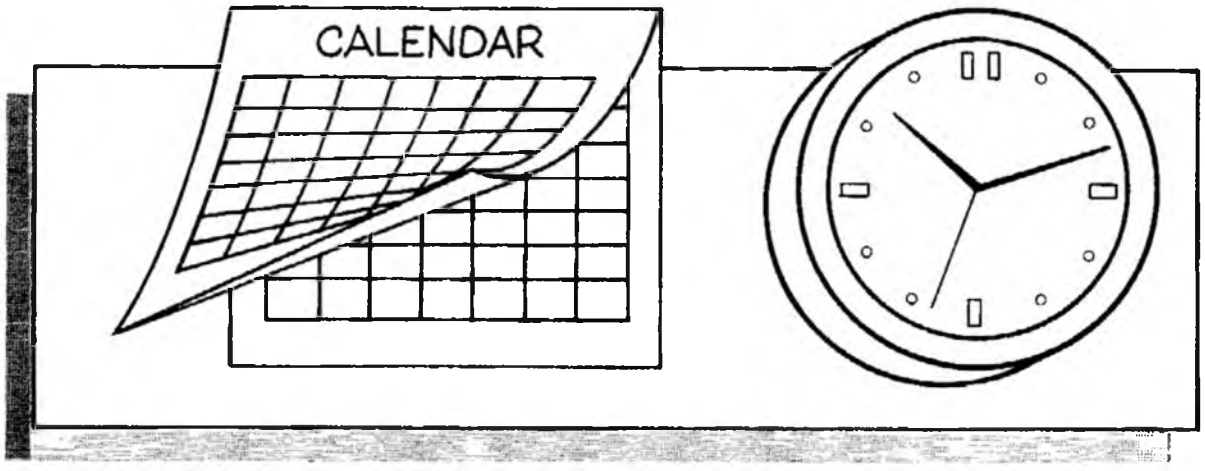
```
This paragraph is PT 1. For an Epson
FX, this produces PICA type (about 10
characters per inch).

This paragraph is PT 2. For an Epson
FX, this produces ELITE type (about 12
characters per inch).

This paragraph is PT 3. For an Epson
FX, this produces PROPORTIONAL type (about
11 characters per inch on the average).
```

NOTE #3 **Embedded Command.** PT is an embedded command, visible as a triangle in the Normal Display. In the Expanded Display, it would appear expanded, as «PT2».

INTRO

XyWrite will enter the correct date and time into your document for you in either of two ways: with the current date and time (with DA and TM) that continually updates or as a time stamp (with TODAY and NOW) which does not update.

CONTENTS



FORMAT

CM **DA** *form*	Current Date
CM **TODAY**	Fixed Date

form (optional) is any combination of **m** (month), **d** (day) and **y** (year) listed later in options, such as **mm/dd/yy**.
DA is an embedded command.
TODAY is an immediate command.

PURPOSE

The **DA** (Date) and **TODAY** commands both insert the current date into your text. DA inserts a *soft* date — it is continually updated, always displaying the current day, month and year. TODAY inserts a *hard* date, as permanent text, never updated. The clock internal to your computer automatically provides the current month, day and year.

You have great flexibility in how you display the date. See the examples later in this section. If you don't specify a form, the date format defaults to "letterhead" style (full month, date, full year), such as:

```
April 1, 1986
```

However, you can change that default by placing a DA setting in the Printer File; see the note "Changing the Default Date Format." The TODAY command does not take a modified form on the Command Line, and can be changed only by changing the default.

ACTION

Inserting the Soft Date.
To insert a date which is continually updated:

1. Move the cursor to where you want the *first character* of the date to be located.
2. Type DA followed by any of the options described on the next page. To give one example:

   Type: F5 da MMMM 'yy ↵

Result: The date appears in the text after an embedded triangle. The triangle is removed upon printing:

▲APRIL '86	as displayed
APRIL '86	as printed

This command would appear in Expanded Display as: «DAMMMM 'yy»

ACTION **Inserting the Hard Date.**
To insert a date as permanent text:

Type: F5 today ↵

Result: The date will be placed at the cursor location in your text. The default format will be used. The date is fixed text — it will not be updated when the file is used later.

The TODAY command will *not* accept date formats on the command line as DA does. You can change the TODAY date form only by putting a DA setting in your printer file, which sets the default form for both DA and TODAY. See the note below.

NOTE **Changing the Default Date Format.** When you use the TODAY command, or DA without a form, the date appears in "letterhead" form. But you can change that default form with a DA setting in your printer file.

For a European-style format (day, month, year), for example, call up your Printer File and insert the following line, using the "less than" symbol (<) after DA:

```
DA<d Mmm yy
```

Once the printer file is saved and loaded, any DA command without a form (or the TODAY command) will yield a date with the form:

```
1 Apr 86
```

See the "Printer File" section in Chapter 6 for more on how to put settings in the printer file.

OPTIONS

Month. Note below that you use uppercase M's where you want uppercase letters to appear in the month.

Format		Examples	
•	Complete name of month: Use four M's.	MMMM Mmmm	APRIL April
•	Three-letter abbreviation: Use three M's.	MMM Mmm	APR Apr
•	Two-digit months 01-12: Use two M's.	mm	04
•	Months 1-12 (No leading zero): Use one M.	m	4

Day. The day of the month can be expressed any of three ways:

•	Leading space for days 1-9: Use three D's.	ddd	1
•	Leading zero for days 1-9: Use two D's.	dd	01
•	No leading zero or space: Use one D.	d	1

Year. The year can be expressed in either of two ways:

•	Complete four-digit year: Use four Y's.	yyyy	1986
•	Two-digit year: Use two Y's.	yy	86

EXAMPLES

Combinations of Month, Date and Year. You can combine the month, day and year in any order. You can also insert any punctuation you want printed. If the current date were April 1, 1986, the following combinations would yield the dates shown:

Mmmm d, yyyy	April 1, 1986
d Mmm yy	1 Apr '86
m/d/yy	4/1/86
m/ddd/yy	4/ 1/86
mm/dd/yy	04/01/86
MMMM 'yy	APRIL '86

FORMAT		
	CM **TM**	Soft Time
	CM **NOW**	Fixed Time
	TM is an embedded command. NOW is an immediate command.	

PURPOSE

The **TM** (Time) and **NOW** commands insert the current time into your text, as provided by the clock internal to your computer. TM enters the *soft* time — it's continually updated (like a clock). It shows the new, current time whenever it appears on the screen or is printed. NOW inserts the current time as non-changing text.

ACTION

Inserting the Soft Time.
To insert the soft time into your text:

1. Move the cursor where you want the first character of the time to be located.
2. Type: F5 tm ↵

Result: The time appears in the text along with an embedded triangle. Being an embedded command, TM would appear in the Expanded Display as «TM».

ACTION

Inserting the Fixed Time.
To insert the current time so it will not change:

1. Move the cursor where you want the first character of the time to be located.
2. Type: F5 now ↵

Result: The current time is inserted, fixed as ordinary text.

NOTE

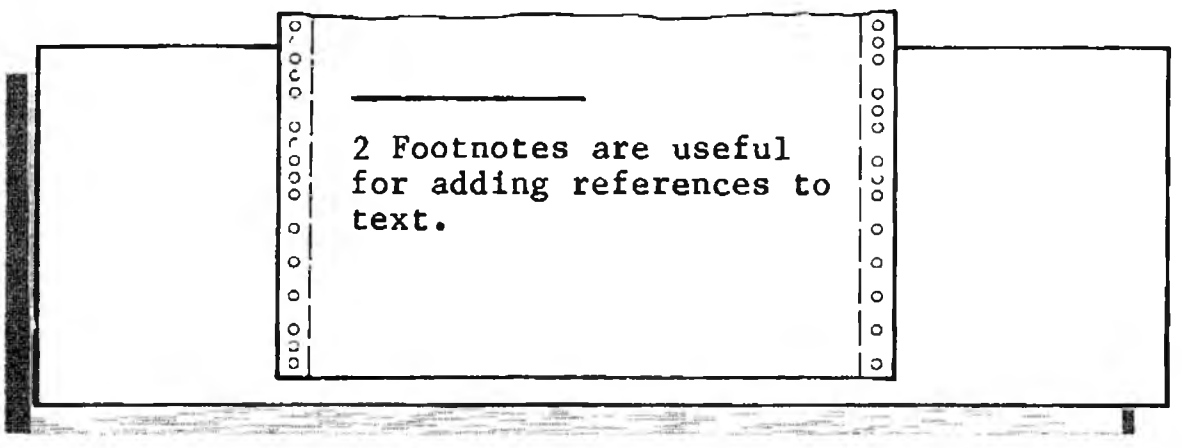
Updating. The time inserted by the TM command does not automatically update on the screen. You must press F3, change a character on that line, or move the TM command off and back on for it to update.

INTRO

If you use footnotes, you'll find XyWrite gives you a great deal of control over how they appear. You can easily place all footnotes at the bottom of the page, as basic footnotes, or at the end of your document, as endnotes. In either case, as you add or delete footnotes, XyWrite handles the numbering and placement of footnotes for you.

CONTENTS

This section begins with an overview and then describes each command individually:



A **footnote** is text which appears at the bottom of a page, referenced by number to a point above in the text. You can create footnotes at any time — either at the initial writing, or later when you return to edit the document. There is no limit to the size of a footnote — it can run more than a paragraph.

XyWrite keeps track of all footnotes, numbering them automatically for you. When you print the document, XyWrite puts each footnote at the bottom of its page.

Footnote Sets. XyWrite allows you to create up to three sets of footnotes in one document. You can make decisions about the style and format of each set independently — you can even elect to print one set at the bottom of the page and another set at the end of the document.

Creating a Basic Footnote is the first procedure described. All that's required for the simplest footnotes are the two commands:

- FN - Footnote command
- FS - Footnote Separator

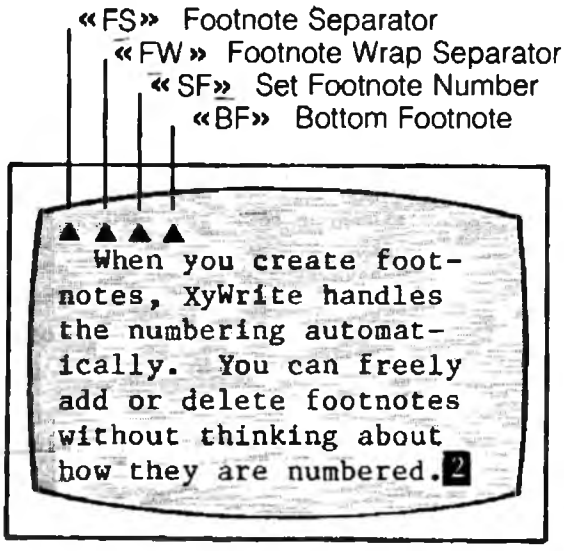
Creating Endnotes is the second procedure described. Endnotes are footnotes placed at the end of a chapter or document. Once you write your footnotes, endnotes are easy to produce, needing only two commands:

- NF - No Footnotes
- DF - Dump Footnotes

Variations on a Footnote describes other commands you can use to modify the appearance of footnotes:

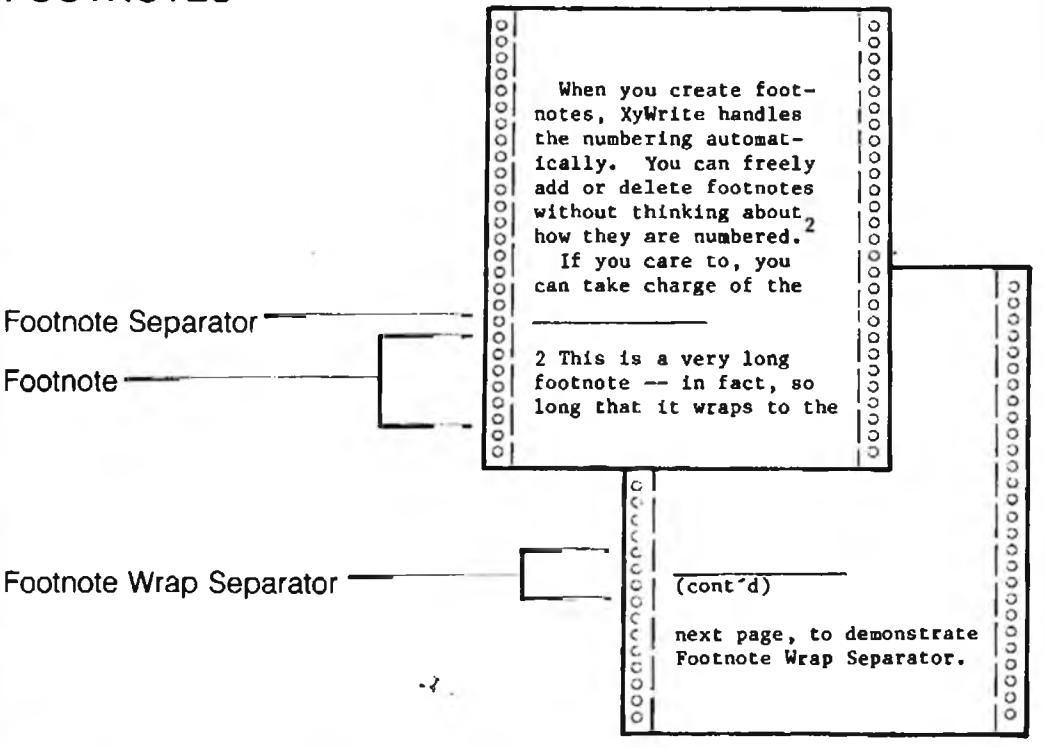
- FW - Footnote Wrap Separator
- SF - Set Footnote Number
- BF - Bottom Footnote
- FM - Footnote Format

DISPLAY OF FOOTNOTES



«FN This is a very long footnote—in fact, so long that it wraps to the next page, to demonstrate Footnote Wrap Separator.»

PRINTOUT OF FOOTNOTES



ACTION

Creating a Basic Footnote

This procedure requires only the two basic footnote commands FN (Footnote) and FS (Footnote Separator). To enter a footnote:

Creating the Footnote (FN).

1. Move the cursor to the point in the body of text you want marked for a footnote (that is, where you want the superscripted number).

2. Type: F5 fn ↵

   Result: The footnote window opens in the middle of the screen. If you had used a footnote set number, say FN2, other footnote commands ending in 2 (such as FM2, FS2 and so forth) would control it.

3. Type the footnote text, for example:

   This is the footnote; it is numbered automatically. When this page is printed, this footnote will automatically be printed at the bottom of the page.

   Be aware that you can use defined text or Save/Gets to help you enter text.

4. Press: F3 (to end the footnote)

   Result: The footnote window closes. The footnote number 1 appears at the cursor position. This footnote number actually represents an embedded command; you can return to edit this footnote text at any time by switching to the Expanded Display with Ctrl F9.

Creating the Footnote Format and Separator.

5. Move the cursor to anywhere ahead of the first footnote (usually the top of your document), in preparation for entering the footnote format.

6. Enter the format for your footnotes using the FM (Footnote Format) command. While this step is not always necessary, we strongly recommend using FM. For example:

   Type: F5 fm lm=8,rm=65 ↵

7. Next enter the FS (Footnote Separator) command:

   Type: F5 fs ↵

   Result: This opens the Footnote Separator window. Type in the footnote separator — the line which separates the body text from the footnotes. For example, we'll choose a series of underlines:

   Type: _____ ↵ ↵

   The second ↵ inserts a blank line between the Footnote Separator and the footnote.

8. Press: F3

   Result: This closes the Footnote Separator window and completes the procedure.

NOTE #1 **First Footnote Separator.** To ensure a uniform appearance, the separator associated with footnote set 1 (FS1) is always printed first at the bottom of any page that contains footnotes. This is true even if there is no footnote associated with set 1 on that page.

NOTE #2 **Editing Footnotes.** There are two ways you can edit the text in a footnote: (1) by switching to Expanded Display with Ctrl F9 or (2) by switching to Edit Footnote. To use Edit Footnote, place the cursor on the footnote number and press Ctrl F3. The footnote window opens up and displays the stored text so you can make edits. Close the footnote window by pressing F3.

NOTE #3 **Previewing the Footnotes.** As noted in the previous procedure, the footnote text is visible only in Expanded Display. However, you can view the footnotes on-screen, positioned properly at the bottom of each page with TYPES.

NOTE #4 **Printing the Footnotes.** After following the procedure "Creating a Basic Footnote," you can use TYPE to print your document with footnotes. The footnotes will automatically be printed just below the text on each page, separated from the text by the footnote separator.

ACTION **Creating Endnotes.**
Endnotes are footnotes which are collected and printed all together at the *end* of the document, rather than sprinkled throughout the document. At the start of the document you tell XyWrite to *hold* all footnotes (NF — No Footnotes); at the end you tell XyWrite to print them (DF — Dump Footnotes).

To enter endnotes, you first create footnotes throughout the document exactly as in the procedure "Creating a Basic Footnote." Then:

1. Move the cursor to the start of the document.

   Type: F5 nf1 ↵ (1 is optional)

2. Move the cursor to the end of the document, where you want to print all of the footnotes (in other words, where you want to *dump* the footnotes).

   Type: F5 df1 ↵ (1 is optional)

Result: The NF1 command (No Footnotes) in Step 1 suppresses the printing of footnotes from that point forward (up to DF1). However, the footnotes are still accumulated internally by XyWrite. The DF1 command dumps, or unloads, all of the footnotes at its location in the text.

If you use more than one set of endnotes (FN1, FN2, FN3), they're all dumped at the DF (or DF1) location. If you want them at separate locations, use NF1, DF1; NF2, DF2; and NF3, DF3).

NOTE #5 **Exact Location of the Endnotes.** The endnotes are not necessarily inserted at the exact location of the DF embedded triangle, but rather at the bottom of that page. The endnotes are positioned higher or lower on the page, according to the setting of BF (Bottom Footnote).

NOTE #6 **Using DF with Footnote Sets.** FS1, FS2 and FS3 create separators for their own sets of footnotes. Without the set number, FS generally creates a separator only for the initial set of footnotes (FS1).

An exception occurs when you use DF (Dump Footnotes) to put all the footnotes in one place. Then the FS or FS1 command is used as the separator for all footnotes.

NOTE #7 **Printing Only the Endnotes.** If you want to print out only the endnotes, use the previous procedure, placing the DF command on its own page at the end of the document. Then print from that page forward (for example: TYPE ,12–). To *view* the endnotes, use Type-to-Screen (TYPES ,12–). This would print all endnotes to the screen rather than to the printer.

ACTION **Variations on a Footnote.**
Beyond the basic footnote are the following variations. For more information, refer to the separate explanations given later on each of these commands.

Selecting a Footnote Wrap Separator. You can specify a different footnote separator to appear in footnotes that continue to a second page. *Refer to FW—Footnote Wrap Separator.*

Setting the Footnote Number. Set the footnote number with SF — all subsequent footnotes renumber automatically. *Refer to SF—Set Footnote Number.*

Positioning the Footnotes Up or Down. Set BF, the Bottom Footnote setting, to 1 or 0. This affects the placement of footnotes on a page. *Refer to BF—Bottom Footnote.*

Setting the Format for Footnotes. Set margins, tabs, line spacing and any other format settings with FM. *Refer to FM—Footnote Format.*

FORMAT	CM FN*s*	Footnote

s (optional) is the number of the footnote set (1, 2 or 3)
FN1 is the same as FN
FN is an embedded command.

PURPOSE

The **FN** (Footnote) command allows you to enter footnotes in text. You type in the text of the footnote as part of the FN command, then end it by pressing F3.

XyWrite lets you create as many as three independent sets of footnotes in the same document. If you need just a single set, use FN, which is the same as FN1. The other sets are FN2 and FN3.

When you're done entering the FN command, the footnote text is not normally visible until you print the document. At that time, XyWrite automatically inserts the footnotes at the bottom of their respective pages.

You can view the text in a footnote in three ways: (1) In Normal Display, when you move the cursor onto the footnote number, you can view the first part of the footnote on the Prompt Line. (2) At this point, if you press Ctrl F3, a screen will open to show you the text. (3) You can also switch to Expanded Display to view the text.

ACTION

Creating a Basic Footnote.
To enter a footnote, follow the procedure given a few pages back, "Creating a Basic Footnote."

NOTE #1

Embedded Command. FN is automatically entered into the text as FN1. Each FN command is embedded in text, visible as a number in Normal Display. In Expanded Display it is shown expanded — for example: «FN1~This is a footnote».

NOTE #2 **Footnote Margins.** To create footnote margins (or tabs or line spacing) different from those used in the text, use the FM (Footnote Format) command. FM1 (or FM) affects all footnotes in its set uniformly, as do FM2 and FM3.

NOTE #3 **Shortcut.** If the footnote text is only one line in length, you can type it all on the Command Line as follows:

Type: F5 `fn This is a one-line footnote.` ↵

FORMAT CM **FS***s* Footnote Separator

s (optional) is the number of the footnote set (1, 2 or 3).
FS1 is the same as FS
FS is an embedded command.

PURPOSE

The **FS** (Footnote Separator) command allows you to specify lines which separate the footnotes from the body text. The lines (which can contain text or can be blank) are located immediately above the footnote. An example of a Footnote Separator is illustrated at the start of the Footnote section. Very often people choose the footnote separator to be three lines:

- A blank line
- A row of hyphens, underlines or asterisks
- Another blank line

A Footnote Separator can be specified for each of three independent sets of footnotes as: FS1, FS2 and FS3. FS1 is the same as FS alone (with no number).

ACTION

Creating a Footnote Separator.
To create a Footnote Separator, follow the procedure given a few pages back, "Creating a Basic Footnote." Be aware of the following notes.

NOTE #1 **Location of FS.** Any Footnote Separator commands you use (FS1, FS2 and FS3) must be placed *ahead* of any footnotes. It is a good practice to place the commands at the top of the document, where they are easy to find.

NOTE #2 **Embedded Command.** FS is automatically entered into the text as FS1 (not simply as FS). Each FS command (FS1, FS2 and FS3) is embedded in text, visible as a triangle in Normal Display. In Expanded Display it is shown expanded, as: «FS2_____»

FORMAT CM FW*s* Footnote Wrap Separator

s (optional) is the footnote set number (1 to 3).
FW1 is the same as FW
FW is an embedded command.

PURPOSE

XyWrite tries to put the entire footnote on the page it belongs. If it won't fit, the footnote will automatically continue (wrap) on the next page. This next page is where the Footnote Wrap Separator is used.

The **FW** (Footnote Wrap Separator) command is very similar to the regular Footnote Separator command FS, but with one difference: FW defines footnote separators only for footnotes that wrap, or continue, from the previous page. Very often people choose the Footnote Wrap Separator to be four lines:

- A blank line
- A row of hyphens, underlines or asterisks
- A line containing the text: *(continued)*
- Another blank line

The line *(continued)* refers to the fact that the footnote is continued from the previous page.

A Footnote Wrap Separator can be specified for each of three sets of footnote separators. The footnote wrap separators correspond directly to the three commands FS1 (or FS), FS2 and FS3.

ACTION

Creating a Footnote Wrap Separator.
To create a footnote wrap separator:

1. Move the cursor to anywhere before the first footnote in the set. (It's a good practice to use the *top* line.)

2. Enter any FW command *in addition* to any FS command.

   Type: F5 fw1 ↵ (the 1 is optional)

   Result: The Footnote Wrap Separator window for footnote set 1 opens on the screen.

3. Now type the lines you want to define as the wrap separator. Important: **Make the Footnote Wrap Separator the same number of lines as the Footnote Separator you are already using**. For example, let's enter a row of underlines and the word "continued" in parentheses:

   Type: ↵ _____ ↵ (continued) ↵

4. Press: F3

   Result: This closes the Footnote Wrap Separator window and completes the procedure.

NOTE #1 **Embedded Commands.** FW is inserted into the text as FW1 (not simply as FW). Each FW command is visible as a triangle in Normal Display. In Expanded Display it is shown expanded — for example: «FW3(continued)»

NOTE #2 **If FW is Left Unspecified.** If you don't specify a Footnote Wrap Separator, XyWrite will use the regular Footnote Separator for footnotes that continue to another page.

NOTE #3 **What Causes a Footnote to Wrap.** Two conditions cause a footnote to wrap to the next page: (1) The footnote reference in the body of text is located near the end of the page, and (2) the footnote is lengthy.

Before causing a footnote to wrap, XyWrite will first attempt to move the text containing the footnote (along with the footnote itself) to the next page. However, it will not allow the text and footnote to fall short of the minimum page length (set by PLmin).

FORMAT CM BF *n* Bottom Footnote

n is 0 or 1
BF is an embedded command.

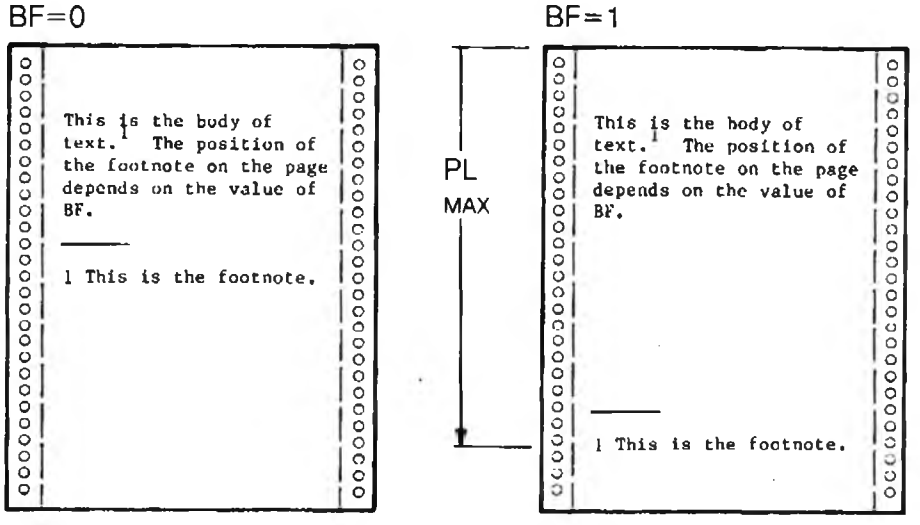
PURPOSE

The **BF** (Bottom Footnote) command allows you to specify whether footnotes (and footnote separators) are placed in either of the two positions shown in the illustration below. BF actually makes a difference only on pages which are not filled with text, such as the last page of a chapter. (Otherwise, the two positions appear exactly the same.)

- BF 0 means footnotes are pushed *up* against the last line of text.
- BF 1 means footnotes are pushed *down* against the last line of PL (which is PLmax).

One way to remember which is which is by remembering that BF 1 means that the Bottom Footnote is in effect— the footnote is against the bottom.

USING BF TO POSITION THE FOOTNOTES



ACTION **Positioning the Footnotes Up or Down.**
To enter the BF command:

1. Move the cursor anywhere before the footnotes you want to affect. (We recommend the top line of your document.)

2. To set the BF command to 1:

   Type: F5 bf 1 ↵

Result: When you print the document, the footnotes will be positioned according to BF 1, described earlier.

NOTE #1 **Embedded Command.** BF is an embedded command — it is embedded in text, visible as a triangle in Normal Display. In Expanded Display it is shown expanded — for example: «BF1»

NOTE #2 **Default.** The default value of BF is 0.

NOTE #3 **Footnote Format.** You can include the BF command in the FM (Footnote Format) command.

FORMAT	CM DF*s*	Dump Footnotes
	s (optional) is the footnote set number (1 to 3). DF1 is the same as DF DF is an embedded command.	

PURPOSE

The **DF** (Dump Footnotes) command is used with the NF (No Footnotes) command to place all footnotes together at the end of the document, as *endnotes*, rather than on the pages to which they refer.

To dump means to output the footnotes as text to the screen, file or printer. DF and NF are complementary commands — DF instructs XyWrite to output the footnotes, while NF instructs XyWrite to hold off outputting them. To produce endnotes, you do the following:

- **NF Command.** Place the NF command at the *top* of your document. NF suppresses the printing of footnotes until a later DF command is encountered in the text.
- **DF Command.** Place the DF command at the *end* of your document. DF causes the printing of footnotes that have accumulated since the NF command. The footnotes start printing at the bottom of the page containing the DF embedded triangle.

You can have up to three independent sets of footnotes. Whenever you use a DF command (DF1, DF2, or DF3), you must use the corresponding NF command (NF1, NF2, or NF3).

XyWrite tries to position all the footnotes on the page that contains the DF embedded triangle. If they won't all fit, they continue onto the next page.

ACTION

Entering the DF Command.
To enter the DF command, follow the procedure given earlier, "Creating Endnotes."

NOTE

Embedded Command. DF is automatically entered into the text as DF1 (not simply as DF). Each DF command is embedded in text, visible as a triangle in Normal Display. In Expanded Display it is shown expanded as «DF1».

ALSO SEE

Related Command. Also see NF (No Footnotes).

FORMAT **CM NF*s*** — No Footnotes

s (optional) is the number of the footnote set (1, 2 or 3).
NF1 is the same as NF
NF is an embedded command

PURPOSE

The **NF** (No Footnotes) command turns off the printing of footnotes. (The footnote numbers still appear in the text, though.)

Using NF in conjunction with the DF (Dump Footnote) command, you can place all the footnotes at the *end* of a chapter or document.

NF and DF are complementary commands. NF turns off the printing of footnotes, while DF turns on the printing of footnotes.

You can have up to three independent sets of footnotes. Whenever you use an NF command (NF1, NF2, or NF3) you must use the corresponding DF command (DF1, DF2, or DF3).

Like all embedded commands, NF takes effect at the point it is placed in the document, and affects the remainder of the document — that is, until a DF command is encountered. Thus, you would place NF at the *top* of a document in order to inhibit the printing of all footnotes.

ACTION

Entering the NF Command.
To enter the NF command, follow the procedure given earlier, "Creating Endnotes."

NOTE

Embedded Command. NF is automatically entered into the text as NF1 (not simply as NF). Each NF command is embedded in text, visible as a triangle in Normal Display. In Expanded Display it is shown expanded as «NF1».

ALSO SEE

Related Command. See the DF (Dump Footnotes) command.

FORMAT **CM SF***s n* Set Footnote Number

s (optional) is the number of the footnote set (1, 2 or 3).
n (optional) is the footnote style and starting value.
SF1 is the same as SF.
SF is an embedded command.

PURPOSE

The **SF** (Set Footnote Number) command sets the style (that is, numbers, letters, symbols) and the starting value of footnotes. You place the SF command ahead of the first embedded footnote command (FN) that you want to affect.

Since XyWrite supports up to three separate sets of footnotes, you can designate a footnote style and starting value independently for each set with SF1, SF2 and SF3. (SF is the same as SF1.)

There are six different styles from which you can choose:

Decimal numbers	(SF 1)
Uppercase roman numerals	(SF I)
Lowercase roman numerals	(SF i)
Uppercase letters	(SF A)
Lowercase letters	(SF a)
Defined string	(SF **n*)

If you want to start the sequence with a different value, use that value in the command. For example, the command SF iii tells XyWrite to number the footnotes in lowercase roman numerals and to start the sequence at iii. If you do not specify an SF command, XyWrite uses decimal numbers starting at 1.

This section includes the following procedures:

- Selecting the Style and Starting Value.
- Restarting the Sequence.
- Setting Unnumbered Footnotes
- Defining Footnote Symbols

ACTION **Selecting the Style and Starting Value.**
To define the style and starting value for footnotes:

1. Move the cursor to anywhere ahead of the footnotes you want to affect.

2. Decide what style and what starting point you want to use. For illustration purposes, let's use lowercase letters starting with "e" for footnote set 2:

   Type: F5 sf2 e ↵

Result: Footnote set 2 is assigned lowercase letters e, f, g, and so on.

ACTION **Restarting the Sequence.**
To restart the selected sequence at the beginning of each page:

1. Move the cursor to the beginning of the file.
2. Select the footnote set number, style and starting value. Insert a hyphen (-) before the starting value to make the sequence restart at each new page:

   Type: F5 sf2 -1 ↵

Result: Footnote set 2 is assigned decimal numbers starting at number 1 on each page. When you choose this option, the on-screen representation of each footnote is a superscript 2. The correct symbols are inserted when the document is printed. (See Note #3.)

ACTION **Setting Unnumbered Footnotes.**
To print a footnote that is unnumbered (has no identification symbol):

1. Move the cursor to the top of the page you want to affect.
2. Select the footnote set that you want to be unnumbered (for example, set 3). Enter the SF command without a style:

   Type: F5 sf3 ↵

Result: Footnotes in set 3 are printed without numbers or symbols of any kind. The footnote numbers are represented on-screen by a superscript n which does not print.

Unnumbered footnotes are useful in several places. For example, for an author's credit on the front page of an article, when you have only one footnote per page or when you are making reference to a registered trademark.

ACTION **Defining Footnote Symbols.**

You can create your own sequence of footnote symbols. You do this by entering into your Printer File a Counter String table (CS:) that lists strings in sequence. (See Note #1.) To use these strings:

1. Move the cursor to the top of the page you want to affect.
2. Select the footnote set that you want to use, and select which string from the Counter String table you want to start with. For example:

   Type: F5 sf3 *2 ↵

Result: For footnote set 3, use the characters that are defined in the Counter String table of the Printer File, starting with the second string in that list, and proceeding to the third, fourth, and so on. If there is no Counter String table defined, XyWrite uses: *, **, ***, etc.

NOTE #1 **Entering the Counter String Table.** The Counter String table in the Printer File defines the strings that can be used as footnote symbols. The format of the Counter String table is:

```
cs:#
string1
string2
  .
  .
  .
string#
```

where # is the number of strings. This table can also be used for page numbers and counters.

NOTE #2 **Recycling Symbols.** If you choose a footnote style with a limited number of symbols (e.g., lowercase letters or the Counter String table), you may run out of symbols before you run out of footnotes. When that happens, XyWrite returns to the beginning of the list of possible symbols and inserts the first one twice, then the second one twice, etc. The list is repeated as many times as necessary.

NOTE #3 **Line Breaks When Restarting the Sequence.** If you are restarting the sequence of footnote symbols on each page, your on-screen line breaks may not match the printed line breaks exactly. This is because the width of the symbol (represented on-screen by a superscript 2) is calculated as three times the width of the character 0 (zero). If the width of a footnote symbol is wider than that, change the FU setting in the Printer File. This setting is FU=*n* where *n* is the number of 0's to count when calculating the width of the footnote symbol. (A second value in the FU setting affects Counters and References.)

For example, suppose you are using asterisks as footnote symbols. You might have four footnotes on one page and therefore four asterisks as a footnote symbol. Unless you change the FU setting, the lines containing four asterisks may extend beyond the right margin.

NOTE #4 **Footnote Numbers in Chained Files.** You can have the Set Footnote Number command affect only the screen display of a document and not the printed output. To do this, insert a question mark (?) in the command, immediately before the footnote style. For example:

Type: F5 sf1 ?a ↵

This command will display footnote symbols starting at "a" for footnote set 1. However, it will have no effect on printed output. That is, when printed, the footnotes will be numbered according to the sequence defined in another file. This command is useful if you are chaining files to the printer (with TYPE @) and want the footnote numbering to proceed sequentially from one file to the next.

FORMAT **CM FM***s nm=n,nm=n,nm=n, . . .* Footnote Format

s (optional) is the number of the footnote set (1, 2 or 3)
nm is a format command (LM, RM, TS, IP, LS, OF, FL, FC, FR, JU, NJ, BF, FT, or SC).
= (equals sign) separates the name from the value.
n is the value of the format command.
, (comma) separates the commands.
FM is an embedded command.

PURPOSE The **FM** command defines the format for footnotes — that is, the margins, tabs, indent paragraph, line spacing and offset. You use FM when you want the footnote format to be different from the format of the text. If you don't use the FM command, all footnotes use the default format for text.

XyWrite supports up to 3 independent footnote sets. You can create a separate format for each set by putting the set number into the FM command — i.e., FM1, FM2 or FM3. Any set of footnotes that does not have an FM command takes the default text format.

ACTION **Setting the Format for Footnotes.**
To set up the format for footnotes in a document:

1. Place the cursor before the first footnote you want to affect. Typically, you would move to the top of the file:

   Press: Ctrl Home

2. Decide which footnote set you want to use. Let's use set 3:

   Type: F5 `fm3 lm=8,rm=65` ↵

Result: Footnotes in set 3 have a left margin of 8 and a right margin of 65.

NOTE #1 **Footnote Transition.** The FT (Footnote Transition) command defines the amount of extra space between footnotes. As with all vertical spacing commands, the amount of space is expressed in number of *lines* (1/6 inch) and can be a whole number or a fraction as small as hundredths (.01). The FT command must be issued as part of the FM command. For example:

F5 fm3 ft=.5 ↵

inserts 1/2 *line* of extra space between footnotes.

NOTE #2 **Superscript Footnote Numbers.** The SC (Superscript) command allows you to change the mode for printing footnote numbers from footnote mode to superscript mode. This command has two forms:

- sc=1 Prints footnote numbers in superscript mode
- sc=0 Prints footnote numbers in footnote mode

The SC command must be issued as part of the FM command. For example:

F5 fm3 sc=1 ↵

If you omit the SC command, footnote numbers print in footnote mode.

NOTE #3 **Bottom Footnote.** You can include the BF (Bottom Footnote) command as part of the FM command. See "Bottom Footnote" earlier for more information.

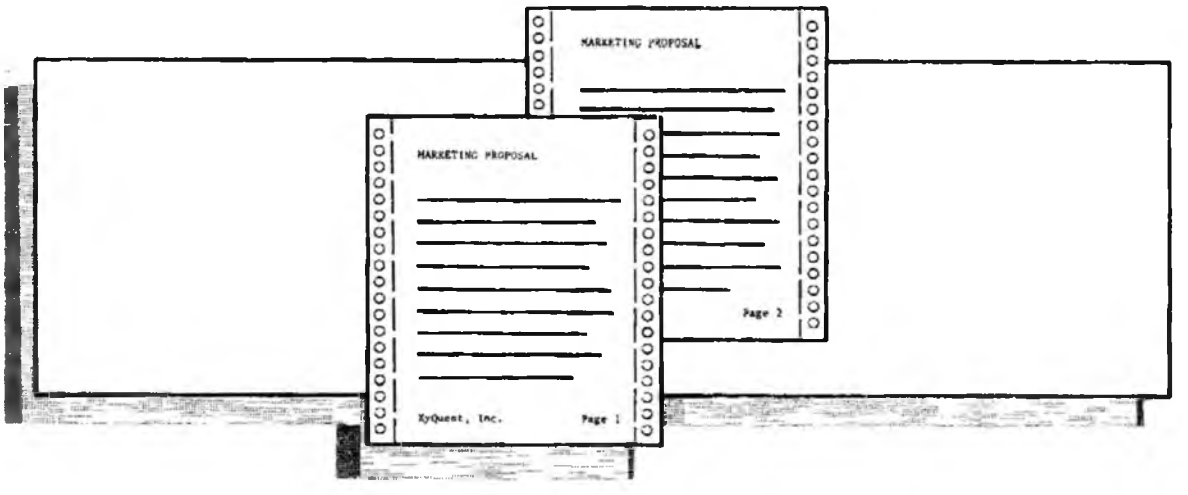
NOTE #4 **Effect of FM on Footnote Separators.** The FM command sets the format for the footnote (FN) text only *without* affecting the format of the footnote separators (FS).

INTRO

The term *running header* refers to text that appears at the top of a page and repeats on successive pages. Similarly, *running footer* refers to text that repeats at the bottom of pages. You can create headers and footers that:

- Number your pages for you
- Title your pages repeatedly on each page

CONTENTS



FORMAT

CM RH	Running Header, All Pages
CM RHE	Running Header, Even Pages
CM RHO	Running Header, Odd Pages
CM RF	Running Footer, All Pages
CM RFE	Running Footer, Even Pages
CM RFO	Running Footer, Odd Pages

These are embedded commands.

PURPOSE

The **RH** (Running Header) command inserts at the top of every page a block of text that you specify. Similarly, the **RF** (Running Footer) command inserts text at the bottom of every page. As text you can include chapter title, page number, date, and even the time. (XyWrite automatically updates the date and time.)

You specify the running text once and it automatically repeats page after page. You can start the text on any page and change or remove it on any successive page.

To have the same text on every page you would use the commands RH or RF. In order to alternate text on odd and even pages, you use pairs of commands: both RHE and RHO or RFE and RFO.

ACTION

Inserting a Running Header or Footer.

Running Headers and Running Footers are both inserted the same way. As an example, we will enter the same Running Header for all pages — the text includes a chapter title and page number:

1. Move the cursor to the beginning of the page where you want the Running Header to start, ahead of all text and spaces. (See Note #3.)
2. Type: F5 rh ↵ (to open a window)
3. Enter any formatting commands that you want for the Running Header. These formatting commands will not affect the body of the document. For example, to center the running header on every page:

   Type: F5 fc ↵
4. Now enter the Running Header text — for example, the chapter title and page number:

   Type: Chapter 1 without pressing ↵

Tab or space over to near the right margin and then type in the page number command (PN):

Type: F5 pn ↵

5. Enter a carriage return for each blank line you want between the text of the Running Header and the text of the document. Let's say you want two blank lines:

   Type: ↵ ↵

6. Close the screen:

   Press: F3

Result: "Chapter 1" and the page number are centered on the same line, followed by two blank lines, at the top of this page and all succeeding pages.

NOTE #1 **Other Page Number Formats.** The PN command alone puts in normal page numbers, starting at 1, 2, etc. But XyWrite can start at any number and create many formats. See "Page Numbering" and "Numbering" sections later.

NOTE #2 **Embedded Commands.** The RH and RF commands are embedded commands — they are embedded in text, visible only as triangles in Normal Display. In Expanded Display they are shown expanded — for example: «RHAThis is a running header».

If you are using the RH (or RF) command to specify the same running text on all pages, the expanded display appears as RHA (or RFA). If you edit while in the Expanded Display, never omit the third letter of the command — XyWrite understands «RHA» and «RFA», but not «RH» and «RF».

NOTE #3 **Where Running Text Takes Effect.** The Running Footer (RF) command takes effect at the bottom of the page on which it is entered. The Running Header (RH) command takes effect on the current page only if it is on the first line of that page, ahead of any text or spaces. (It can be preceded by other embedded commands.) Anywhere else on the page will cause it to take effect at the top of the next page.

NOTE #4 **Preventing an Extra Blank Page with a Running Header From Being Printed.** *Symptom*: At the end of your document your printer prints an extra page with only a running header or footer (and no other text) on it. *Cause*: You are probably ending your document with extra carriage returns which are forcing a new page, or with a PG (Page Break) command followed by more than one carriage return. *Remedy*: Remove the last PG command and any unnecessary carriage returns from the end of the file. You can also turn on the P-L indicator (with Shift F9) to preview the page numbers.

NOTE #5 **Viewing the Running Header and Running Footer.** In order to preview the running header and footer on-screen as they will be printed, you can use TYPES (Type to Screen).

NOTE #6 **Counters in the Header or Footer.** XyWrite has an automatic numbering capability (called a counter), which is described later in this chapter. You can insert the number currently stored in a counter into the text of a Running Header or Footer without increasing the value of the counter. See "Counter Command."

NOTE #7 **Entering a One-Line Header or Footer.** You can type in a one-line header by typing the text on the Command Line after the command. This is a shortcut for the method shown under Action. For example:

Type: F5 `rh This is a running header` F9

NOTE #8 **Editing Headers and Footers.** You can edit the text of headers and footers by switching to Expanded Mode (Ctrl F9) or by moving the cursor to the embedded command triangle and pressing Ctrl F3.

ALSO SEE **Page Format.** To see how the Running Header and Footer interact with other page format commands such as PL (Page Length) and FN (Footnotes), refer to the section "Page Length Procedures."

FORMAT		
	CM **PN**	Page Number
	CM **FP**	Final Page
	PN and FP are embedded commands.	

PURPOSE

The **PN** (Page Number) command inserts the page number in the text. The page number starts at 1 with the first page of a document and automatically increments for successive pages. You can use SP (Set Page Number) to change the starting page number and the numbering style (i.e., i, ii, iii... or a, b, c...).

FP (Final Page) inserts the number of the last page — useful for a "Page 5 of 7" format.

To automatically number the pages, insert the PN command in a running header or footer. The procedure "Inserting a Running Header or Footer" earlier in this section shows how. When you place PN (or FP) directly in the text (rather than in a running header or footer), it displays the current (or final) page number just once, at that point.

ACTION

Numbering the Pages.
The procedure to insert page numbering in a document is given in "Inserting a Running Header or Footer."

ACTION

Numbering Pages in the "Page ... of ..." Format.
To insert "Page ... of ..." style in a header, you must insert a Running Header (RH) command:

1. Move the cursor to the top of the page where you want the Running Header to start.
2. Type: F5 rh ↵
3. Type: Page F5 pn ↵
   Type: of (include a space after "of")
   Type: F5 fp ↵
4. Type: F3 (to close the header)

Result: Printed page numbers look like "Page 1 of 8," "Page 2 of 8" and so on.

NOTE #1 **Changing the Starting Number.** If you want pages in your document to start at a number other than 1, use the SP command described next.

NOTE #2 **Determining the Page Number.** To tell what the current page number is:

Press: Shift F9

Result: The current Page-Line numbers show at the top of the screen. This number adjusts for any change of page numbers due to an SP command in the text.

NOTE #3 **Embedded Command.** PN and FP are embedded commands — they are embedded in text, visible as triangles in Normal Display. In Expanded Display they are shown expanded as «PN» and «FP».

FORMAT **CM SP** *n* — Set Page Number

n is the starting page number
SP is an embedded command.

PURPOSE

The **SP** (Set Page Number) command sets the starting page number and defines the series as numerals, letters or symbols. You place the SP command on the *first line* of the page you want to affect, ahead of the embedded Page Number command (PN).

There are six different styles from which you can choose:

Decimal numbers	**SP 1**
Uppercase roman numerals	**SP I**
Lowercase roman numerals	**SP i**
Uppercase letters	**SP A**
Lowercase letters	**SP a**
Defined string	**SP ****n*

If you do not specify an SP command, XyWrite uses decimal numbers starting at 1. If you want to start the sequence with another number, or in another of the styles shown above, use that value in the command. For example, the command "SP iii" tells XyWrite to number the pages in lowercase roman numerals and to start the sequence at "iii."

ACTION

Setting the Starting Page Number.
To define the kind of page number and/or the starting page number:

1. Move the cursor to the first line of the page you want to affect, ahead of any text or spaces. (SP is ignored if not on the first line.) Be sure the cursor is before the PN command.
2. Decide what style and what starting point you want to use. Let's use decimal numbers starting with number 5:
3. Type: F5 sp 5 ↵

Result: When printed, this page will be numbered page 5, the next page will be page 6, and so on. The PN command reflects this change of page numbering.

ACTION **Using Your Own Numbering Strings.**
You can enter a table in the Printer File that lists strings you want to use as page numbers. (See Note #1 to enter a string into the Printer File.) To use these strings in your document:

1. Move the cursor to the top of the page you want to affect.
2. Enter the SP command with an asterisk:

   Type: F5 sp * ↵

Result: Page numbers will be printed out as the strings defined in the Printer File, starting with the first string on the first page, the second string on the second page, and so on. If there is no Counter String Table, XyWrite uses an asterisk, and repeats it as many times as necessary: *, **, ***....

NOTE #1 **Entering the Counter String Table.** The Counter String Table you put in the Printer File defines the strings that can be used as page numbers. (You can have only one Counter String Table.) A string can be a single character or many. The format of the Counter String Table is:

```
cs:#
string 1
string 2
  .
  .
  .
string #
```

where # is the number of strings in the table. This table can also be used for footnote numbers and counters. For example:

```
cs:4                                    (4 lines of strings)
!
@
#%
$
```

Counter Strings are discussed further in "Define Counter," Note #3, later in this chapter.

NOTE #2

Recycling Number Strings. If you choose uppercase or lowercase letters (or the Counter String Table) as a numbering style, you could have fewer characters (or strings) available than pages in your document. Once XyWrite has gone through all the characters, it returns to the first character and inserts it twice, then the second one twice, and so on (aa, bb, cc...). The list is repeated as many times as necessary.

By putting just one character in the Counter String table (for example, +), you can get pages numbered with multiples of that character (such as +, ++, +++, ...).

NOTE #3

Page Numbers in Chained Files. You can have the Set Page Number command affect only the screen display of a document (that is, the P-L field of the header) and not the printed output. This is very useful if you are using chain printing — that is, if you are using TYPE @ to continue a sequence of page numbers across more than one file.

You would use this feature in all the files after the first. Then, if you added or subtracted pages from the first file, the page numbers in the subsequent files would still print out correctly.

To do this, insert a question mark (?) immediately before the page number style in all files following the first in a set to be chain printed. For example:

Type: F5 sp ?1 ↵

This command will display numerical page numbers, starting at 1, in the P-L field. However, it will have no effect on printed output, which will number continuously from the first file through the last — in the style set by the first file's SP command.

NOTES

INTRO

XyWrite's automatic numbering system, described in this section, lets you number your document in virtually any format found in print.

You can number chapters, paragraphs, sections, lists and outlines — all at the same time, if need be. All numbers stay in order, even when material is added, deleted or switched around.

Automatic numbering can use numbers (1, 2, 3), upper- or lowercase roman numerals (I, II, III, i, ii, iii) or letters (A, B, C, a, b, c), or even special numbering sequences that you create in advance.

You also can create references within your text, such as "Refer to Graph 6, Chapter 3, p. 112". While not precisely numbering, this function is intimately involved with automatic numbering; that is, you can link it to a counter or footnote. And it works the same way — as you move the referenced passage around during editing, XyWrite always keeps the reference accurate regarding counter or footnote number, page and chapter.

Even if you do only one kind of numbering, you might read this section just to discover how powerful the system is.

Like other sections, this one takes you through procedures in the first part, and then covers commands individually.

CONTENTS

PURPOSE

DC (Define Counter) and **C** (Counter) allow you to automatically number blocks of text — outlines, lists, chapters, sections, and paragraphs.

Numbering is a two-step process. First you use the DC command at the start of your document to establish the style of the numbers that you want to use. Then you use the C counter where you want to place the automatic numbers. Several examples of how you use these commands are shown in this section.

A special C counter, **C0** (C-zero), is designated for numbering chapters, and two commands — **SR CH** (Set Record Chapter) and **REC** (Reference Chapter) — give the chapter numbers of text marked with C0 counters. Two other commands, **REF** (Reference) and **REP** (Reference Page), also track the location of reference ("Refer to...") text, which you mark within a C counter or with **LB** (Label).

ACTION

Making a Simple Numbered List.
Let's start with the assumption that you have typed out a list and are ready to number it. To number the items shown in Example #2:

1. Move the cursor ahead of the list.

2. Type: F5 dc 1=(A) ↵

   Result: This embeds a format triangle in the document and defines counter 1 to start with the value A enclosed in parentheses.

3. Now insert the C1 counter in the text at each place you want the automatic numbers to occur.

   Type: F5 c1 ↵

Result: C1 takes (A) as its first value and increments with each use, as shown in Example #2. Each of the letters A through F is generated automatically by XyWrite.

ACTION **Automatic Numbering of Paragraphs.**
As an example, we will choose a style found in many legal and government reports — numbers separated by periods. We'll use Example #4 which follows to illustrate the point.

1. Move the cursor ahead of the list.

2. Define the numbering scheme that you want to use:

   Type: F5 dc 1=1.1.1 ↵

   Result: This defines three-level numbering; each level starts at 1. The 1 on the left side of the equal sign means you begin with C1.

ACTION **Making an Outline.**
By giving a DC (Define Counter) command for each counter, you can create an automatically numbered outline with many levels (I, A, 1, a, . . .). For example, to create a three-level outline:

1. Move the cursor to the top of the document.

2. Type: F5 dc 1=I A 1 ↵

   Result: This defines the top level.

3. Now enter each of these DC commands to ensure that each level of the outline resets all lower levels.

   Type: F5 dc 2=A 1 ↵

   Type: F5 dc 3=1 ↵

4. Now enter the C1, C2 and C3 counters in the list, indented as shown in Example #5.

ACTION **Automatic Numbering of Chapters.**
Only one C counter, **C0**, is used for automatic numbering of chapters, although it also can be used for other numbering. Commands to display the current chapter number look specifically for C0. To initiate automatic roman numeral chapter numbers, for example:

1. Move the cursor to the top of the document.
2. Type: F5 dc 0=I ↵
3. Move the cursor to start of the chapter title.

   Type: F5 c0 ↵
4. Repeat Step 3 for each chapter title.

Result: The first chapter will be numbered I, the second II, and so on.

NOTE #1 **Page Numbering in Chapter-Page Format.** The C0 counter enables you to do automatic page numbering in the popular chapter-page format, as is done in this manual. If you wanted to combine roman numeral chapter numbers (shown above) with page numbers, you would create a running header or footer (see "Inserting a Running Header or Footer" earlier in this chapter). In place of the PN (Page Number) command in a header or footer, do the following:

1. Type: F5 c0 ↵
2. Type: Alt Ctrl 45 (hard hyphen)
3. Type: F5 pn ↵

Result: Your finished document will contain page numbers that look something like this: II-43 (A counter in a header or footer will not throw off the count. It merely reads the current value of the counter in text.)

NOTE #2 **Chapter Numbering in More than One File.** Chapter numbering works even if your chapters are in several files. When you link files for chain printing with TYPE @ (or for display with TYPES @), chapters are numbered consecutively — as if they all were in one file. (See Chapter 2 for more on TYPE @ and TYPES @.)

Using the example from Note #1, to number chapters consecutively in a set of files:

1. Put DC 0=I *only* in the *first* chapter file displayed or chain printed.
2. When you create the parent file containing filenames to be displayed or printed, be sure to list them as a *set*; that is, without a [↵] between the filenames. In this form, they are treated as one document.

NOTE #3 **Counter Format for Chain Files.** *All* counters continue through a chained set of files, unless they are reset with a DC command in one of the files. But if you work with an individual file without DC commands in it, all its counters take on the 1, 2, 3 format.

To solve this, you can put a modified DC command in each file for each counter — one that will show the desired format, yet increment properly in a chain file. Simply place a question mark (?) before the DC format in all but the first file. (The "?" means the final value is currently unknown.)

For example, we've already put the DC 0=I in the first of our chain files to count chapters in roman numerals (see Note #1). Now, at subsequent chapter titles,

Type: [F5]dc 0=?I[↵]

Result: Subsequent files, displayed separately, will increment starting at roman numeral one (I); when chain printed or displayed, they will be consecutively numbered: I, II, III, IV, and so on.

NOTE #4 **Chapter-Page Numbering in the Index.** XyWrite provides a command, **SR CH**, that captures in your index the chapter-page number format of your document. It is covered in "Table of Contents & Index," Chapter 5.

NUMBERING

EXAMPLE #1

(No DC statement)

PRINTOUT

```
1  Illinois
2  Massachusetts
3  Washington
4  California
5  Michigan
6  Florida
```

EXPANDED DISPLAY

```
<C1> Illinois
<C1> Massachusetts
<C1> Washington
<C1> California
<C1> Michigan
<C1> Florida
```

EXAMPLE #2

<DC1=(A)>

```
(A)  Illinois
(B)  Massachusetts
(C)  Washington
(D)  California
(E)  Michigan
(F)  Florida
```

```
<C1> Illinois
<C1> Massachusetts
<C1> Washington
<C1> California
<C1> Michigan
<C1> Florida
```

EXAMPLE #3

<DC1=A-1>

```
A  Illinois
   A-1  Chicago
   A-2  Springfield
B  Massachusetts
   B-1  Boston
   B-2  Cambridge
```

```
<C1> Illinois
     <C2> Chicago
     <C2> Springfield
<C1> Massachusetts
     <C2> Boston
     <C2> Cambridge
```

EXAMPLE #4

<DC1=1.1.1>

```
1  Animals
   1.1  Birds
        1.1.1  Albatross
        1.1.2  Penguin
   1.2  Mammals
        1.2.1  Monkey
        1.2.2  Lion
2  Plants
   2.1  Trees
        2.1.1  Oak
        2.1.2  Magnolia
   2.2  Flowers
        2.2.1  Daisy
        2.2.2  Rose
```

```
<C1>  Animals
      <C2>  Birds
            <C3>  Albatross
            <C3>  Penguin
      <C2>  Mammals
            <C3>  Monkey
            <C3>  Lion
<C1>  Plants
      <C2>  Trees
            <C3>  Oak
            <C3>  Magnolia
      <C2>  Flowers
            <C3>  Daisy
            <C3>  Rose
```

NUMBERING

	PRINTOUT	EXPANDED DISPLAY
EXAMPLE #1		(No DC statement)
	1 Illinois	≪C1≫ Illinois
	2 Massachusetts	≪C1≫ Massachusetts
	3 Washington	≪C1≫ Washington
	4 California	≪C1≫ California
	5 Michigan	≪C1≫ Michigan
	6 Florida	≪C1≫ Florida
EXAMPLE #2		≪DC1=(A)≫
	(A) Illinois	≪C1≫ Illinois
	(B) Massachusetts	≪C1≫ Massachusetts
	(C) Washington	≪C1≫ Washington
	(D) California	≪C1≫ California
	(E) Michigan	≪C1≫ Michigan
	(F) Florida	≪C1≫ Florida
EXAMPLE #3		≪DC1=A-1≫
	A Illinois	≪C1≫ Illinois
	A-1 Chicago	≪C2≫ Chicago
	A-2 Springfield	≪C2≫ Springfield
	B Massachusetts	≪C1≫ Massachusetts
	B-1 Boston	≪C2≫ Boston
	B-2 Cambridge	≪C2≫ Cambridge
EXAMPLE #4		≪DC1=1.1.1≫
	1 Animals	≪C1≫ Animals
	1.1 Birds	≪C2≫ Birds
	1.1.1 Albatross	≪C3≫ Albatross
	1.1.2 Penguin	≪C3≫ Penguin
	1.2 Mammals	≪C2≫ Mammals
	1.2.1 Monkey	≪C3≫ Monkey
	1.2.2 Lion	≪C3≫ Lion
	2 Plants	≪C1≫ Plants
	2.1 Trees	≪C2≫ Trees
	2.1.1 Oak	≪C3≫ Oak
	2.1.2 Magnolia	≪C3≫ Magnolia
	2.2 Flowers	≪C2≫ Flowers
	2.2.1 Daisy	≪C3≫ Daisy
	2.2.2 Rose	≪C3≫ Rose

NUMBERING

EXAMPLE #5
(Properly constructed DC command)

PRINTOUT

```
I  Animals
   A  Birds
      1  Albatross
      2  Penguin
   B  Mammals
      1  Monkey
      2  Lion
II Plants
   A  Trees
      1  Oak
      2  Magnolia
   B  Flowers
      1  Daisy
      2  Rose
```

EXPANDED DISPLAY

```
«DC1=I A 1»
«DC2=A 1»
«DC3=1»

«C1» Animals
     «C2» Birds
          «C3» Albatross
          «C3» Penguin
     «C2» Mammals
          «C3» Monkey
          «C3» Lion
«C1» Plants
     «C2» Trees
          «C3» Oak
          «C3» Magnolia
     «C2» Flowers
          «C3» Daisy
          «C3» Rose
```

EXAMPLE #6
(Improperly constructed DC command)

```
I  Animals
   A  Birds
      1  Albatross
      2  Penguin
   B  Mammals
      3  Monkey
      4  Lion
II Plants
   C  Trees
      5  Oak
      6  Magnolia
   D  Flowers
      7  Daisy
      8  Rose
```

```
«DC1=I»
«DC2=A»
«DC3=1»

«C1» Animals
     «C2» Birds
          «C3» Albatross
          «C3» Penguin
     «C2» Mammals
          «C3» Monkey
          «C3» Lion
«C1» Plants
     «C2» Trees
          «C3» Oak
          «C3» Magnolia
     «C2» Flowers
          «C3» Daisy
          «C3» Rose
```

EXAMPLE #7
(Chapter-Page No.)

```
II-13
```

```
«DC0=I»

«RFA«C0»-«PN»»
```

NUMBERING

EXAMPLE #5
(Properly constructed DC command)

PRINTOUT

```
I  Animals
   A  Birds
      1  Albatross
      2  Penguin
   B  Mammals
      1  Monkey
      2  Lion
II Plants
   A  Trees
      1  Oak
      2  Magnolia
   B  Flowers
      1  Daisy
      2  Rose
```

EXPANDED DISPLAY

```
«DC1=I A 1»
«DC2=A 1»
«DC3=1»

«C1»  Animals
      «C2»  Birds
            «C3»  Albatross
            «C3»  Penguin
      «C2»  Mammals
            «C3»  Monkey
            «C3»  Lion
«C1»  Plants
      «C2»  Trees
            «C3»  Oak
            «C3»  Magnolia
      «C2»  Flowers
            «C3»  Daisy
            «C3»  Rose
```

EXAMPLE #6
(Improperly constructed DC command)

```
I  Animals
   A  Birds
      1  Albatross
      2  Penguin
   B  Mammals
      3  Monkey
      4  Lion
II Plants
   C  Trees
      5  Oak
      6  Magnolia
   D  Flowers
      7  Daisy
      8  Rose
```

```
«DC1=I»
«DC2=A»
«DC3=1»

«C1»  Animals
      «C2»  Birds
            «C3»  Albatross
            «C3»  Penguin
      «C2»  Mammals
            «C3»  Monkey
            «C3»  Lion
«C1»  Plants
      «C2»  Trees
            «C3»  Oak
            «C3»  Magnolia
      «C2»  Flowers
            «C3»  Daisy
            «C3»  Rose
```

EXAMPLE #7
(Chapter-Page No.)

II-13

```
«DC0=I»

«RFA«C0»-«PN»»
```

Refer to the previous two pages of examples. These should give you a good idea of how to use the DC commands and C counters.

Example #1. This is the simplest numbering scheme. If you give no DC command, all counters assume the simple numbers 1, 2, 3, . . .

Example #2. With DC 1=A, the C1 counter numbers the items alphabetically.

Example #3. With DC 1=A-1, the C1 counters increment with A, B, C, and the C2 (second-level counters) take on two-part numbers (such as A-1). Notice the hyphen specified in the DC command appears in the numbering.

Example #4. With DC 1=1.1.1, the C1 counters are one-part, the C2 counters are two-part and the C3 counters are three-part. Notice how the DC command specifies the form for the lowest-level number — three numbers separated by periods. Also notice that a period appears only when the number to its right appears.

Example #5. The three DC commands are necessary here to produce the result shown. The first DC command ensures that C1 resets the C2 and C3 levels. The second DC command ensures that C2 resets the C3 level. (Compare with Example #6.)

Example #6. These DC commands are improperly constructed. Notice how the numbering is A, B, C, D — the second C1 did not cause C2 to reset to A. This demonstrates why it is necessary to follow the DC command in Example #5.

Example #7. Use of the C0 counter to create a chapter-page number (described in Note #1, earlier) is shown in this footer. This example assumes a DC 0=I command has been put at the top of the document, as described in previous section, and that a C0 counter precedes each chapter title.

NOTE #5 **Default Counter Definitions.** If you leave a counter undefined, it defaults to a starting number of 1.

NOTE #6 **Count Sequence for Letters.** Letters increment from A through Z. After Z comes AA, BB, and CC through ZZ. The sequence continues with AAA, BBB, CCC, etc. The same pattern holds for lowercase letters.

NOTE #7 **Range of Roman Numerals.** The sequence of roman numerals starts with I (the value 1) and goes through 64,000.

NOTE #8 **Initial Values for Letters.** The initial value of any letter definition is the letter you specify. However, if that letter can also mean a roman numeral then it is taken as a roman numeral. Letters used as roman numerals are I, V, X, L, C, D and M.

If you want to start a list with the letter L (rather than the roman numeral fifty, which is also L), then you precede the letter with a double quote mark:

```
CMdc 1="L
```

NOTE #9 **Punctuation.** Punctuation you specify in the DC command does not appear at the *end* of the numbers (except as noted below) — it appears only between numbers. In Example #4, if you wanted a period to follow the numbers on both of the C1 lines (such as 1. and 2.), you would insert them manually after the C1 triangles.

Every rule has an exception. Four characters are permitted to be printed immediately after a number: The closing parenthesis ')', square bracket ']', curly brace '}' and angle bracket '>'. This lets you make definitions that include numbers or letters in parentheses.

Any punctuation can appear *ahead* of the first number (*n1*). Example #2 demonstrates an open parenthesis being used in the DC command ahead of the A. Also note that you are not limited to only one character of punctuation between *n1* and *n2*, for instance — there is no limit.

NOTE #10 **Setting Up a Six-Level Outline.** To set up a six-level outline, enter the following DC commands. The DC 2 through DC 6 commands ensure that each level of the outline resets all lower levels.

F5 dc 1=I A 1 a(1)(a) ↵
F5 dc 2=A 1 a(1)(a) ↵
F5 dc 3=1 a(1)(a) ↵
F5 dc 4=a(1)(a) ↵
F5 dc 5=(1)(a) ↵
F5 dc 6=(a) ↵

Notice there is at least one punctuation mark between numbers — either a space or parenthesis. The resulting outline would have the six levels:

I A 1 a (1) (a)

NOTE #11 **Numbering Within Numbered Sections.** You may want to run an independent numbered list within a section that is numbered — say, a numbered outline within a section which is already numbered. There are 10 counters, starting with 0 and ending with 9. If you use the lower counters, say 0 through 5, for your automatic chapter and section numbering, you can use counters 6 through 9 for numbering within a section.

NOTE #12 **Assigning the Counter Commands to the Keyboard File.** You can assign the commands C0, C1, C2, and so on, along with tabs (or spaces), to individual keys. Of course you still need to set the proper TS settings for the tabs to work. Refer to the note in the Counter Command section which follows for details.

NOTE #13 **Creating References.** The "Reference Commands" section which follows describes how to set up references.

FORMAT

CM **DC #=*n*** (for simple numbering)
CM **DC #=*n1.n2.n3.n4.n5.n6.n7.n8.n9.n10*** (for nested numbering)

is the level you are defining (0 - 9).
n determines the style and starting value for the counter
n1 determines the style and starting value for the first level (#) of a nested numbering system
n2 determines the style and starting value for the second level (# + 1) of a nested numbering system
...and so on up to the last counter used (10 levels max).
. (period) represents any punctuation (one or more characters) you want to appear between the numbers.
DC is an embedded command.

EXAMPLE

CM dc 1=I A 1

PURPOSE

DC (Define Counter) and **C** (Counter) allow you to automatically number blocks of text — chapters, paragraphs, sections, lists, and outlines. Using counters is a two-step process. First you use the DC command at the start of your document to establish the style of the numbers that you want to use. Then you use the C counter where you want to place the automatic numbers.

ACTION

Using the DC Command.
Refer to the earlier section "Automatic Numbering" for the procedures on how to use the DC command.

NOTE #1

Defining the Styles for Numbers. You can use DC to define up to ten levels; you define a style for each level. The five basic styles (with their typical starting values shown) are:

Numeric	1
Uppercase roman numerals	I
Lowercase roman numerals	i
Uppercase letters	A
Lowercase letters	a

A sixth style, which lets you create your own sequence, also is available. (See Note #3).

NOTE #2

A Closer Look. When you're setting up a nested numbering scheme, such as an outline or sub-paragraphs (i.e., 1.2.2), the values you give with the DC command (# and *n1*, *n2*, *n3*, . . .) correspond to the counters as follows:

DC 0=*n1*	Defines C0
DC 0=*n1.n2*	Defines C0 and C1
DC 0=*n1.n2.n3*	Defines C0, C1, and C2

and so on.

DC 1=*n1*	Defines C1
DC 1=*n1.n2*	Defines C1 and C2
DC 1=*n1.n2.n3*	Defines C1, C2, and C3

and so on.

DC 8=*n1*	Defines C8
DC 8=*n1.n2*	Defines C8 and C9

No other levels can be specified with DC 8.

If any level is left unspecified, it takes on the default value of 1, as the earlier Example #1 demonstrates.

NOTE #3

Creating Your Own Set of Counters. You can create a totally original numbering set with XyWrite. Say, for a special project, you want to number footnotes, pages or sections with the series α, β, Γ, π, Σ. You'd place a Counter String table CS:*n* (*n* is the number of strings in the table), in the Printer File as follows (for more on the Printer File, see Chapter 6):

```
cs:5
α
β
Γ
π
Σ
```

Be sure to store and load the Printer File. Then, when you set the number in the document, use the appropriate number-setting command with an asterisk. For example

```
dc 1=*                    (to define a counter)
```

If you *don't* want numbers to start with the first symbol in the string, add a number for how far down to start. To start page numbering at the second symbol, for instance, use dc 1=*2. The second time through a series, each character is doubled; the third time, tripled; and so on: αα, ββ, ΓΓ,...; ααα, βββ, ΓΓΓ,....

FORMAT	CM C# - Counter
	# is the counter number (0-9) C0 is used for chapter numbering - (optional) lets you use the current counter value without incrementing the counter. C0 - C9 are embedded commands.
EXAMPLE	CM c1

PURPOSE

The **C0** through **C9** (Counter) commands insert the current value of the counter into the text.

The DC (Define Counter) command defines a set of counters C0 through C9. Through proper selection, you can do paragraph, section, chapter and outline numbering as well as the simple numbering of lists.

In its simplest use, numbering a list of things, you can use C0 through C9 without giving a DC command. Refer to Example #1 in the previous section "Automatic Numbering Procedures."

ACTION

Using the Counter Command.
Refer to the earlier section "Automatic Numbering Procedures" for procedures on how to use the C0 - C9 commands.

NOTE #1

Using the Current Count Twice. Sometimes you need to make use of the same automatic number in more than one place. For instance, you might want to refer to the section number in the text itself:

Section 3. In Section 3 we discuss . . .

Let's say you're numbering sections with counter C1. To repeat the section number in the text, use the hyphen (-) option of the C counter (note the space between the counter number and the hyphen):

Press: F5 c1 -↵

In Expanded Display the line appears:

Section «C1». In Section «C1-» we discuss . . .

NOTE #2 **Using the Current Count in Headers and Footers.** You can refer to the current count in headers and footers using the normal C0-C9 commands. They will be printed without incrementing the automatic numbering in the text.

NOTE #3 **Putting the Counter Function Calls in the Keyboard File.** You can set up your keyboard to simplify the insertion of automatic numbers. The following example assigns *tabs* ^O and commands C0 - C9, one level of counter to each individual key: Ctrl Shift 1 for C1, Ctrl Shift 2 for C2, and so on. (GT means "Go To Text Area" in case the cursor starts out on the Command Line.) Entries into the keyboard table would look like this:

```
TABLE=CTRL+SHIFT

2=GT,,←
3=GT,^O,c2,,←
4=GT,^O,^O,c3,,←
5=GT,^O,^O,^O,c4,,←
etc.
```

Result: After you load the Keyboard File with the LDKBD command, press Ctrl and Shift together with the numbers along the top of the keyboard to enter the desired level of counter C1, C2, and so forth.

NOTE #4 **Chapter Counter.** Always use counter C0 to do automatic chapter numbering. C0 is the counter referenced by the REC (Reference Chapter) command, described later in this chapter, and tracked for indexing by the SR CH (Set Record Chapter) command, described in "Table of Contents & Index," Chapter 5. If you are not doing chapter numbering, you can use C0 for any other purpose—numbering paragraphs, lists, etc.

NOTE #5 **Line Breaks.** Because the number stored in the counter is not displayed on the screen, XyWrite must estimate its width in order to calculate line breaks. XyWrite allows a width equal to 5 times the width of the character 0 (zero). If the width of the counter value exceeds that width, the line containing the counter may extend beyond the right margin when you print the document. To increase the space reserved for counter values, change the FU setting in the Printer File. Refer to Chapter 6 for more details.

FORMAT	CM **REP** *label*	Reference to Page Number
	CM **REC** *label*	Reference to Chapter Number
	CM **REF** *label*	Reference to Counter Number
	CM **LB** *label*	Label
	label is the name given to identify the reference text REP, REC, REF and LB are embedded commands	
EXAMPLE	CM rep bargraph	

PURPOSE

Suppose you want to write "See Chapter 7, Section 4, 'The War Years,' p. 361"; but you aren't sure The War Years will stay on page 361 — or will even stay in Chapter 7. With XyWrite you *label* the part called The War Years, then use the **REP**, **REC** and **REF** commands to refer to the label. You can use these commands to automatically update all references to page numbers, chapter numbers, paragraph numbers, footnotes numbers, heading numbers, or any other counters C0 - C9.

Each procedure for creating references has two parts:

- Part I. **Label the Text** — The label is a unique name which tags the passage so you can track its page, chapter, footnote and counter number.
- Part II. **Use the Reference Commands** — The REF, REP and REC commands are placed in your referral statement. They indicate where the reference text is.

We use the term *referral statement* to mean any statement such as "See page 6" which refers to labeled text.

We illustrate referencing with the following options, each of which requires Parts I and II listed above.

- **Referring to the Page Number of Any Text.**
  (Option 1a) You mark the text with an LB (Label) command and use REP (page) to refer to that text.
- **Referring to Chapter & Page Numbers of Any Text.**
  (Option 1b) You mark the text with LB, and use REC (chapter) and REP (page) in the referral statement.
- **Referring to a Counter.**
  (Option 2) You can attach a label to a counter, such as an illustration number or section number. REF in the referral statement will yield the current counter number, REP the page number and REC the chapter number (if chapter counter C0 is used).

- **Referring to a Footnote Number.**
  (Option 3) You can put the LB command in a footnote. You then put REF in the referral statement to produce the footnote number, REP its page number and REC the chapter number (if a chapter counter C0 is used).

Each option is discussed below as an action, divided into Parts I and II.

ACTION *(Option 1a)*

Referring to the Page Number of Any Text.
The simplest reference is to a page; to track it, you put an LB (Label) command in the text you are referring to and a REP command at the referral statement ("See page ..."):

PART I

Label the Text — Use the LB command.

1. Go to the page which contains the text you're referring to, say page 6.
2. Put the cursor at the beginning of the reference text.
3. Let's say this paragraph you want to refer to pertains to travel, so let's use that as a label:

   Type: F5 lb travel ↵

Result: The text is now labeled; a triangle is embedded in the text. In Expanded Display it looks like:
«LBtravel»

PART II

Use the Reference Command — Elsewhere in your text, create the referral statement "See page ...":

1. Move to the page where you want to refer to the text.
2. Type: See page (add a space after "page")
3. Type: F5 rep travel ↵

Result: The referral statement includes an embedded triangle where the page number goes.

See page «REPtravel» (Expanded Display)
See page ▲ (Normal Display)
See page 6 (as printed out with TYPES or TYPE)

ACTION *(Option 1b)*

Referring to the Chapter & Page Numbers of any Text.

If you are numbering pages with a chapter-page format (as this manual does), you need referral statements that track the chapter number as well as page number.

To begin with, you must have a chapter counter (C0) inserted at each chapter title (Automatic Numbering Procedures earlier tells how to use the Chapter Counter).

PART I

Label the Text — Use the LB command exactly like Option 1a.

PART II

Use the Reference Commands — Use the REC (chapter) and REP (page) commands:

1. Move to the page where you want to refer to the text.
2. Type: See page (add a space after "page")
3. Type: F5 rec travel ↵ (chapter no.)
4. Type: Alt Shift 45 (hard hyphen)
5. Type: F5 rep travel ↵ (page no.)

Result: The referral statement includes an embedded triangle where the page number goes:

See page «RECtravel»-«REPtravel» (Expanded Display)
See page ▲-▲ (Normal Display)
See page 4-16 (printed out with TYPES or TYPE)

ACTION *(Option 2)*

Referring to a Counter.

Let's say you want to refer the reader to a title "Section D, Five-Year Plan." You've used counter C2 for the section letter (which may not stay section "D").

PART I

Label the Counter — Add a name to the existing counter command, in this case C2 (rather than using the LB command).

1. Move the cursor to the triangle representing the counter command C2.
2. You'll replace the counter command, so delete it:

   Press: Del
3. Let's label this text "plan".

   Type: F5 c2 plan ↵

Result: The referral statement would look like:

Section «C2plan». Five-Year Plan	(Expanded Display)
Section ▲. Five-Year Plan	(Normal Display)
Section D. Five-Year Plan	(Printed out)

PART II

Use the Reference Commands — To write a phrase like "See Section D, Five-Year Plan" elsewhere, with the current section letter, you use REF to track the labeled counter "C2plan."

1. Move to the page where you want to refer to the text.
2. Type: See Section (add space after line)
3. Type: F5 ref plan ↵
4. Finish the statement:

   Type: , Five-Year Plan (note the comma)

Result: The referral statement includes an embedded triangle where the page number goes:

See Section «REFplan», Five-Year Plan	(Expanded Display)
See Section ▲, Five-Year Plan	(Normal Display)
See Section D, Five-Year Plan	(Printed out)

ACTION *(Option 3)*

Referring to a Footnote Number.
To refer to a footnote number, such as "Refer to footnote 4," do the following:

PART I

Label the Footnote — Use an LB command to label the footnote, and REF to track the labeled footnotes.

1. Place the cursor where you want the footnote number to appear in your text.
2. Create the footnote with the FN command:

   Type: [F5]fn[↵]
3. The *first thing* in your footnote must be a label — for example, "authors":

   Type: [F5]lb authors[↵]
4. Now write your footnote and close the screen:

   Type: The great American authors include Thoreau, Twain and Hemingway.

   Type: [F3]

Result: A footnote number appears in your text. In Expanded Display, it appear like:

«FN1«LBauthors»The great American authors...»

PART II

Use the Reference Commands — Use any of the REP, REF, and REC commands you want. In this case, we will refer to only the footnote number.

1. Move to the page where you want to refer to the text.
2. Type: Refer to footnote (add space at end)
3. Type: [F5]ref authors[↵]. (include the period)

Result: The referral statement has an embedded triangle where the footnote number goes. In Expanded Display it looks like this:

Refer to footnote «REFauthors»	(Expanded Display)
Refer to footnote ▲	(Normal Display)
Refer to footnote 4	(Printed out)

NOTE #1 **Combining Commands.** We started this section with the example, "See Chapter 7, Section 4, 'The War Years,' p. 361." See if you can create this reference by combining all three commands as described above. (The answer is given in the next paragraph.) In addition to the referral statement, 1) you need a C0 counter at chapter titles; and 2) the label itself goes with the section title, "The War Years" — either with an LB command or inside a C counter if you're using counters on section titles.

If your label is "war," the referral statement in Expanded Display should look like this:

> See Chapter «RECwar», Section «REFwar», 'The War Years,' p. «REPwar».

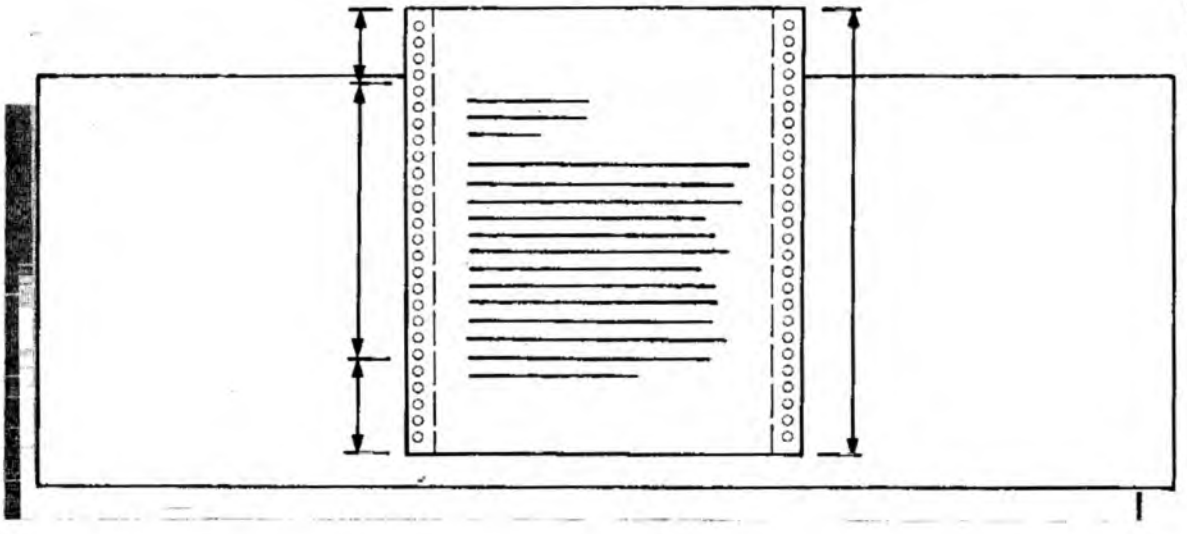
NOTE #2 **Labeling Existing Counters or Footnotes.** The previous examples inserted labels as you created the counter or footnote commands. You could simply insert a label within an existing counter or footnote command by moving the cursor to the embedded command (▲) and pressing Ctrl F3. This lets you edit the command quickly, without switching to Expanded Display with Ctrl F9.

NOTE #3 **Line Breaks.** Because the number stored in the Reference commands is not displayed on the screen, XyWrite must estimate its width in order to calculate line endings. XyWrite allows a width equal to five times the width of the character 0 for the value stored in the Reference command. If the width of the reference value exceeds that width, the line containing the reference may extend beyond the right margin when you print the document. To increase the space reserved for reference values, change the FU setting in the Printer File. Refer to Chapter 6 for more information.

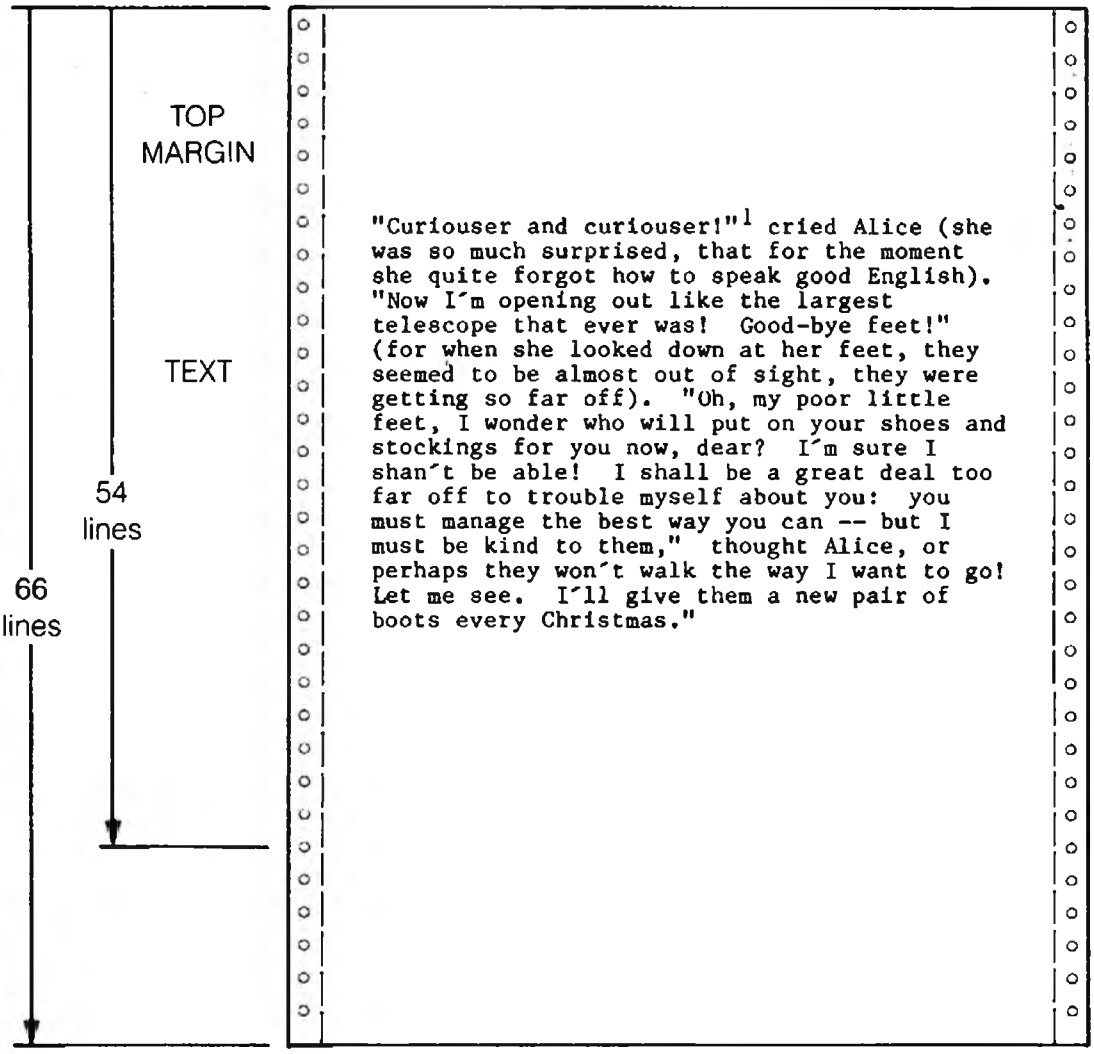
INTRO

In this section we have grouped together commands which affect the length of the printed page. Because these commands interact, we begin with a set of basic procedures which includes all of the essential settings.

CONTENTS



DEFAULT PAGE LENGTH SETTINGS



PURPOSE

XyWrite is preset for the simple format shown in the illustration on the facing page. The default settings allow you to set up a page quickly and simply for printing. The default settings include:

- 11″-long sheet of paper
- 6 lines per inch
- Single-spaced text
- 54 lines of text per page (which runs 9″ in length)
- When one line of a paragraph ends a page (orphan) or starts a page (widow), it will automatically be moved to join the rest of the paragraph

ACTION

Using the Default Page Length Settings

The only adjustment you need to make is the top margin:

1. Align the top edge of the paper with the printhead. (This is a good practice to follow — otherwise, any running footer you later add will likely fall at the top of the next page.)
2. To set a top margin an all pages of a document, enter the TP (Top Margin) command at the top of your document. Place it on the top line ahead of any text or spaces. This example creates a one-inch top margin:

   Type: F5 tp 6 F9

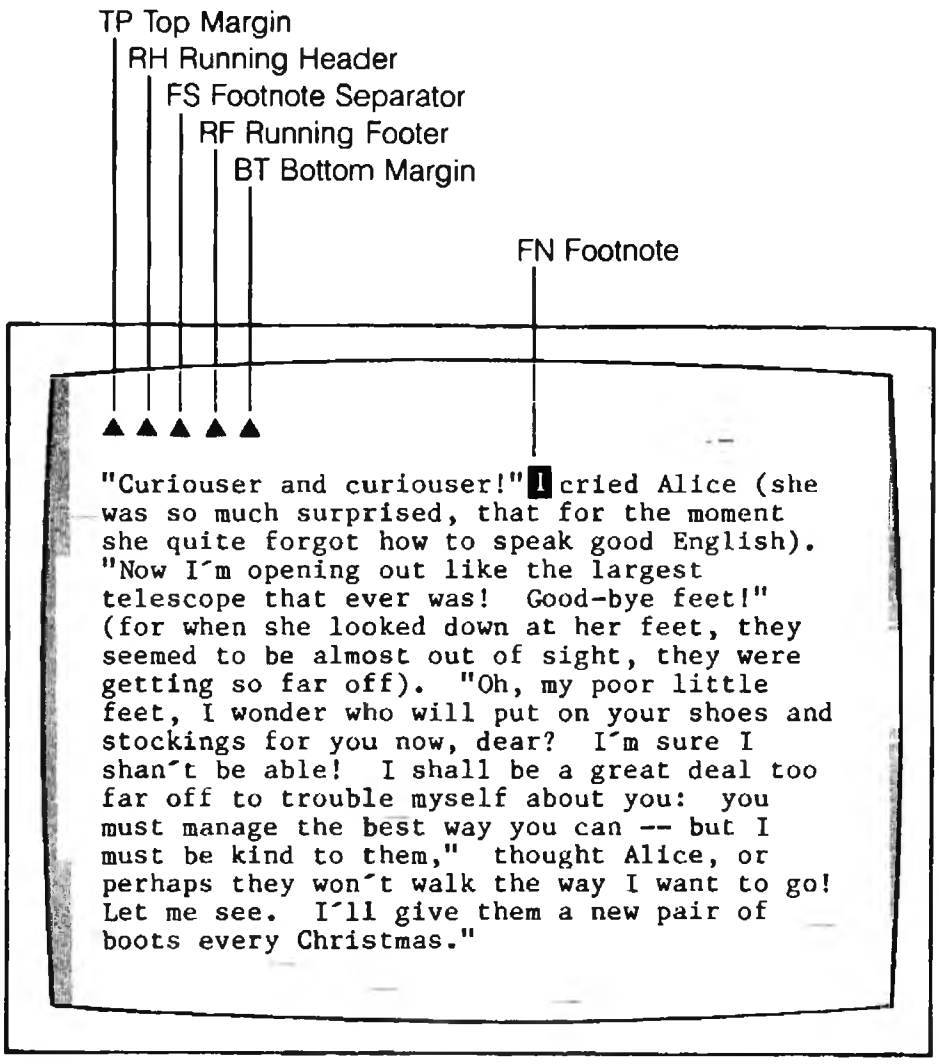
This method insures that the document will be printed with a one-inch margin at the top of every page. Refer to the Top Margin (TP) description later in this section for a few other ways to set the top margin.

Result: Your document is now set up for a one-inch top margin (which with these defaults leaves a one-inch margin at the bottom). When you give the TYPE command, the document is printed out with 54 lines to a page (give or take a line to eliminate widows and orphans). The next procedure shows you how to modify these page length settings.

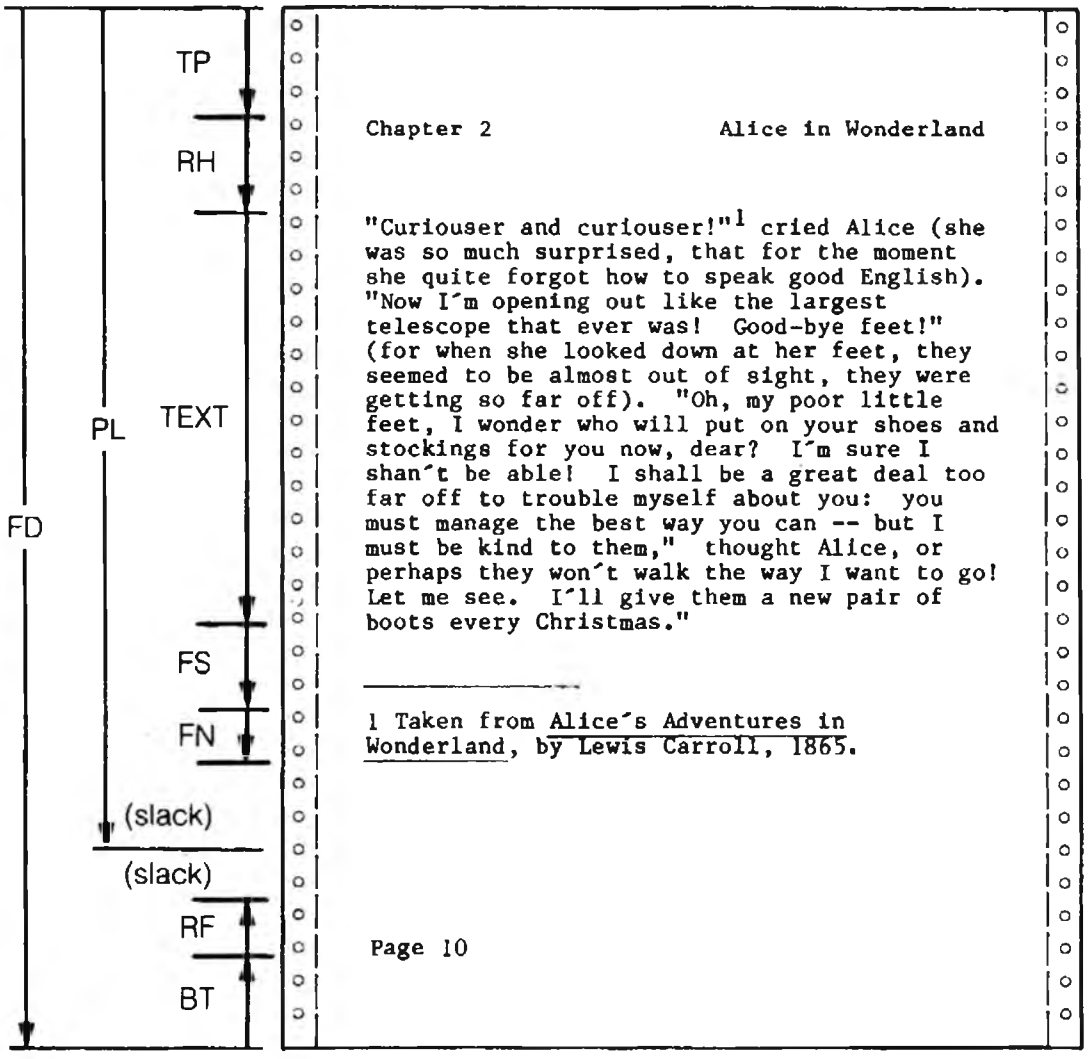
NOTE

Illustrations. The figures and list on the next three pages illustrate the page length settings at your control. The two figures compare how a document displays and prints out.

DISPLAY SHOWING PAGE LENGTH COMMANDS



PRINTOUT SHOWING PAGE LENGTH COMMANDS



PAGE LENGTH SUMMARY

FD - Form Depth. The number of lines in the overall length of the sheet of paper. The default is set to 66 lines, which corresponds to 11" x 6 lines/inch.

PL - Page Length. The number of lines from the top of a page all the way down to the last line of a footnote (if any). It includes the top margin, running header, body text, and footnotes, but not running footer or bottom margin.

TP - Top Margin. The number of lines left blank as a top margin on the page.

RH - Running Header. Text repeated page after page — can include page number, date, title, even the time of day. You can make the header the same on all pages, or different on even (left-hand) and odd (right-hand) pages.

Body Text. The main text of the document.

FS - Footnote Separator. Indicates the characters and blank lines that you want to appear between the body text and the footnotes. In this case, a row of underlines is used.

FN - Footnote. Footnotes for the page are inserted here. XyWrite will put the entire footnote on the page, if possible.

Slack. If the text is less than a full page in length, the blank lines appear between the footnote and running footer. (This is the default setting, for BF=0.)

RF - Running Footer. Text repeated at the bottom of each page. Running footers have all the features of running headers listed above. (Odd or even is optional).

BT - Bottom Margin. Blank lines left as the bottom margin on the page.

PAGE LENGTH SUMMARY

FD - Form Depth. The number of lines in the overall length of the sheet of paper. The default is set to 66 lines, which corresponds to 11" x 6 lines/inch.

PL - Page Length. The number of lines from the top of a page all the way down to the last line of a footnote (if any). It includes the top margin, running header, body text, and footnotes, but not running footer or bottom margin.

TP - Top Margin. The number of lines left blank as a top margin on the page.

RH - Running Header. Text repeated page after page — can include page number, date, title, even the time of day. You can make the header the same on all pages, or different on even (left-hand) and odd (right-hand) pages.

Body Text. The main text of the document.

FS - Footnote Separator. Indicates the characters and blank lines that you want to appear between the body text and the footnotes. In this case, a row of underlines is used.

FN - Footnote. Footnotes for the page are inserted here. XyWrite will put the entire footnote on the page, if possible.

Slack. If the text is less than a full page in length, the blank lines appear between the footnote and running footer. (This is the default setting, for BF=0.)

RF - Running Footer. Text repeated at the bottom of each page. Running footers have all the features of running headers listed above. (Odd or even is optional).

BT - Bottom Margin. Blank lines left as the bottom margin on the page.

PURPOSE

The **PL** (Page Length) command is the primary command used in the following procedure. This command determines the number of lines from the top edge of the paper down to the last line of text. Refer to PL shown in the previous illustrations.

For more detail on any command, refer to its description elsewhere in this chapter.

ACTION

Changing the Page Length Settings

Use the following procedure to change any of the page length parameters from their default settings:

1. **Position the Cursor.** Move the cursor to the top line of the page you want to affect. Insert all of the following commands (Steps 2-10) at this point.

2. **Form Depth.** Figure out the total number of lines on a complete sheet of paper. For example, at 6 lines per inch, an 11-inch sheet holds 66 lines (6 x 11).

   Type: F5 fd 66 ↵

   (Since FD 66 is the default setting, you are not *required* to type this in; however, doing so allows the form depth to be carried along with the document in case you later change the default.)

3. **Determine Page Layout.** Decide how many lines you want to reserve for each of the following. We have chosen the following values for our example.

Top Margin (TP)	6 lines
Running Header (RH)	3 lines
Running Footer (RF)	2 lines
Bottom Margin (BT)	5 lines

4. **Page Length.** The PL command has three parts to it: PLnom, PLmax, and PLmin. Determine these values, then enter them into your document in Step d.

   a. **PLmax.** Determine PLmax equal to FD–TP–RH. (Get TP and RH from Step 3). In our example, 66–5–2 = 59 :

      PLmax is 59.

b. **PLnom.** Decide on a value for the nominal page length. PLnom is typically PLmax minus 2 (for single-spaced documents).

   PLnom is 57.

   You have freedom in choosing a value for PLnom (depending on what you want for widow/orphan controls and non-breakable blocks). See the Page Length command later in this section for more details.

c. **PLmin.** Decide on a value for the minimum page length. A typical value for PLmin is PLnom minus 4:

   PLmin is 53.

d. **PL.** Type the PL command into your document, collecting the values from steps a, b and c:

   Type: F5 pl 57,59,53 ↵

5. **Top Margin.** Set the Top Margin to the value you decided in Step 3:

   Type: F5 tp 6 ↵

   Note: For TP to take effect on the *first* page, you must insert it on the first line of the document, before any text or spaces.

6. **Running Header.** Create the Running Header with the number of lines you decided in Step 3. For example, to say "Proposal" followed by two blank lines:

   Type: F5 rh ↵

   Type: Proposal ↵ ↵ ↵ F3

7. **Body Text.** You don't need to do anything else to set the number of lines of body text — you already set this number when you specified PL in Step 4. In our example, the body text has at most 50 lines (PLmax–TP–RH = 59–6–3 = 50).

8. **Footnote and Footnote Separator.** You don't need to do anything special to set the number of lines for footnotes — XyWrite treats them as body text. In our example, they would be included in the 50 lines mentioned in the previous step.

9. **Running Footer.** Set the Running Footer to the value you decided in Step 3 — a line for the page number followed by one blank line (to move the footer up off the bottom of the page):

   Type: F5 rf ↵

   Type: F5 pn ↵ ↵ F3

   Note: If there is a conflict, PL takes precedence over RF (and BT). That is, if you specify a PL that is too large, the footer will get pushed down off the page.

10. **Bottom Margin.** Set the Bottom Margin to the value you decided in Step 3:

    Type: F5 bt 5 ↵

    The only thing BT can ever do is move the running footer up from the bottom of the page. It cannot affect the length of body text — that is determined by PL.

NOTE #1 **Positioning the Paper in the Printer.** It is important when using running footers *you start with the top edge of the paper lined up with the printhead.* Then set the top margin using TP. If you were to manually advance the top edge of the paper *past* the printhead, you might find each running footer neatly (but mistakenly) printed at the *top* of the next page.

NOTE #2 **Vertical Measurement Unit.** The descriptions of the Page Length commands all use *lines* as the vertical measurement unit. In this context, 1 line is 1/6 inch, not one line of text. If you have chosen another vertical measurement unit, you must use that unit in all Page Length commands. (Refer to "Vertical Spacing" later in this chapter.)

FORMAT

CM PL *nom,max,min* Page Length

nom is the nominal number of lines allowed for PL, and includes the top margin, running head, body of text and footnotes.

max (optional) is the maximum value for PL.

min (optional) is the minimum value for PL.

PL is an embedded command.

PURPOSE

The **PL** (Page Length) command sets the number of lines from the top edge of the paper down to the last line of the footnotes (if any). As shown in an earlier figure, PL includes the top margin, running header, body text, and footnotes, but does not include running footer or bottom margin.

The maximum and minimum values allow the body of text to vary in length, to accommodate non-breakable blocks, widows and orphans. The default value originally set up in the STARTUP.INT file is:

```
PL 54,60,50
```

This setting allows for anywhere between 50 and 60 lines for PL, with 54 as the nominal length. (This PL setting is in the STARTUP.INT file on the original XyWrite disk you purchased — you are free to change this default setting.) The PL values have the following meanings:

- PLnom - This number of lines of text will appear on a page barring any forced breaks due to widows, orphans, non-breakable blocks or footnotes.
- PLmax - Body text will never, ever run past PLmax — this is an absolute maximum. Only running footers can appear below PLmax. A value for PLmax is necessary for widow control and can be important for placement of footnotes.
- PLmin - This allows text to be shorter when using orphan control or when placing footnotes.

ACTION

Setting the Page Length

To set the Page Length within a document:

1. Move the cursor anywhere on the page whose length you want to set.
2. To set a range of 56 to 60 lines for PL with a nominal length of 58 lines:

   Type: F5 pl 58,60,56 ↵

Result: This document can have anywhere from 56 to 60 lines including top margin, running header and footnotes. The variation can be due to non-breakable blocks (BB, NB), widows (WD), and orphans (OP). (A non-breakable block can cause the minimum to be less than 56.)

NOTE #1 **Embedded Command.** PL is an embedded command, visible as a triangle in the Normal Display. In the Expanded Display it would appear expanded — for example: «PL58,60,56».

NOTE #2 **Default Setting.** The default value for PL is originally located in your STARTUP.INT file as the statement DEFAULT PL=54,60,50. You can alter this setting as you wish. You can also move this statement to your Printer File as DF PL=54,60,50 if you wish.

NOTE #3 **Line Spacing.** If you print a document double-spaced (LS 2), the number of lines on the page is still between 50 and 60. PL specifies the number of lines output by the printer, regardless of whether they are printed on or are blank.

FORMAT CM FD *n* Form Depth

n is the total number of lines on a sheet of paper.
FD is an embedded command.

PURPOSE

Form Depth is our term for the length of the sheet of paper you print on. You set the Form Depth (FD) to equal the total number of lines on a sheet of paper. In other words, FD measures the number of lines from the top of one page to the top of the next.

ACTION

Setting the Form Depth

To set the form depth within a document:

1. Move the cursor to the top of the document whose sheet length you want to set.
2. For example, if you are printing 6 lines per inch on paper which is 14″ long, you would set the form depth at 84 lines as follows:

   Type: F5 fd 84 ↵

Result: The document is now set for a sheet of paper that is 84 lines in length.

TIP

A Matter of Convenience. You may find that the most convenient way to set the form depth is to set it once for all documents, rather than setting it separately within each document as shown above. Use the DEFAULT command to set a global FD, for all documents. See Note #2.

NOTE #1

Embedded Command. FD is an embedded command — it is embedded in text, visible as a triangle in Normal Display. In Expanded Display it is shown expanded — for example: «FD66».

NOTE #2

Default Setting. The default value for FD is 66 unless you use the DEFAULT command to change it. You can insert the DEFAULT command into the STARTUP.INT file or the Printer File (as DF).

FORMAT

CM **TP** *n* Top Margin
CM **BT** *n* Bottom Margin

n is the number of blank lines of margin.
TP and BT are embedded commands.

PURPOSE

The **TP** (Top Margin) command defines the number of blank lines that XyWrite automatically inserts between the top of a page and the first line of text (either running header or body text). (See Note #3.)

The **BT** (Bottom Margin) command moves the running footer up on the page. This is the only thing BT does. It has absolutely no effect on the length of body text or the placement of footnotes — those are determined strictly by PL.

ACTION

Setting the Top and Bottom Margins.
To set the Top and Bottom Margins:

1. Move the cursor to the *top line* of the document, ahead of any text or spaces. (Only embedded triangles can precede it on the top line.)
2. Type: F5 tp 3 ↵

Result: The top margin is set to 3 blank lines. You set BT in a similar manner.

NOTE #1

Ignoring the Top Margin Command. If you prefer to set the top margin manually (that is, by changing the position of the paper in the printer), you can use the DEFAULT command to change the TF setting. If TF=1, then any TP commands you issue are ignored. If TF=0 (the default), then TP commands are honored.

NOTE #2

Embedded Command. TP and BT are embedded in text, visible as triangles in Normal Display. In Expanded Display they are shown expanded: «TP3» and «BT3».

NOTE #3

Default Margin Settings. The defaults are TP 0 and BT 0 unless you use the DEFAULT command to change them. You can set up the DEFAULT command in the STARTUP.INT file or the Printer File.

ALSO SEE

Page Format. To see how the Top and Bottom Margins interact with the other page format commands, refer to the earlier section, "Page Length Procedures."

FORMAT CM PG *n* Page Break

n is the number of lines needed on a page before it breaks to a new page (optional).

PG is an embedded command.

PURPOSE

When you are printing a document, a **PG** (Page) command embedded in your document advances the paper to the top of the next page. This command is similar to giving a form feed to the printer. It can serve several purposes:

- **Starting a New Page.**
  Unconditional Page Break *(Option 1a)*
  When you insert PG in text with no value, it always causes the *next line* to be the first line of a new page.

  Conditional Page Break *(Option 1b)*
  If you use the value PG 20, the page will break *only* if the current page has 20 or more lines on it. In other words, PG 20 prevents the current page from printing with fewer than 20 lines.

- **Ejecting the Last Page.** *(Option 2)*
  You can use PG to automatically eject the last page of a document. When inserted at the very end of a document, PG ejects the last page from the printer. (More simply, you can eject using the DEFAULT EJ = 1 command — see Chapter 6.)

In all cases, the PG command has no effect on the line in which it is embedded. It causes the *next* line to be the start of a new page.
If you *don't* use the PG command to break a page, XyWrite creates a page break for you, after the number of text lines given by the PL (Page Length) command.

ACTION
(Option 1a)

Starting a New Page
— Unconditional Page Break

To create a hard page break in text:

1. Move the cursor to one line above where you want the new page to begin.
2. Enter the PG command:

   Type: F5 pg ↵

ACTION *(Option 1b)*

Starting a New Page
— Conditional Page Break

To conditionally break a page:

1. Move the cursor to *one line above* where you want the new page to begin.
2. Enter PG along with the fewest number of lines you want to appear on the page before the page breaks. For example:

   Type: F5 pg 40 ↵

Result: If the current page reaches 40 or more lines, a new page starts at the line following the PG command. If the current page does *not* reach 40 lines, the page does not break at the PG command.

ACTION *(Option 2)*

Ejecting the Last Page

To cause the printer to automatically eject the last page of a document:

1. Move the cursor to the very end of the document.
2. Type: F5 pg ↵

Result: When the document is printed, the PG command causes the last page to be ejected from the printer.

NOTE #1 **P-L Number.** Press Shift F9 to turn on the Page-Line (P-L) number at the top of the screen.

NOTE #2 **Extra Page with Running Header.** In Option 2 above, be sure you place the PG embedded triangle on the *very last line* of the document. If your document contains a running header and you follow the PG command with more than one carriage return, that running header will be printed on the next page.

NOTE #3 **Positioning the PG Command.** We recommend you place the PG command at the end of a line of text, rather than on a line of its own. In the latter case if the PG command fell immediately after a soft page break, XyWrite would produce a *blank* page. In the first case, XyWrite would instead produce a page with one line of text on it (which is easier to identify as being associated with a page break).

FORMAT		
	CM NB	Non-Breakable Block
	CM BB	Breakable Block
	NB and BB are embedded commands.	

PURPOSE

The commands **NB** (Non-Breakable Block) and **BB** (Breakable Block) allow you to keep a block of text on one page. You might define a paragraph, table, column of figures, or any other text as a non-breakable block. NB and BB always appear in pairs:

- NB begins the non-breakable block.
- BB ends the non-breakable block.

ACTION

Setting an Unbreakable Block of Text

To make a block of text unbreakable:

1. Move the cursor to the *start* of the text you want kept unbroken.

   Type: F5 nb ↵

2. Move the cursor to the *end* of the text you want kept unbroken.

   Type: F5 bb ↵

Result: All of the text between NB and BB is now an unbreakable block; that text cannot be split between two pages.

NOTE #1

Embedded Command. NB and BB are embedded commands — they are embedded in text, visible as triangles in Normal Display. In Expanded Display they are shown expanded as «NB» and «BB».

NOTE #2

Checking NB and BB Pairs. The commands NB and BB must be paired and they must alternate. You must not have more of one than the other. To check, use the search command:

CM SE /«LB»/

To type the four characters between the slashes, press:

Ctrl <

Alt Shift L

B

Ctrl >

ALSO SEE

Related Commands. Page Length command, PL, which has an effect on determining which page the non-breakable block is assigned to. With the command PL 54,60 an unbreakable block would stay on the page if it fits anywhere within the range of lines 54 to 60. Experiment to find which values for PL yield the best results for your application.

The conditional page break command (PG *n*) can serve a function similar to NB and BB. Refer to PG on the previous pages.

FORMAT

CM OP *m* Orphan

CM WD *n* Widow

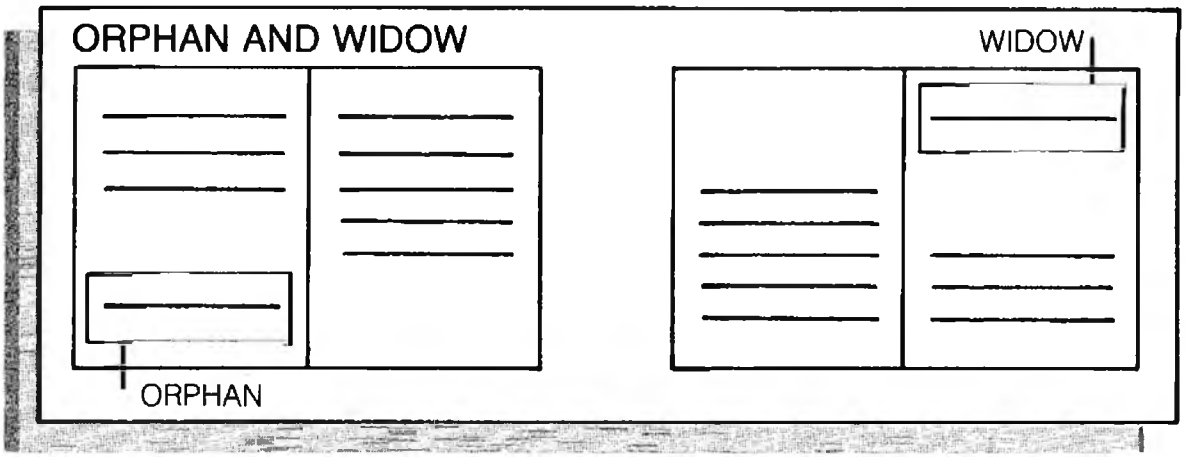
m is the minimum number of lines of a paragraph allowed at the bottom of a page.
n is the minimum number of lines of a paragraph allowed at the top of a page.
OP and WD are both embedded commands.

PURPOSE

A *widow* is the last few lines of a paragraph which are carried over to the top of the next page. Likewise, an *orphan* is the first few lines of a paragraph which appear at the bottom of a page. Widows and orphans of one or two lines are generally frowned upon because the lines appear estranged from their paragraphs.

The default values are WD 2 and OP 2. These settings allow only two or more lines of a paragraph to break away to a separate page. XyWrite is preset for this condition. WD and OP count only lines of text. (If you are double-spacing a document, it does not count the blank lines.)

Requirements. WD and OP require proper values for PLmax and PLmin, the second and third values in the PL (Page Length) command. The built-in default of PL 54,60,50 allows you to specify an orphan value of 1 to 5, and a widow value of 1 to 7.



ACTION **Setting Up Orphan Control.**
XyWrite has the default value of OP 2. To set OP to another value:

1. Move the cursor to the top line of the document.
2. Enter the OP command. For example, to allow the first 3 lines of a paragraph to fall at the bottom of a page, set OP to 3 as follows. (This prevents the paragraph from breaking at the first 1 or 2 lines.)

   Type: F5 op 3 ↵

   Result: This embeds the orphan command in the text, visible as a triangle. (Orphan control will not work unless you specify a value for PLmin less than or equal to PLnom. See Note #1.)

NOTE #1 **Calculating the Orphan Value.** To calculate the largest acceptable orphan value, find the PL settings you are using — subtract PLmin from PLnom and then add 1. For example:

54 - 50 + 1 = 5

In this example, 5 is the largest value you can specify in the OP command. (Note that you *must* specify a value for PLmin for orphan control to work.)

ACTION **Setting Up Widow Control.**
XyWrite has the default value of WD 2. To set WD to another value:

1. Move the cursor to the top line of the document.
2. To allow the last 3 lines of a paragraph to fall at the top of a page, set WD to 3 as follows. (This prevents the paragraph from breaking at the last 1 or 2 lines.)

   Type: F5 wd 3 ↵

   Result: This embeds the widow command in the text, visible as a triangle. (Widow control will not work unless you specify a value for PLmax greater than or equal to PLnom.)

NOTE #2 **Calculating Widow Values.** To figure the largest acceptable widow value, calculate PLmax minus PLnom and then add 1. For example

60 - 54 + 1 = 7

In this example, 7 is the largest value you can specify in the WD command.

NOTE #3 **Embedded Command.** OP and WD are embedded commands — they are embedded in text, visible as triangles in Normal Display. In Expanded Display they are shown expanded — for example: «OP3».

NOTE #4 **Default Orphan and Widow Settings.** The default values are OP 2 and WD 2 unless you use the DEFAULT commmand to change them. You can change these with the DEFAULT command in the STARTUP.INT file or the DF command in your Printer File.

NOTE #5 **Troubleshooting.** If you can't get the orphan or widow control to work:

- You may have mistakenly set a PL command with only one number, such as PL 58. For OP to work, you must specify a value for PLmin; for WD to work, you must specify a value for PLmax. See "Page Length Procedures" earlier in this chapter.
- If your document does *not* contain a Page Length command (PL), you should go ahead and set one. This will override any improper Page Length command that might be located in a DEFAULT PL statement in the STARTUP.INT file or in a DF PL statement in your Printer File.

NOTE #6 **How Orphan and Widow Control Works.** Let's illustrate how orphans and widows work using OP 3, WD 3, and the built-in default of PL 54,60,50. First of all, XyWrite makes sure that PL minimum (50) and maximum (60) values are never violated by orphan and widow control. That means that the printed text on a page cannot be made less than 50 lines by moving an orphan to the next page, nor can it be made more than 60 lines by keeping a widow on the same page.

When XyWrite prepares a document for printing, it scans the page endings. When it has finished scanning line 52, XyWrite looks ahead to see if a new paragraph would begin on line 53 or 54. If so, it starts that paragraph on a new page, preventing an orphan of one or two lines. If the paragraph is still continuing at line 54, XyWrite looks ahead to see if that paragraph would end on line 55 or 56. If so, it allows that paragraph to finish on that page, preventing a widow of one or two lines, and starts the next paragraph on the new page.

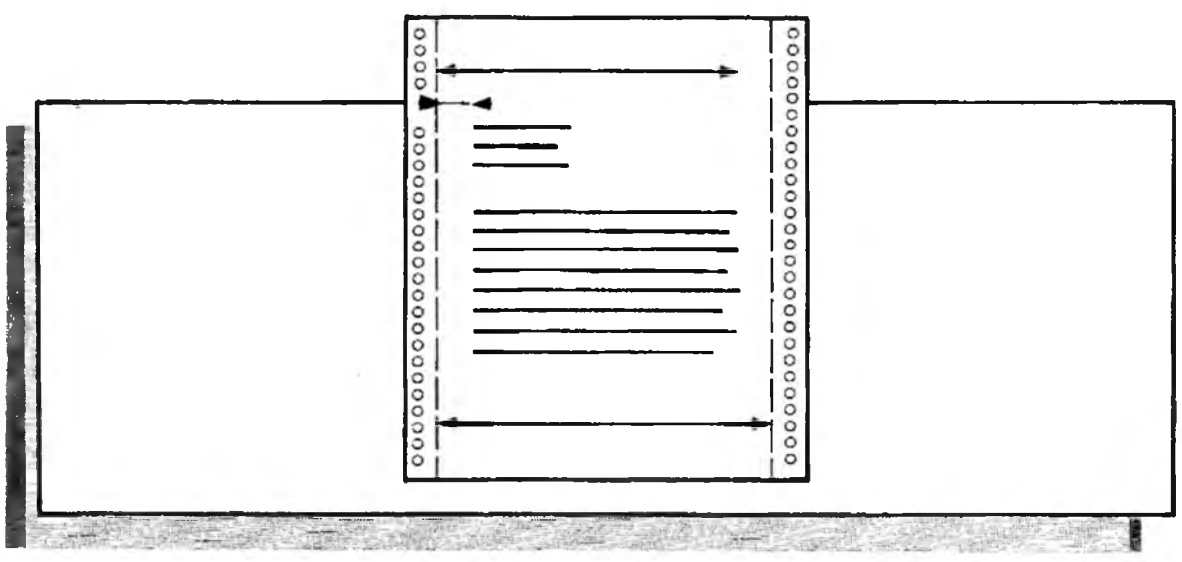
NOTES

INTRO

In this section we cover the commands which affect the *width* of your document. Except for Offset, all of these commands make themselves apparent as symbols on the Ruler. You can make changes either directly on the ruler or by specific command.

We start with a description of the ruler and how you can make format changes directly on the ruler. Individual discussion of each page width command follows.

CONTENTS



The Ruler is the third line from the top of the screen. It has 80 marks, one for each column. The marks in the ruler are as follows. (You can change these markers in the Printer File if you wish, with the RL setting.)

Marker		Example of Command
‖	Left Margin	LM 5
‖	Right Margin	RM 70
▶	Normal Tab	TS 10
◀	Flush Right Tab	TS 10R
▼	Flush Center Tab	TS 10C
⌂	Decimal Tab	TS 10D
⌊	1st Indent	IP **5**,10 *(first number)*
⌉	2nd Indent	IP 5,**10** *(second number)*

The cursor marker is the mark on the ruler that moves as the cursor moves through text. (The ruler symbols show through it.)

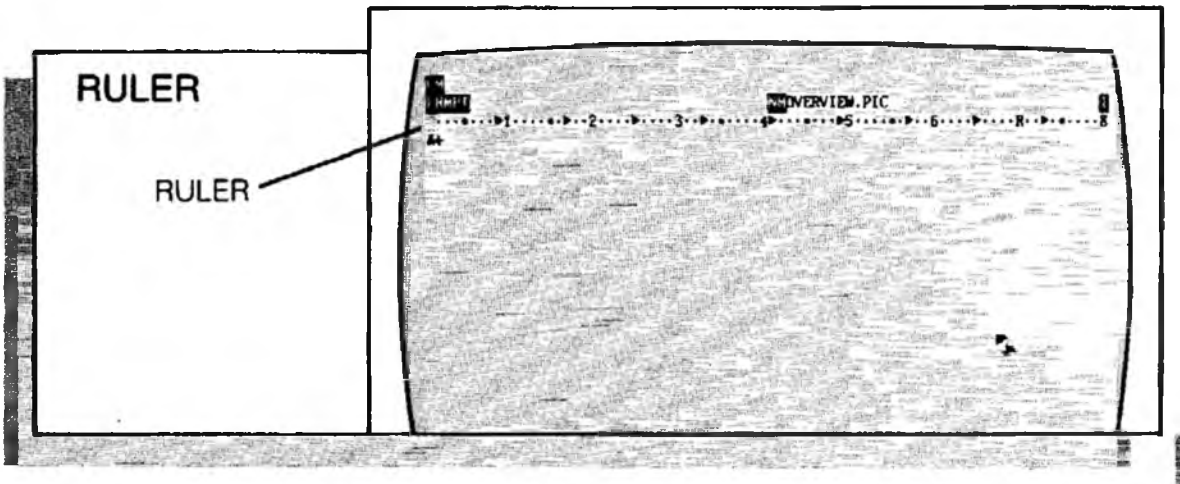
The display is 80 characters in width. Your document, however, can be as wide as 256 characters. The display automatically scrolls horizontally, if need be, as you move the cursor.

The numbers 1 through 8 in the ruler represent columns 10 through 80. If you print at 10 characters/inch, this ruler corresponds to inches.

Each TS, TR, LM, RM and IP command alters the markers displayed on the ruler line. You change these commands throughout your document for different purposes — perhaps one for paragraphs, another for tables, still another for headings.

You can write as many different formats in a document as you want; each one takes effect where the embedded format triangle is located, and stays in effect until another triangle is encountered, or to the end of the document.

The ruler indicates the tabs, indents and margins which are in effect *at the location of the cursor*. As you move the cursor down the screen, each time you pass a format change (visible as an embedded triangle), the ruler changes to indicate the new settings.



FORMAT Alt Tab Ruler Menu
This is an immediate command.

PURPOSE

Alt Tab lets you change the ruler settings in a simple and direct manner. By moving the cursor along the ruler, you can modify the left margin, right margin, tabs and indents without having to enter their respective commands LM, RM, TS and IP.

ACTION

Changing the Ruler Settings.
To change the tab settings, left margin, right margin or indents:

1. Press: Alt Tab

   Result: Notice the choices that are listed.

2. Let's select Alt Tab to "remove prompt menu."

   Press: Alt Tab (again)

   Result: This removes the prompt menu so you can see your document while you set the tabs.

3. Move the cursor marker left or right along the ruler to the position you want to change.

4. Select the action you want:

   L Left margin
   R Right margin
   I Indent of first line in paragraph
   H Hanging indent (remaining lines in paragraph)

   T Tab stop for a normal tab (flush left)
   G Flush right tab
   C Centered tab
   D Decimal tab

   X Clears an individual marker
   Z Clears all tab and indent markers
   A Abandons the menu without making any changes to the ruler, and returns to your document

5. Repeat steps 3 and 4. To finalize your selections:

   Press: ↵

Result: The necessary LM, RM, TS and IP commands are inserted in the document at the cursor location in the text, visible as embedded triangles (▲).

EXAMPLE **Changing the Tab Settings.** Let's change the tab stops to 5 and 10 with a decimal tab at 25:

1. Press: Alt Tab
2. Press: Alt Tab (again)
3. Press: Z (clears the tab markers)
4. Move the cursor to position 5.
5. Press: T (sets a tab stop)
6. Move the cursor to position 10.
7. Press: T (sets another tab stop)
8. Move the cursor to position 25.
9. Press: D (sets a decimal tab stop)
10. Press: ↵

Result: The embedded command «TS5,10,25D» is entered into the document at the cursor location.

NOTE **Setting Defaults with the Ruler.** When you enter selections with the Alt Tab menu and there is no document open on the screen, the selections you make become defaults — they become the initial formats for *all* files — both new and existing — until you QUIT XyWrite. (These settings are overridden by TS commands embedded in your document.) This feature is useful for making temporary ruler settings for new files.

SEE ALSO **Related Commands.** The style commands (SS, US, NS, PS) are another way to easily change the ruler settings. You can create your own standardized rulers and switch between them at will.

FORMAT		
	CM LM *n*	Left Margin
	CM RM *n*	Right Margin
	n is the column number (0 to 255) for the margin	
	LM and RM are embedded commands.	

PURPOSE

The **LM** (Left Margin) and **RM** (Right Margin) commands adjust the margins in a document. When you move the left margin, the tabs do not move along with it (unless you use Relative Tabs by specifying RT ON, explained in "Tab Settings" later).

ACTION

Setting the Left or Right Margin.
To set a margin:

1. Move the cursor to start of the line whose margin you want to change, or to a previous line. (See Note #2.)
2. Enter LM or RM. For example, to set the left margin to position 10:

   Type: F5 lm 10 ↵

Result: The LM command is embedded in the text as a triangle. The left margin remains in effect until another LM command is encountered.

NOTE #1

Embedded Command. LM and RM are embedded commands — they are embedded in text, visible as triangles in Normal Display. In Expanded Display they are shown expanded — for example: «LM10» and «RM80».

NOTE #2

Immediate Effect. To make the LM or RM command take effect on the same line on which it rests, make sure there is no text or spaces ahead of it on that line. (You may place other embedded triangles ahead of it.)

NOTE #3

Default Margins. You can also set up LM and RM using the DEFAULT command (in the STARTUP.INT file or the Printer File, or on the Command Line). The defaults are LM 0, RM 78 (unless you use the DEFAULT command to change them).

ALSO SEE

Related Commands. IP (Indent Paragraph) is similar to LM. IP allows you to indent the left margin. RT (Relative Tab) lets you set tabs relative to the left margin.

FORMAT — CM**OF** *r,l* — Offset

r and *l* are amount of offset for right and left pages, respectively (0 to 255).
If *l* is omitted, all pages take on the *r* offset.
OF is an embedded command.

PURPOSE

The **OF** (Offset) command shifts the horizontal position of the text on the page for printing. OF shifts the entire document — the left and right margins, all text, tabs and indents. While the offset appears on the printout, it is not visible on the display. (This enables you to view 80 columns of text on-screen.)

The *r* and *l* in OF *r,l* are actually the number of *margin units* you want the page shifted. If only one (*r*) is used, all pages are shifted by that number; if two numbers are inserted, *r* is the number the odd (right-handed) pages are offset, *l* the number for even (left-handed) pages.

Allowing different offsets for left- and right-handed pages is invaluable to allow space along the binding edge. Space for staples, punched holes or other binding is thereby created to the right of text on left-handed pages, and to the left of text on right-handed pages.

In 10-pitch type, there are 10 margin units to the inch. To shift the page one inch to the right, for example, use:

```
.of 10
```

This means that you can display a full-screen document with no left margin, and then print it with a 1-inch left margin. This is especially useful when printing with 12-pitch type — it allows you to use the entire 80-character display and still have 1" for left and right margins. (80 characters divided by 12 characters/inch equals just over 6-1/2.")

Without using the OF command, if you were to insert a 1" left margin on the display, you would have to use horizontal scrolling to display 80 characters of text.

ACTION **Setting the Same Offset for All Pages.**
For example, to create a 1" left margin on a printer set to a pitch of 12 characters/inch, you would set the offset to 12:

1. Move the cursor to the top of the document. (The offset takes effect starting with the line where the OF command is embedded.)

2. Type: F5 of 12 ↵

Result: The document is printed with a blank left margin of 12 spaces (1"). The entire document is shifted this amount.

ACTION **Setting Left and Right Offsets.**
For the same 12-character pitch setting, to create a 1 1/2" left margin for odd pages and a 1" left margin on even pages, you would set the offset to 18 and 12, respectively:

1. Move the cursor to the top of the document.

2. Type: F5 of 18,12 ↵

Result: When the document is printed, even pages will have a 12-space left margin; odd pages, 18 spaces.

NOTE #1 **Embedded Command.** OF is an embedded command — it is embedded in text, visible as a triangle in Normal Display. In Expanded Display it is shown expanded — for example: «OF18,12».

NOTE #2 **Previewing the Offset.** You may view the offset on the display, if you wish, by using the TYPES (Type to Screen) command.

NOTE #3 **Default Offset Settings.** You can also set OF in the STARTUP.INT file or the Printer File using the DEFAULT command. The default is OF 0,0 (unless you use the DEFAULT command to change it).

FORMAT	CM IP *m,n*	Indent Paragraph
	m and *n* are described below. IP is an embedded command.	

PURPOSE

The **IP** (Indent Paragraph) command gives you an easy way to indent paragraphs. You can use IP rather than LM to indent whole paragraphs. Tab stops do not change position when you insert an IP command.

The indent is measured from the current Left Margin (LM). The *m* is the number of spaces the first line of the paragraph is indented; *n* is the number of spaces subsequent word-wrapped lines are indented.

The IP command offers you several ways to style a paragraph, as shown in the illustration on the next page.

ACTION

Setting a Paragraph Indent.
To set a paragraph indent:

1. Move the cursor to start of the line you want indented, or to a previous line. (See Note #2 below.)

2. For example, to indent the first line 10 spaces and the rest of the entire paragraph 5 spaces:

   Type: F5 ip 10,5 ↵

   Result: The IP command is embedded in the text as a triangle. All paragraphs which follow are indented, until another IP command is encountered.

3. To discontinue paragraph indenting:

   Type: F5 ip ↵

   Result: All subsequent lines are not indented. The command «IP0,0» is inserted into the text.

COMPARISON OF PARAGRAPH INDENTS

▲

Example of IP 5,0. This is a paragraph with a normal indent; the first line is indented five spaces while the remaining lines are not indented. This IP command is embedded in the above triangle.

▲

Example of IP 10,10. You can also indent entire paragraphs from the left margin like this using the IP command.

▲

Example of IP 5,10. Similarly, you can do a hanging indent, where each paragraph hangs down from the first line. Some people call this a negative indent.

▲▲

Be Inventive!

Example of IP 0,20 with TS 20. You can use hanging indents in novel ways such as this, where the title sits out in the left margin. After typing the title, you tab over to the start of the paragraph. Each line in this paragraph word-wraps back to column 20. When writing the manuscript for this XyWrite Reference Guide, we used these two commands to achieve the hanging indents that you see.

▲

Example of IP 0,0. This paragraph is an example of text without indents. Notice the paragraph is up against the left margin of text.

COMPARISON OF PARAGRAPH INDENTS

▲

Example of IP 5,0. This is a paragraph with a normal indent; the first line is indented five spaces while the remaining lines are not indented. This IP command is embedded in the above triangle.

▲

Example of IP 10,10. You can also indent entire paragraphs from the left margin like this using the IP command.

▲

Example of IP 5,10. Similarly, you can do a hanging indent, where each paragraph hangs down from the first line. Some people call this a negative indent.

▲▲

Be Inventive!

Example of IP 0,20 with TS 20. You can use hanging indents in novel ways such as this, where the title sits out in the left margin. After typing the title, you tab over to the start of the paragraph. Each line in this paragraph word-wraps back to column 20. When writing the manuscript for this XyWrite Reference Guide, we used these two commands to achieve the hanging indents that you see.

▲

Example of IP 0,0. This paragraph is an example of text without indents. Notice the paragraph is up against the left margin of text.

NOTE #1 **Technical Description.** The way IP *m,n* works is:

- *m* determines the amount of indent for a line which is preceded by a *hard return*.
- *n* determines the amount of indent for a line which is *not* preceded by a hard return (a line which is word-wrapped).

NOTE #2 **Immediate Effect.** To make the IP command take effect on the same line on which it rests, make sure there is no text or spaces ahead of it on that line. (Only other embedded triangles can be placed ahead of it.)

NOTE #3 **Omitting a Value.** If you do not specify one or the other value in the IP command, the missing value is interpreted as a zero. Thus, IP 5 is equivalent to IP 5,0; and IP,10 is equivalent to IP 0,10.

NOTE #4 **Embedded Command.** IP is an embedded command — it is embedded in text, visible as a triangle in Normal Display. In the Expanded Display, IP would appear embedded in the text — for example: «IP5, 10»

NOTE #5 **Default Indented Paragraph Settings.** You can also set up IP using the DEFAULT command (in the STARTUP.INT file or the Printer File). The default is IP 0,0 (unless you use the DEFAULT command to change it).

FORMAT		
	CM **TS** *n1,n2,n3, . . .*	Tab Set
	CM **TR**	Tab Reset
	CM **RT** *x*	Relative Tabs

n1,n2,n3 are column numbers for tab stops (0-255).
x is ON or OFF
Maximum of 21 tab stops are allowed in the TS command.
TS, TR and RT are embedded commands.

PURPOSE

The **TS** (Tab Set) command sets tab stops in your text. A **tab stop** is a pre-set column position to which the cursor moves when you press the tab key.

Four kinds of tabs are available: Left Flush, Right Flush, Center Flush and Decimal. They are described extensively later in this section under "Flush Tabs."

The **TR** (Tab Reset) command resets tab stops to the default settings — that is, 8, 16, 24, and so on (unless you have used the DEFAULT command to change them).

The **RT** (Relative Tabs) command lets you specify whether the tabs are relative to the left margin (RT ON) or not (RT OFF). The default is RT OFF. To set relative tabs for all files, use DEFAULT RT=1 in STARTUP.INT (see Default in Chapter 6).

We'll cover the following three procedures:

- Creating Tab Stops (TS)
- Changing the Tab Stops
- Resetting the Tab Stops (TR)

ACTION

Creating Tab Stops (TS).

To create a new set of tabs:

1. Move the cursor to the point in the text where you want the tab settings to begin.
2. Enter the TS command. For example, to set normal tabs at 3, 6 and 9 spaces in from the left margin:

   Type: F5 ts 3,6,9 ↵

   Important — don't insert spaces after the commas.

Result: The TS command is embedded in the text, displayed as a triangle. The tabs take effect from that point forward, until another TS triangle is encountered or until the end of the document.

3. (Optional) If you want to be able to change the left margin and have the tabs move right along with it, specify relative tabs:

   Type: [F5]rt on[↵]

Result: If you view this command in expanded mode, you'll notice it appears as «RT1» (not «RTON»). Similarly, RT OFF appears as «RT0».

NOTE #1 **The Ruler.** The ruler indicates the tabs (and margins) which are in effect *at the location of the cursor*. As you move the cursor down the screen, each time you pass a Tab Setting, the ruler changes to indicate the new tab settings.

NOTE #2 **Using Several TS Commands.** You can put as many varied TS commands as you like throughout your document. Typically you would use them for different formats — perhaps one for paragraphs, another for tables where you have data in columns.

ACTION **Changing the Tab Stops.**
To add, delete or move tab stops in an existing TS triangle:

1. Move the cursor to the TS embedded triangle which contains the tabs you want to change.
2. Press: [Ctrl][F9]

   Result: This switches to the Expanded Display, revealing the tab settings — for example: «TS3,6,9»
3. Add, delete or change the tab numbers to the new tab settings you want.
4. Press: [Ctrl][F9]

Result: This returns you to Normal Display. The new tab settings are now in effect following the command.

ACTION **Resetting the Tab Stops (TR).**
To reset the tab stops to their default settings:

1. Move the cursor to the point in the text where you want the default tab settings to begin.

2. Type: F5 tr ↵

Result: The TR command is embedded in the text. The default tab settings take effect at this point. The original default settings are 8, 16, 24 and so on. You can use the DEFAULT command to change these settings.

NOTE #3 **Embedded Commands.** TS, TR, and RT are embedded commands. They are embedded in text, visible as triangles in Normal Display. In Expanded Display they are shown expanded — for example: «TS3,6,9»

NOTE #4 **Default Tab Settings.** You can also set up TS using the DEFAULT command (in the STARTUP.INT file or the Printer File). The default is TS 8,16,24, . . . and so on, to 168 (unless you use the DEFAULT command to change it).

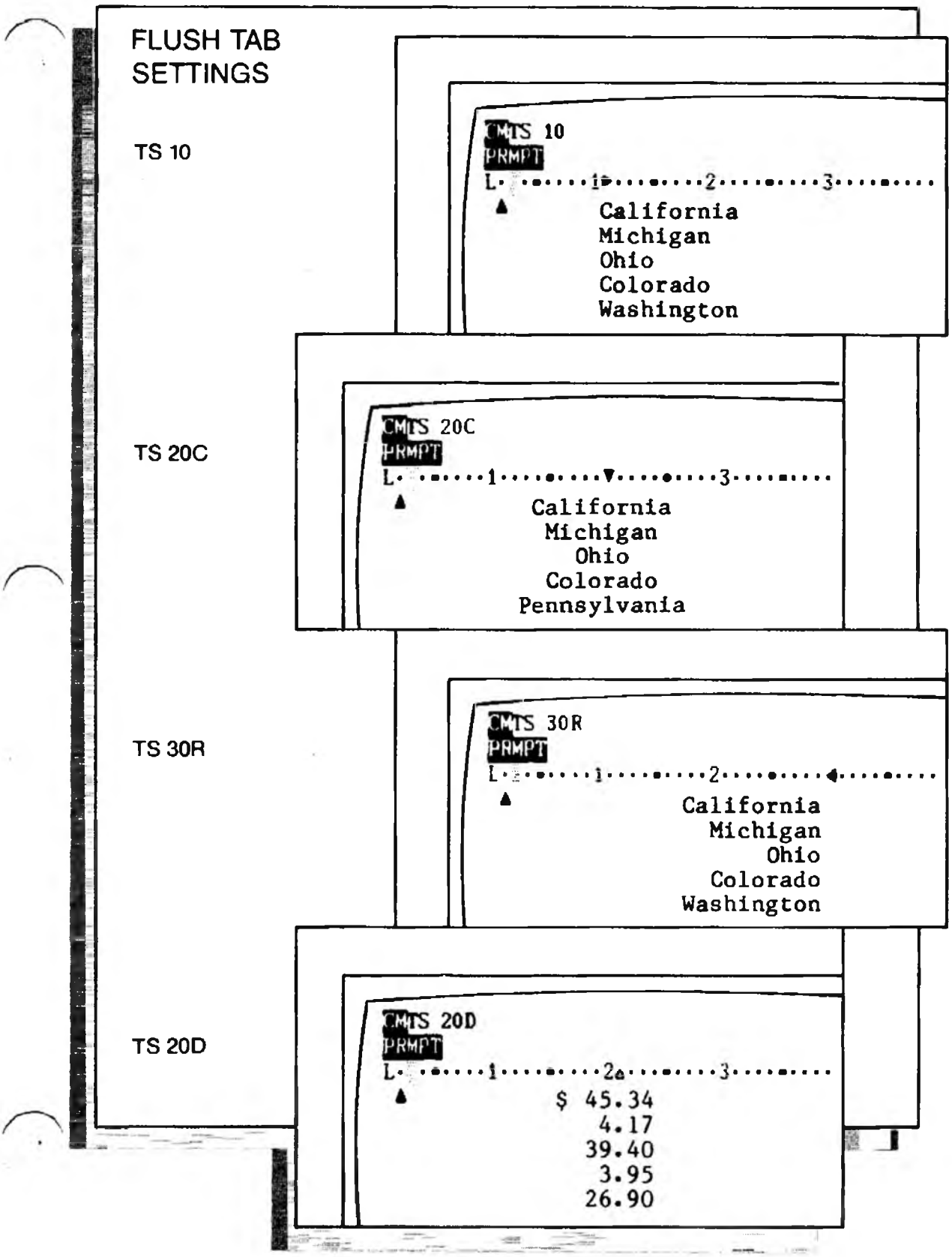
NOTE #5 **Flush Tabs.** XyWrite provides you with the following four kinds of tab stops (illustrated on the following page). The first three all follow the same rule:

- ▶ Normal Tab — Aligns text flush left against the specified tab stop. (Also known as a flush left tab.) Example: TS 10

- ◀ Flush Right Tab — Aligns text flush right against the specified tab stop. Example: TS 30R

- ▼ Flush Center Tab — Centers text on the specified tab stop. Example: TS 20C

- ⌂ Decimal Tab — Lines up a column of numbers on their decimal points. Example: TS 20D


FLUSH TAB SETTINGS
TS 10
CMTS 10
PRMPT
L· ·····1·····2·····3·····
California
Michigan
Ohio
Colorado
Washington
TS 20C
CMTS 20C
PRMPT
L· ·····1·····▼·····3·····
California
Michigan
Ohio
Colorado
Pennsylvania
TS 30R
CMTS 30R
PRMPT
L· ·····1·····2·····◄·····
California
Michigan
Ohio
Colorado
Washington
TS 20D
CMTS 20D
PRMPT
L· ·····1·····2a·····3·····
$ 45.34
4.17
39.40
3.95
26.90

NOTE #6 **The Tab Key.** The tab key on the keyboard gives you control over movement of the cursor and text:

Tab	If you press the tab key in the middle of text, the text (and cursor) moves one tab to the right. (A tab character *is* entered invisibly into the text.) In Overstrike mode, a character is deleted.
Ctrl Tab	Moves the cursor to the next tab *without* moving the text along with it. (A tab character is *not* entered into the text.)
Shift Tab	Moves the cursor to the *previous* tab without moving text.

To move text *back* one tab space, use the Backspace key to delete the tab character.

Once you have tabbed the cursor over to the last tab, pressing the tab key moves the cursor only one space at a time.

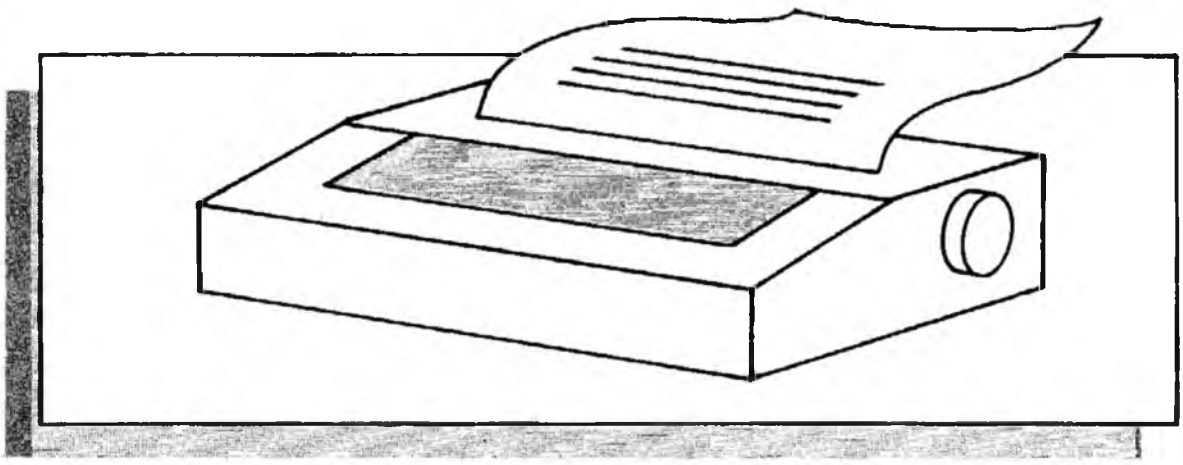
NOTE #7 **The Tab Character.** When you press the tab key, the character ASCII 9 is entered invisibly into the text. Even though you can't see the character, you can treat it like any ordinary character — for instance, you can delete it and search for it. To execute the search, press F5, type se /, press the Tab key, type another /, and press ↵.

NOTE #8 **Related Commands.** XyWrite also provides the flush commands FL (Flush Left), FR (Flush Right) and FC (Flush Center); they position *all* the text between the margins on all lines following the command. They should not be confused with the flush tabs.

INTRO

When you print documents, you often want special control over your printer. With XyWrite you can cause your printer to stop on a given line or at the end of specified pages and have it prompt you when it's completed its job. You can also request special features from your printer (using the PC command), such as feeding sheets from either of two paper bins.

CONTENTS



FORMAT		
	CM **PA** *message*	Pause
	CM **PR** *message*	Prompt

These are embedded commands.

PURPOSE

PA (Pause) stops the printing of a file at the point the PA command is embedded in the text, and displays its message on the prompt line. You press [+] to continue printing.

PR (Prompt) displays a message on the prompt line during the printing of a file *without* stopping the printer. When XyWrite encounters PR during printing, it displays the message and continues to output to the printer. The message is removed when you strike the next key (or when the next PR or PA message is encountered).

You can insert PR in your document to monitor the progress of a remote printer. Include a message such as "Printing is complete" at the very end of your document.

You may want to use these commands to provide a reminder of what should be done when the printer stops — to change a printwheel, insert a different kind of paper (such as letterhead), or change a ribbon.

ACTION

Causing the Printer to Pause.

To cause your printer to stop and display a message:

1. Move the cursor to the point in text where you want the printer to stop.
2. Type: [F5]pa Install Italic Printwheel[↵]

Result: When you print this document, it automatically stops at the point where the PA command is embedded in the text — the message "Install Italic Printwheel" appears on the prompt line. Press the plus key [+] to restart the printer.

You enter PR in the same way you entered PA in Step 2 above. The only difference is that the printer will not stop printing when it displays its message.

ALSO SEE

Related Commands. The P option in TYPE *filename*,P stops the printer after each page.

FORMAT

CM **AP**	AutoPause
CM **NP**	No Pause

These are embedded commands.

PURPOSE

AP (AutoPause) causes a document to pause at the end of each page. You resume printing with [+]. You can embed the AP command wherever you want the pause to begin.

NP (No Pause) defeats AutoPause. You use it after an AP command to allow the document to once again print continuously without pausing.

ACTION

Inserting Page Pausing.
To cause a document to pause at the end of certain pages:

1. Move the cursor to the page you want pausing to begin.
2. Type: [F5] ap [↵]
3. Move the cursor to the page you want continuous printing to resume.
4. Type: [F5] np [↵]

Result: When you print your document, it prints without stopping until it reaches the page containing the AP command. When the printer stops, press [+] to resume printing. Printing stops after every page until it reaches the NP command — it then continues printing without stopping.

NOTE #1

Embedded Commands. AP and NP are embedded commands — they are embedded in text, visible as triangles in Normal Display. In Expanded Display they are shown expanded — for example: «AP».

NOTE #2

Related Commands. The AP command causes the printer to pause exactly the same as the P option with the TYPE command. The difference is that AP is *embedded* in the document, while the P option is not. With AP you can cause some pages to pause and others not to pause.

FORMAT **CM PC #** Printer Control

is the row number in the Printer Control table corresponding to the control string you want to send
PC is an embedded command.

PURPOSE

PC (Printer Control) allows you to send control strings to the printer. These strings are defined in a Printer Control table you specify in the Printer File. (See Note #1.) The PC command allows you to control features of your printer from within the text file being printed. For example, a multiple-bin sheet feeder on a printer needs control codes to switch bins. With PC commands in your mail-merge letter file, the printer automatically switches to letterhead or envelope bins as needed.

ACTION

Inserting a Printer Control String.
This procedure assumes that you have already created a Printer Control table in your Printer File and that this table contains, in row 3, the code to activate Bin B.

Let's assume your printer is normally set up to print on letterhead paper from Bin A (for page 1). You want to switch to Bin B, which has plain paper, for pages 2 and up.

1. Move the cursor to the point in your document where you want to activate Bin B — that is, to the start of page 2. (Place the cursor ahead of any text or spaces on that page, although other embedded commands can be ahead of the cursor.)
2. Enter the PC command along with the row number of the string that you want to send.

   Type: F5 pc 3 ↵

Result: When you print your document, the code to switch to Bin B is sent to the printer at the top of page 2. You could include a PR (Prompt) command to flash a message on the prompt line, to remind you what the PC command is for.

NOTE #1 **Printer Control Table.** The PC command requires that a Printer Control table be set up in your Printer File. This table contains the codes that control the printer, such as switching paper bins, changing ribbon color, or ejecting a page. Each control string is a separate line. See the Printer File section of Chapter 6 for more details.

FORMAT **CM PI** *string*

string is a printer control code.

This is an embedded command.

PURPOSE

PI allows you to send control strings directly to the printer. Unlike the PC (Printer Control) command, which references strings that are defined in the PC table of the printer file, the PI command contains the actual control codes. XyWrite does not respond to these control codes; it simply passes the codes to the printer when you use the TYPE command.

ACTION

Inserting a Printer Control String.
Let's assume you are using a Corona laser printer. Like many laser printers, the Corona has its own commands for drawing rules and boxes. To insert a rule, say between two tables:

1. Move the cursor to the point in your document where you want to insert the rule.

2. Enter the PI command along with the printer-specific control code for drawing a rule. For a Corona laser printer:

   Type: F5 `pi @rule 100,100 2000,110;` ↵

   Result: When you TYPE your document, the code to draw a rule is sent to the printer.

FORMAT **CM in** *filename,depth*

filename is the name of the file you want included.
depth is the vertical space required by *filename*

This is an embedded command.

PURPOSE

The **IN** (Include) command allows you to have the contents of another file (one created on XyWrite or on another software system) merged into your document when you send it to the printer. The separate file can be a graphics file, a spreadsheet, or even another text file, but it must be *output ready*. That is, it must be formatted for output to the specific printer that you're using. For example, it you want to "include" another XyWrite file, first create a formatted version of the file (using TYPEF), and specify the name of the formatted version with the IN command.

For XyWrite to maintain the proper page-line count, you must specify the depth or amount of vertical space required by the foreign file, including white space above and below it. Specify this depth using the vertical measurement unit in effect for your document. Typically, this unit is a line, or 1/6 inch. To insert a 3-inch graphic into your document, you must specify a depth of 18.

ACTION

Including Another File.
To merge another file into a XyWrite file for printing:

1. Display the XyWrite file on the screen.
2. Move the cursor to the point where you want the graphics file to be inserted.
3. Type: F5 `in orgchart,12` ↵

   Result: An embedded triangle appears in your file, indicating the point where ORGCHART will be inserted when you print your document. The P-L indicator reflects the 12 lines reserved for ORGCHART.

NOTE

Printer Settings. If the foreign file contains codes that change a printer setting, you have to use the PI (Printer Insert) command in your document to reset the printer to its original state after the foreign file has been output.

INTRO

A style is a set of embedded format commands (such as left margin, right margin, tabs and offset) defined together under one name. This allows you to standardize your own formats, and facilitates switching between them.

You might define one style for letters (and name it LETTER), another for memos (MEMO), still another for reports (REPORT), and so forth. Once you define styles by name (using SS), you can recall them (using US) by name. You can also define a set of ordered styles and use them by simply saying next style (NS) or previous style (PS).

CONTENTS

FORMAT **CM SS** *name,nm=n,nm=n,nm=n, . . .* Save Style

name is a name you specify for this style you are defining.
nm is the two-letter name of the command (see below).
= (equal sign) separates the name from the value.
n is the value of the command.
, (comma) separates the commands.
SS is an embedded command.

PURPOSE

SS (Save Style) saves the current default settings under a name you specify. Once you define a style with SS, you use the **US** (Use Style) command at any point in text where you want the style to take effect.

The SS command in fact does *more* than save the settings you specify in the command. It also saves all of the other current settings implicit at that point in the document, including all of those listed below.

When you later invoke US LETTER, you will be recalling *all* of the format settings saved by the SS command.

Normally, you embed the SS command at the beginning of a document and refer to it with the US command throughout the document. Thus, you can standardize on styles you use regularly — one style for letters, another for reports, and still another for outlines.

You can specify for *nm* any DEFAULT setting. For example:

AL	Automatic leading	MD	Any character mode
AP	Autopause	NJ	No justification
FD	Form depth	NP	No pause
FC	Flush center	OF	Offset
FL	Flush left	PL	Page length
FR	Flush right	PT	Print type
HY	Hyphenation	RM	Right margin
IP	Indent paragraph	SP	Set page number
JU	Justification	TP	Top margin
LL	Line leading	TS	Tab settings
LM	Left margin		
LS	Line spacing		

ACTION **Saving a Style.**
To set up a style definition:

1. Go to the top of your document:

   Press: Ctrl Home

2. Type: F5 `ss letter,lm=8,rm=65,md=nm,ip=5,0` ↵

Result: This defines a style named LETTER with a left margin of 8, a right margin of 65, normal character mode, and an indent of 5 at the first line of each paragraph. To make use of this style, it is necessary for you to invoke it with the Use Style command.

NOTE #1 **Style Within Same Document.** The Save Style command must be in the document that uses it. While it is an embedded format command, it cannot be set up as a default.

NOTE #2 **Placing the SS Command.** The SS command does not need to be at the very beginning of the file, so long as it is placed before the first Use Style (US) command that invokes it.

ALSO SEE **Related Commands.** The set of format commands given on the previous page (LM, RM, TS, etc.) is the same set used by the DEFAULT command. For more information, see the DEFAULT command in Chapter 6, Customizing.

FORMAT

CM US *name*	Use Style
CM NS	Next Style
CM PS	Previous Style

name is the name of the style you want to use — it must have been defined previously with the SS command. These are embedded commands.

PURPOSE

The **US** (Use Style) command invokes the style called for by the *name*. The *name* must be previously defined (somewhere in the same document) with the Save Style (SS) command.

Once you have defined several styles, you can change styles without calling them by name. Knowing their sequence in the document, you can invoke either the Next Style (NS) or the Previous Style (PS). This is especially useful for outlines with many levels.

ACTION

Using a Style by Name

To call a style by name:

1. Move the cursor to the point in text you want to start a new style.
2. Enter the US command along with the name of the style you want. For example:

   Type: F5 `us letter` ↵

Result: The style named LETTER takes effect at that point in the document. For US to work, LETTER must be defined earlier in the document with the SS command.

ACTION **Using Next Style and Previous Style.**
Before starting, assume we have set up a series of styles called HEAD, SUBHEAD, and TEXT (in that order) with three SS commands.

Next assume that we have given a US HEAD command to format our first heading. To use the next style, which is SUBHEAD:

Type: F5 ns ↵

Result: At the place we embed this command, the SUBHEAD style becomes the new format of the document. To use the TEXT style:

Type: F5 ns ↵

Result: Now the style is changed to the TEXT style. After typing text, when you come to a point where you want to use the SUBHEAD style again:

Type: F5 ps ↵

NOTE **Inserting Other Format Commands.** When you give new format commands that you want to affect large parts of the document (such as LM, RM or IP), be aware that they are superceded by the next US, NS or PS command that follows. We recommend you include these in the SS command — in fact, the more format commands you can include, the more predictable your results will be.

NOTES

INTRO

To ensure readability of your printed page when you are mixing fonts of different sizes, you must take into account the vertical spacing you are using. Vertical spacing of type is measured from the baseline of one line to the baseline of the next. This measurement is commonly referred to as *lead* or *leading*. Printers allow leading to be adjusted in very small increments, as illustrated below.

CONTENTS

The amount of space between lines is known as leading. There is no set rule to follow. Too much leading can sometimes be as bad as not enough. Type faces with

The amount of space between lines is known as leading. There is no set rule to follow. Too much leading can sometimes be as bad as not enough. Type faces with

The amount of space between lines is known as leading. There is no set rule to follow. Too much leading can sometimes be as bad as not enough. Type faces with

Vertical Spacing

PURPOSE

XyWrite gives you a variety of options for defining and changing the leading values. Some of these options involve settings in the Printer File while others involve embedded commands. Because of the close link between the Printer File settings and the embedded commands, we discuss both here.

There are three settings in the Printer File that affect vertical spacing:

- Minimum vertical spacing unit (VS)
- Vertical measurement unit (VU)
- Automatic vertical leading (VL)

The first two settings work with the embedded vertical spacing commands to tell the printer exactly how much vertical movement is required. The third setting, automatic vertical leading, provides the correct leading value for whatever font you are using — a valuable tool if you are using different type sizes within a document.

A brief description of each of these settings follows. For more in-depth descriptions of Printer File settings, refer to Chapter 6.

Minimum Vertical Spacing Unit (VS). Each printer has a minimum vertical spacing unit that it can move. For example, the minimum vertical movement of many daisy-wheel printers is 1/48 inch, while for most laser printers it is 1/300 inch. The VS (Vertical Spacing) table in the Printer File contains the printer codes to create increments of this minimum vertical movement.

Vertical Measurement Unit (VU). The next value you need to consider is the unit you are going to use for your vertical measurements. A typical vertical measurement unit is a *line*, which is equal to 1/6 of an inch. Some people prefer to work in *points*, which is equal to 1/72 of an inch. Once you decide on this unit (by specifying VU), you must use it for *all* commands that involve vertical measurements (including PL, FD, TP, BT, PG), not just for leading commands.

The VU (Vertical Unit) setting in the Printer File defines the number of minimum vertical movements the printer must make to equal one of the vertical measurement units you are using. For example, assume you are using an Epson printer that has a minimum vertical movement of 1/216 inch and you want to make your vertical measurements in lines (1/6 inch). The Epson printer must make 36 minimum vertical movements in order to move the desired 1/6 inch (1/216 x 36 = 1/6). If you choose to work in points (1/72 inch) rather than lines, the VU setting would be 3 (1/216 x 3 = 1/72).

Whenever you issue a vertical leading command, XyWrite takes the value you specify in the command and multiplies it by the VU value. The product of this calculation is the number of incremental steps (defined in the VS table) that are required to output the correct leading.

(The VU setting in the Printer File also contains values for screen display and precision. These values are discussed in Chapter 6.)

Automatic Vertical Leading (VL). The third vertical spacing setting in the Printer File is VL. This setting, which is part of the Font Table, defines two values: the total leading associated with each font and the amount of leading that is output before a line. VL is used in documents that contain text in a variety of point sizes.

Remember that when you install XyWrite, your Printer File is already set up with standard values for your printer — you don't have to worry about it. And calculations for the printer instructions are transparent to you — they happen in the background when you use the TYPE command. The descriptions presented above are designed to help you understand the kind of information that is stored in the Printer File so that you can use the Vertical Spacing commands effectively.

There are four Vertical Spacing commands described on the following pages:

- Automatic Leading (AL)
- Extra Leading (EL)
- Line Leading (LL)
- Line Spacing (LS)

Each of these descriptions uses ***lines*** as the vertical unit of measurement. Remember that a ***line*** is 1/6 of an inch. When we use the term to refer to this specific measurement, we italicize it to avoid confusion with number of lines of text.

We have chosen to use ***lines*** in our examples because they are the most commonly used vertical measurement unit on printers, but don't forget that you can modify your Printer File so that you can use other units of vertical measurement.

FORMAT **CM AL** *n* Automatic Leading

n is 0 (off) or 1 (on)
AL is an embedded command.

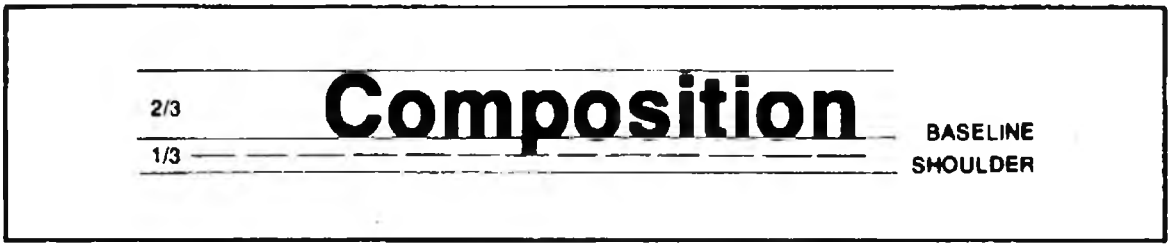
PURPOSE

The **AL** command turns automatic leading off and on. When you turn automatic leading *on*, the vertical spacing values are supplied by the Font Table in the Printer File. When you turn automatic leading *off*, the vertical spacing values are supplied by the LS (Line Spacing) command. The AL setting also affects the baseline calculations.

The *baseline* is an imaginary line on which the characters rest. When you are mixing point sizes within a document, the location of the baseline becomes critical; if you don't allow enough space between lines, the characters on one line touch the characters on the previous line. XyWrite automatically performs the baseline calculations for you based on the maximum leading value of each line. The formulas for the calculations differ slightly, depending on whether automatic leading is on or off.

When automatic leading is *on* (AL 1), XyWrite looks at the VL setting for every font in a line to determine the maximum leading for that line. The VL setting in the Font Table of the Printer File contains two arguments. The first argument defines the total leading associated with a font; the second argument defines the amount of leading that should be output before the line of text is printed.

Once XyWrite has determined the maximum leading associated with a line, it looks at the second argument in the VL setting. If the second argument is 0 or is omitted, XyWrite outputs 2/3 of the maximum leading value before the line of text and the remaining 1/3 after the line.



If the second argument in VL does have a non-zero value, that value defines the amount of leading that takes place before every line of text on a page. The value of this argument is then subtracted from the maximum leading value (which is defined by the first argument) to determine the amount of leading after each line of text.

When automatic leading is *off* (AL 0), XyWrite uses the value supplied by the LS command and multiplies it by the Vertical Unit (VU) defined in the Printer File to determine the amount of space output between lines. (See the Overview of this section for a description of VU). The VL setting is ignored.

To summarize, the AL command has two settings:

- AL 0 means that automatic leading is OFF and the LS command is honored.
- AL 1 means that automatic leading is ON and the vertical spacing values in the Font Table of the Printer File are honored. All LS commands are ignored.

ACTION **Selecting Automatic Leading.**
If you are using a laser printer and mixing point sizes within a document, you probably want to use automatic leading. That way, you don't have to worry about adjusting your line spacing to accommodate larger and smaller point sizes. To turn automatic leading on:

1. Move the cursor to the top of your document.
2. To turn AL on:

   Type: F5 al 1 ↵

Result: All text that follows the AL command automatically uses the leading values defined in the Font Table of the Printer File.

NOTE **Default Setting.** You can turn automatic leading on and off using the DEFAULT command in the STARTUP.INT file, in the Printer File, or on the Command line. The XyWrite default is 0 (off).

FORMAT

CM **EL** *n* — Extra Lead

n is the number of *lines* to insert after the current line. (This number can be expressed in decimals as small as hundredths.) In this context, a *line* is 1/6 inch.
EL is an embedded command.

PURPOSE

EL inserts the specified amount of vertical space only once, at the end of the current line. You must issue this command again for each line that needs extra lead.

ACTION

Inserting Extra Lead in a File.
To insert extra lead in a file:

1. Move the cursor to the line after which you want extra leading.
2. Type: F5 el 1.25 ↵

Result: When you print the file, an extra 1-1/4 *lines* of space will appear once, after the line where you inserted the EL command.

NOTE

Vertical Measurement Unit. This description uses *lines* as the vertical measurement unit. When used in this context, it means 1/6 of an inch, not one line of text. To avoid confusion, we have italicized the word when it is referring to the unit of measure.

FORMAT **CM LL *p,l*** Line Leading

p is the amount of extra space between paragraphs
l is the amount of extra space between lines of text
Both values are expressed in *lines*. The numbers can be expressed in hundredths (.01) of a line. In this context, a *line* is 1/6".
LL is an embedded command.

PURPOSE

LL allows you to automatically insert extra space between paragraphs. It also allows you to modify the leading values for a document without changing and reloading the Printer File. Note that the values you specify are *added* to whatever leading value is in effect.

ACTION

Inserting Extra Vertical Space.
To automatically insert an extra *line* between paragraphs and an extra 1/4 *line* between lines of text:

1. Move the cursor to the point where you want the extra vertical spacing to begin (normally between paragraphs).

2. Enter the LL command with values for extra spaces between paragraphs and between lines:

   Type: F5 ll 1,.25 ↵

Result: When you TYPE the document, the interparagraph spacing (spacing between paragraphs) will be increased by 1 *line* and the interline spacing (spacing between lines) will be increased by 1/4 *line*. This spacing remains in effect for the rest of the file or until you issue a different LL command.

NOTE

Vertical Measurement Unit. This description uses *lines* as the vertical measurement unit. When used in this context, a *line* means 1/6 of an inch, not one line of text. To avoid confusion, we have italicized the word when it is referring to the unit of measure.

FORMAT

CM **LS** *n* Line Spacing

n is the number of *lines* advanced by the printer (This number can be expressed in decimals as small as hundredths.) In this context, a *line* is 1/6".
LS is an embedded command.

PURPOSE

LS (Line Spacing) sets the number of *lines* advanced by a printer at the end of every line.

The LS command enables you to write a document single-spaced, but then very easily print it out double-spaced, triple-spaced, etc.

- LS .25 is quarter-line spacing
- LS .5 is half-line spacing
- LS 1 is single spacing (the default)
- LS 1.5 space-and-a-half spacing
- LS 2 is double spacing

ACTION

Setting the Line Spacing.
To set the number of *lines* of spacing output by the printer at the end of every line:

1. Move the cursor to the top of your document (or to the point you want to change the line spacing).
2. Enter the LS command along with the line-spacing value you want:

   Type: F5 `ls 2` ↵

Result: This embedded command tells the printer to double-space this document. Go ahead and use TYPE to print the document.

NOTE #1

Vertical Measurement Unit. This description uses *lines* as the vertical measurement unit. When used in this context, a *line* actually means 1/6 of an inch, not one line of text. To avoid confusion, we have italicized the word when it is referring to the unit of measure.

NOTE #2 **Default Line Spacing Setting.** The default is LS 1 unless you use the DEFAULT command to change it. You can set up the DEFAULT command in the STARTUP.INT file, in the Printer File, or on the Command Line.

NOTE #3 **On-Screen Line Spacing.** The LS command normally does not affect the spacing you see on the screen as you type. (The exception is column tables, which always show true line spacing.) If you want the display to reflect the current line spacing, enter the setting LF=1 in the Printer File. To restore the on-screen display to single spacing, enter LF=0. You can also use the DEFAULT command to set on-screen line spacing.

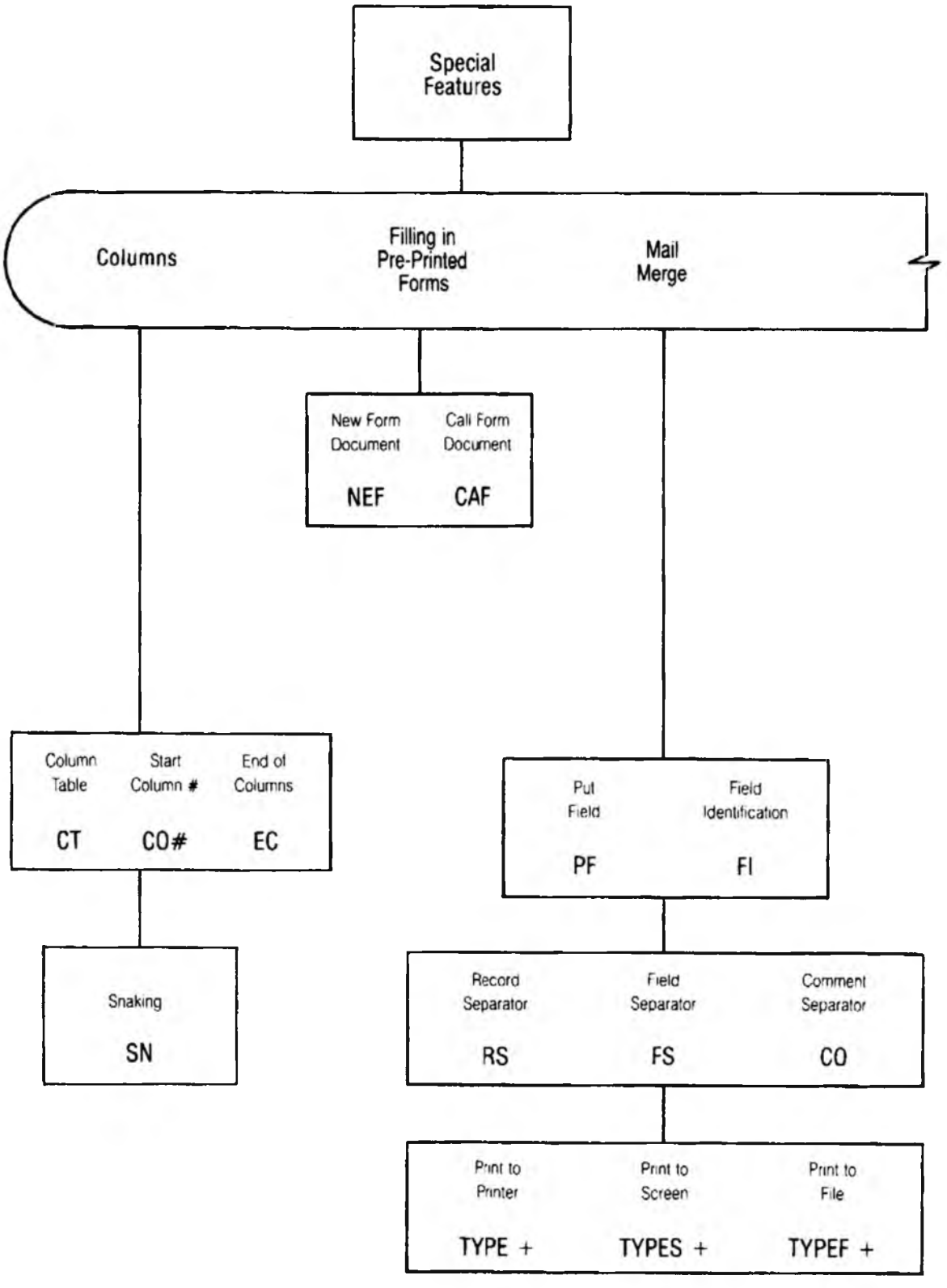
NOTE #4 **Automatic Leading.** When Automatic Leading is in effect (AL 1), XyWrite ignores LS commands and uses the line spacing information from the VL setting in the Font Table of the Printer File.

Special Features

INTRO

Once you've learned the basics, it's time to go on to the extended capabilities of XyWrite. We include step-by-step procedures for each of the following topics. Commands are listed following each procedure.

CONTENTS


Special Features
Columns
Filling in Pre-Printed Forms
Mail Merge
New Form Document
NEF
Call Form Document
CAF
Column Table
CT
Start Column #
CO#
End of Columns
EC
Snaking
SN
Put Field
PF
Field Identification
FI
Record Separator
RS
Field Separator
FS
Comment Separator
CO
Print to Printer
TYPE +
Print to Screen
TYPES +
Print to File
TYPEF +

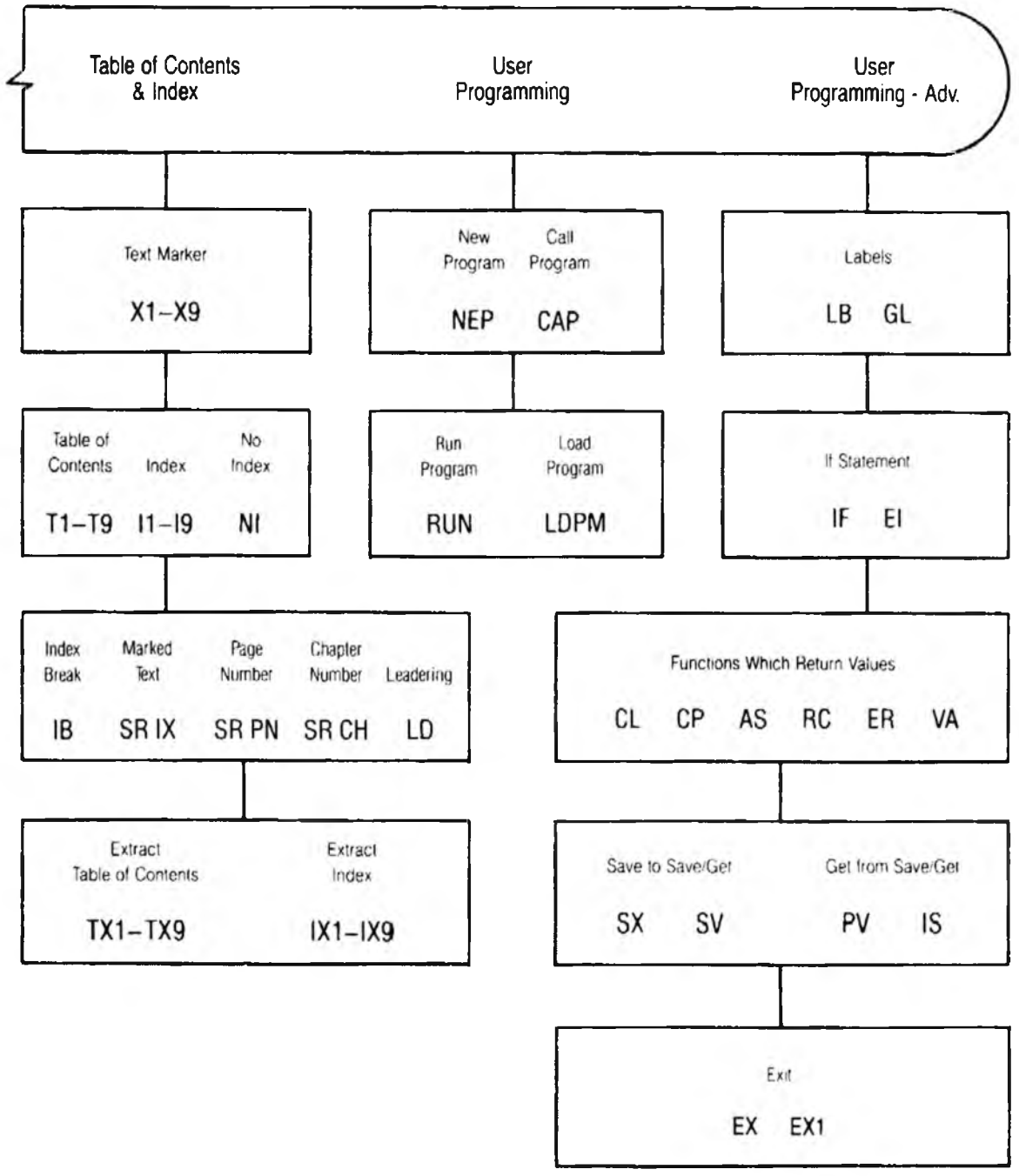


Table of Contents & Index
User Programming
User Programming - Adv.
Text Marker
X1–X9
New Program
Call Program
NEP
CAP
Labels
LB
GL
Table of Contents
Index
No Index
T1–T9
I1–I9
NI
Run Program
Load Program
RUN
LDPM
If Statement
IF
EI
Index Break
Marked Text
Page Number
Chapter Number
Leadering
IB
SR IX
SR PN
SR CH
LD
Functions Which Return Values
CL
CP
AS
RC
ER
VA
Extract Table of Contents
Extract Index
TX1–TX9
IX1–IX9
Save to Save/Get
Get from Save/Get
SX
SV
PV
IS
Exit
EX
EX1

NOTES

INTRO

The Columns feature of XyWrite provides two basically different kinds of column features: Column Tables and Newsletter Style Columns. In both cases, text within a column *word-wraps within its own column.* These two methods are designed for different purposes:

Column Tables. Think of this as a spreadsheet for text. You can create up to sixteen columns on-screen with as many text entries in a column as you want. (A text entry can be any length.) You can *add or delete text in any "cell" without disturbing any of the other cells.* This works great for handling blocks of text, such as in lists, calendars, appointment books, and even screenplays (each stage character gets his/her own column).

Newspaper-Style Columns. In this case *the text wraps from the bottom of one column to the top of the next*— hence the term "snaking." You create your file as a single column and print it in up to six columns. This is useful for printing long lists, reports or newsletters.

CONTENTS

Column Tables

PURPOSE

Suppose we want to set up a schedule of events in which we have the days stretched across the top and activities listed down the page. See the accompanying illustration.

Notice this table is made up of a number of text blocks, or "cells." This is very much like a spreadsheet for text. You can enter as much or as little text in each cell as you want without disturbing the other cells. The text within a cell is considered an "entry."

Making a Text Table. The way to make such a table is described in the next few pages. To summarize:

- **Create the Column Structure.** Use the CT (Column Table) command to define the column widths. You can also define a format style for each column.

- **Enter the Text.** Then you enter the text into the table. Certain keys allow you to add rows to the table and to move about, from cell to cell.

NOTE

Planning Your Table. Before you start, plan the number of columns that you need. If you know how many rows will be in the table, then add them, also, before starting the text entry (using Shift Ins). Details on this follow in the description of Editing Column Tables.

COLUMN TABLE

ROW OF TEXT

COLUMN OF TEXT

CELL

Time	Monday	Tuesday	Wednesday	Thursday	Friday
9:00 AM	Bus #1 from Atlanta	Breakfast	Breakfast	Breakfast	Breakfast
10:00 AM	Bus #2 from Charlotte	Paint the barn and mow the grass	Hike to top of Mt. Monadnauk	Swimming	Clean up the camp
11:00 AM	Assemble campers for orientation			Chop firewood	
12:00 AM	Lunch	Lunch	Lunch on top	Lunch	Lunch
2:00 PM	Swimming	Swimming	Find lost hikers	Pottery workshop Make ash trays	Swimming
4:00 PM	Counselors Practice CPR	Tennis	Swimming	Tennis	Load campers on the buses and move 'em out

FORMAT

CM CT *offset,width1,width2*, . . . (Option 1)

CM CT *offset,width1/style1,width2/style2*, . . . (Option 2)

offset shifts the horizontal position of the table from the left margin.

width1 is the width of the first column of text,
width2 is the width of the second column of text,
and so on up to 16 columns.

/style1 is the format style for the first column,
/style2 is the format style for the second column,
and so on up to 16 columns.

CT is an embedded command.

PURPOSE

The **CT** (Column Table) command sets up the basic structure for the column table. You define the column widths and, optionally, the styles for the columns.

The maximum number of columns you can specify is 16. Notice for the value *offset* that the left-most position (no offset) is 0 (zero). The amount of space between columns of text is 1 by default. You can increase this space with format commands — see "Creating a Column Table with Styles" which follows.

ACTION

(Option 1)

Creating a Column Table.

To insert a column table into a document, enter CT along with an offset for positioning the table horizontally, plus the width for each column. All numbers are measured along the ruler. For example:

Type: F5 ct 0,20,10,10 ↵

Result: This sets up a three-column table; the first column runs from 0 to 18, the second from 20 to 28 and the last from 30 to 38. Between the columns are gutters one-space wide at positions 19 and 29. (Notice that this command embeds *four* triangles into the text — CT, CO2, CO3 and EC. See Note #2.) To add rows of cells to the table, as in the previous illustration, press Shift Ins.

NOTE #1

Typing in Text. You do not need to set up the number of lines of text in each cell row beforehand. Simply enter text when you're ready; XyWrite automatically pushes down the lower boundary of the whole row to fit it.

ACTION

Setting the Format for Single Cells.
As in the previous procedure, the CT command establishes the overall column widths. In addition, if you wish you can set up an entire text format within an individual cell as you would with any document — using the LM (Left Margin), RM (Right Margin) or other format commands within the cell boundaries. Because each cell is independent of the others, other cells are unaffected. Let's see an example.

Once you have created a column table, to change the format within a single cell:

1. Move the cursor to the cell you want to change. To get there, use Shift ← or Shift → to move left or right.

2. Position the cursor within the cell at the point where you want to change the format. Enter the format command you want. For instance, in the previous example, the text width was 19. To make the text narrower, you can change the right margin within the cell:

   Type: F5 rm 15 ↵

Result: This changes the format within a single cell. To modify an entire column, follow the next procedure.

ACTION *(Option 2)*

Creating a Column Table with Styles.
To create a column table with styles:

1. Use SS to define the style you want within a column. When using columns, RM 10 means set the right margin ten spaces from the left edge of the cell. For example, create a style called BODY:

   Type: F5 ss body,lm=2,rm=10 ↵

2. Create the column table, assigning the style to the column(s) you want:

   Type: F5 ct 0,20,10,10/body ↵

Result: The style BODY is assigned to the third column of text.

NOTE #2

The CO and EC Commands. Entering the CT command automatically enters other commands which define the bounds of the columns. In the example, when you entered the CT command you also embed the commands CO2, CO3 and EC. If you count the cells going from left to right, row by row:

CT stands on the line preceding the table.
CO2 separates the first and second cells of each row.
CO3 separates the second and third cells of each row.
and so on...
CO1 separates the last cell in a row from the first cell in the next row.
EC indicates the end of the very last cell.

CO1 is present only if there is more than one row of cells.

NOTE #3

Entering the CO or EC Commands. While you can enter the CO and EC commands from the command line, you should not. The CT command does so for you automatically. You should instead use Shift Ins to add new columns.

NOTE #4

Inserting Column Tables Within Tables. XyWrite does not allow nesting of one column table within another.

FORMAT

Shift ←	Move Cursor One Column to the Left
Shift →	Move Cursor One Column to the Right
Shift Home	Move to Top of the Current Cell
Shift End	Move to Bottom of the Current Cell
Shift Del	Delete a Row of Cells
Shift Ins	Insert a New Row of Cells
Shift F1	Define the Current Cell

These are all immediate commands.

PURPOSE

You use these keystrokes to move around and edit within column tables. Refer to the previous illustration "Column Tables". Think of the column table as a grid of cells. The table is very similar to a spreadsheet, except the cells have variable length.

The text in any cell can run as long as you want — the cell will automatically grow in length to accommodate the text. (Other cells in the same row of cells will grow along with the longest cell.)

ACTION

Adding a Row of Cells to a Column Table.
To add a new row of cells to an existing column table:

1. Move the cursor to the row of cells *above* where you want to insert the new row.
2. Press: Shift Ins

Result: A new row of empty text cells is inserted below the current row. The cursor moves to the first position of the empty cell below. Without moving the cursor, you can begin typing text into that cell.

NOTE

Changing the Width of a Column. To change the width of a column in an existing column table, move the cursor to the embedded CT command. Press Ctrl F9 for Expanded Display and change the appropriate column width number. When you return to Normal Display, the column width is changed and the row depth altered to fit the text in the row.

Changing column size is easy; adding new columns is not. If you're unsure how many columns you'll need, it's best to over-estimate; empty columns will be ignored during printing.

TIP **Building a Column Table.** When building a table, it is useful to add all the rows you anticipate needing before you begin your text entry.

ACTION **Moving/Copying Text from One Cell to Another.** To move or copy text from one cell to another:

1. Move to the cell you want to move or copy.
2. Press: Shift F1
3. Move to the target cell.
4. Press F7 to copy or F8 to move the text.

NOTE #1 **Defining Text Within Cells.** You can use F1 (define a block of text) or F4 (define by line) or Alt F4 (define by word), but defining by sentence and paragraph is restricted.

NOTE #2 **Using Column Functions Outside a Table.** If you use the column functions outside the column table, they have no effect. These functions can be used only when the cursor is between the pair of commands CT (Column Table) and EC (End Column).

NOTE #3 **Simple Columns with Hanging Indent.** If you have a one-line header or number to the left of a paragraph (such as the style of this note), it may be easier to use the IP (Indent Paragraph) command rather than the CT command. For example:

1. Move to where you want to begin.
2. Type: F5 ip 0,16 ↵
3. Type: F5 ts 16 ↵

Result: You can start each paragraph with a heading (of up to 15 characters) followed by a Tab — the paragraphs hang at column 16.

FORMAT

CM SN *c1,c2,c3,c4,c5,c6* Snaking

c1 is the starting position of 1st column of text,
c2 is the starting position of 2nd column of text,
and so on, up to *c6*.
, (comma) separates the values.
This is an embedded format command.

EXAMPLE

CM sn 5,40

PURPOSE

The SN (Snaking) command sets up single-column text to print in multiple columns (up to 6) on a single page. The unique property here is the text wraps from the bottom of each column to the top of the next.

See the accompanying illustration. For example, suppose you have a telephone list of 300 people that would take 6 pages to print out as one long, narrow list. You can use the SN command to print this list as 6 columns on one page.

Or maybe you want to print a newsletter with two or more columns side-by-side. the SN command enables you to do that, too.

When you specify more than one column, you will see only one long column on the screen, but the document prints as multiple columns.

ACTION

Snaking Columns.

To print text with snaking columns: (1) Move to the top of the text (for Steps 1 and 2).

1. Set the right margin with RM. This sets the width of the *text* within the borders of the columns. For example:

   Type: F5 rm 16 ↵

2. Set up the columns using the snaking command SN. To create columns of text starting at positions 10, 30, and 50:

   Type: F5 sn 10,30,50 ↵

3. Print the text (with TYPE) or send it to a file (with TYPEF) to see the multiple columns:

   Type: F5 type ↵

Result: The printout has three columns, with a 4-space gutter between columns (since the columns are 20 wide and the text is 16 wide).

NOTE #1 **Spacing Between Columns.** If the spacing between columns is too tight, you can either (1) make the text narrower by moving the margins in with LM or RM (which probably makes the text longer), or (2) make the columns wider by changing the starting positions of your columns within the SN command.

NOTE #2 **Column Breaks.** When you want to break a column, use the PG (Page Break) command. This causes the text which follows to start at the top of the next *column* (not the next page). If you are in the last column, the text starts on the next page.

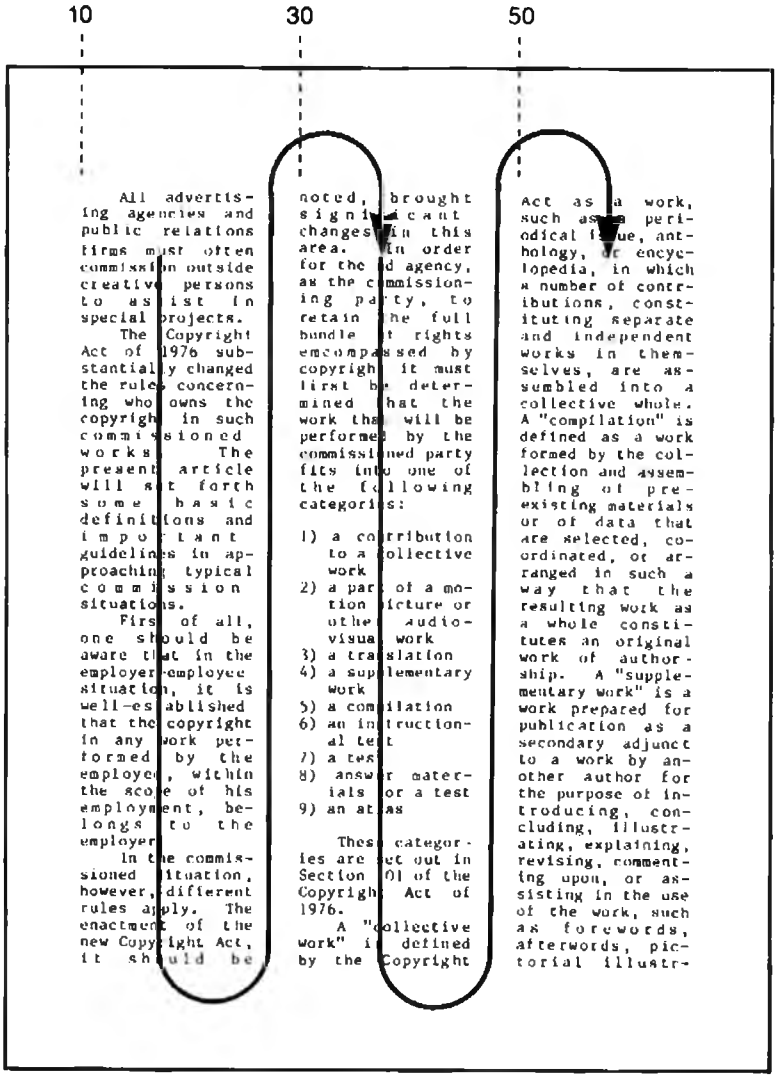
NOTE #3 **Controlling Column Length.** XyWrite uses the PL (Page Length) values for controlling the text length within the columns. Each column is treated as a separate page. In addition, the Widow and Orphan settings are also observed when determining the text length.

NOTE #4 **Footnotes**. You can add footnotes to your text. They print out at the foot of the column they are referenced within. Footnotes are counted in the Page Length for the column unless the BF (Bottom Footnote) was previously set to 1.

NOTE #5 **Full-Page Requirement.** You cannot start or stop snaking columns in mid-page. In other words, you cannot include straight text on a page that includes snaking columns.

SNAKING COLUMNS OF TEXT – PRINTOUT

Example: SN 10,30,50



All advertising agencies and public relations firms must often commission outside creative persons to assist in special projects.

The Copyright Act of 1976 substantially changed the rules concerning who owns the copyright in such commissioned works. The present article will set forth some basic definitions and important guidelines in approaching typical commission situations.

First of all, one should be aware that in the employer-employee situation, it is well-established that the copyright in any work performed by the employee, within the scope of his employment, belongs to the employer.

In the commissioned situation, however, different rules apply. The enactment of the new Copyright Act, it should be noted, brought significant changes in this area. In order for the ad agency, as the commissioning party, to retain the full bundle of rights encompassed by copyright it must first be determined that the work that will be performed by the commissioned party fits into one of the following categories:

1) a contribution to a collective work
2) a part of a motion picture or other audio-visual work
3) a translation
4) a supplementary work
5) a compilation
6) an instructional text
7) a test
8) answer materials for a test
9) an atlas

These categories are set out in Section 101 of the Copyright Act of 1976.

A "collective work" is defined by the Copyright Act as a work, such as a periodical issue, anthology, or encyclopedia, in which a number of contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A "compilation" is defined as a work formed by the collection and assembling of pre-existing materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship. A "supplementary work" is a work prepared for publication as a secondary adjunct to a work by another author for the purpose of introducing, concluding, illustrating, explaining, revising, commenting upon, or assisting in the use of the work, such as forewords, afterwords, pictorial illustr-

NOTES

Fill-In Forms

INTRO

XyWrite provides you with the ability to fill in pre-printed forms — or create your own forms to fill in. Typical forms are questionnaires, surveys, tax forms, credit applications, and invoices — any sheet that needs to be filled in with information.

This section begins with a general procedure for filling in pre-printed forms and then describes the commands.

CONTENTS

Fill-In Forms Procedure

PURPOSE

The best place to start is by looking at the illustration on the next page. A typical scenario is this:

Filling in a Pre-Printed Form. You might start with a pre-printed form like the one shown. You wish to feed this into your printer and fill in the blanks using XyWrite. This procedure is easy:

- **Create the Form File.** You create a Form File to match your pre-printed form.
- **Fill In the Form File.** To fill in the pre-printed form, you type into the blanks of the Form File you created. You then print onto your pre-printed form.

ACTION

Creating the Form file.
To create a Form File, do the following:

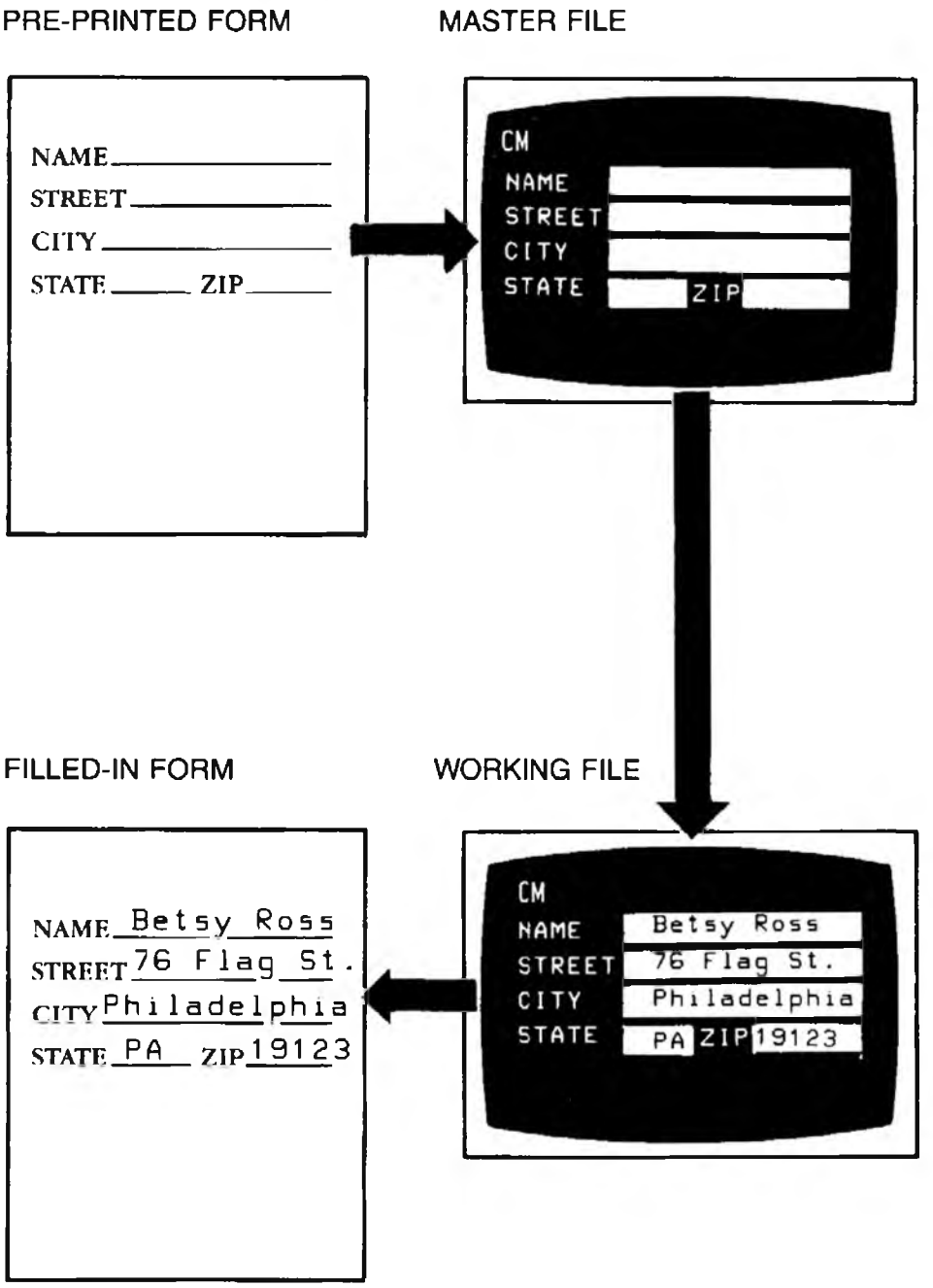
1. **Print the Grid.** (Optional) Print the file GRID.FRM onto an extra copy of your pre-printed form (See Note #2 — GRID.FRM is provided on your original XyWrite disk.) Do this as follows:

   Insert an extra copy of your pre-printed form into your printer. *Note the position of the form as you insert it*, so that later you can insert other pages the same way.

   Type: F5 type grid.frm

   Result: A grid pattern of numbers overwrites the pre-printed form, giving you coordinates for Step 3. (If the pre-printed form has non-standard vertical spacing, you can insert printer control codes directly into GRID.FRM to make the pages coincide.)

FILLING IN A PRE-PRINTED FORM



2. **Start a New File.** Open a new file. Let's call it MASTER:

   Type: F5 new master ↵

3. **Duplicate the Pre-Printed Form.** Now duplicate the pre-printed form by typing it into your computer as follows. Using the page you printed in Step 1 as a guide, enter the text and fill-in areas at the lines and columns indicated:

   - Enter the fixed text in Reverse Mode. For example, to enter the word NAME:

     Press: Ctrl 4

     Type: Name

   - Enter the blank fill-in areas in Normal Mode. For example:

     Press: Ctrl 1

     Press: Space Bar

     Press the Space Bar as many times as you need to establish the length of the fill-in area. Do *not* use the Tab key (it will not work properly with NEF or CAF).

Repeat the previous two items until you have entered all the text and fill-in areas. When you have finished the Form File, store it:

Type: F5 store ↵

Result: You are now done creating a blank fill-in form. This is our master template.

ACTION

Filling in the Form

To fill in a pre-printed form using the Form File you produced earlier (in "Creating the Form File"):

1. **Create the Working Form.** Make a working copy of the Form File. Let's call this copy WORKFORM.

   Type: F5 `nef workform,master` ↵

   Result: You have created a new form with spaces waiting for you to fill in. Notice that the cursor is confined to move only in the fill-in areas.

2. **Fill In the Blanks.** Fill in the blanks of this working copy by typing into them. For instance:

   Type: `Betsy Ross`

   Result: This Form File looks like:

   ```
   NAME Betsy Ross          ←
   ADDRESS                  ←
   STATE        ZIP         ←
   ```

   Notice the words "NAME", "ADDRESS", "STATE", and "ZIP" are fixed text; you cannot alter them. You move the cursor from one field to the next with the ↵ (Enter) key.

3. **Print the Forms.** Finally, insert the pre-printed form into your printer. Be sure to position the paper the same as you positioned the page you printed the grid onto (in "Creating the Form File"). Print the information from the working form:

   Type: F5 `type` ↵

   The fixed text will not print; only the information entered in the blanks (during Step 2) is printed.

NOTE #1 **Edit a Form File.** If you should ever need to call up a Form File to change only the text in the fill-in areas, use the CAF command. For example, if the form is named WORKFORM:

Type: F5 caf workform ↵

NOTE #2 **Using the File GRID.FRM.** To locate the fill-in areas of your pre-printed form, we provide a file called GRID.FRM. You print the following grid onto an extra copy of your pre-printed form:

```
Line 1 78901234567890123456789012345678901234567890
01234567890123456789012345678901234567890123456789
Line 3 78901234567890123456789012345678901234567890
01234567890123456789012345678901234567890123456789
```

NOTE #3 **Carriage Return.** If you enter the carriage return ↵ in Normal Mode, the length of the blank area will be flexible, adjusting to the length of the text being entered — even if it is several lines. If you enter the carriage return in Reverse Mode, it *fixes* the length of the blank area.

NOTE #4 **Create Your Own Forms.** In the earlier procedure, the fixed text did not print. You can cause the fixed text to be printed by using Ctrl 2 (Boldface) or Ctrl 3 (Underline) for fixed text instead of Ctrl 4 (Reverse). When you print the Form File, the fixed text will print out in the Character Mode you choose — either bold or underline.

NOTE #5 **Printer File Requirement.** All XyWrite Printer Files come prepared for printing on pre-printed forms — they are set up to *not* print any text displayed in the Reverse Mode. (Each character displayed in Reverse Mode is printed as a space.) Your Printer File is originally set up with the following statement:

```
MD RV* ←
```

As usual, if you remove or re-insert this statement, the change doesn't take effect until you re-load the printer file with LDPRN.

FORMAT

CM **NEF** *d:workform,d:masterform*

workform is the name you specify for the new form.
masterform is the master Form File that you want to copy from.
NEF is an immediate command.

PURPOSE

NEF (New Form) creates a working copy of the master Form File. This copy is a new, blank form, ready for you to fill in.

NEF functions the same as NEW, except NEF prevents you from altering any text which is not in Normal mode including Reverse, Underline, or Bold — this is fixed text.

ACTION

Starting a New Fill-In Form from a Master

To start a new fill-in form (say, INVOICE1) from a master (called MASTER):

Type: F5 `nef invoice1,master` ↵

Result: This creates a new file called INVOICE1 and copies the contents of MASTER into it. XyWrite will not create a new file if one with the same name already exists on the specified drive.

FORMAT

CM **CAF** *d:workform*

workform is the name of the form.
CAF is an immediate command

PURPOSE

CAF (Call Form) calls a Form File from the disk to the display for changing or adding to the information in the fill-in areas.

Any fixed text (Reverse, Underline, or Bold) cannot be altered. (To modify fixed text, use the CALL command instead.)

If you prefer, you can use EDF (Edit Form) instead of CAF; EDF is just another name for CAF.

ACTION

Calling a Working Form

Call the Form File you want to fill in. For example, if its name is TAXFORM:

Type: F5 `caf taxform` ↵

Result: This file is ready for you to revise in the fill-in areas. (Notice the cursor is confined to the fill-in areas.)

NOTE

Any File is a Working Form. Technicaliy, *any* file can be called up as a working form with CAF. When it is, the cursor moves only in areas that are displayed in Normal mode — not into areas that are Reverse, Underline or Bold. This points out that what confines the cursor is the *manner* in which you call the file (CALL or CAF), rather than anything special about the file itself.

Mail Merge

INTRO

This section fully describes the Mail Merge features of XyWrite. We begin this adventure with an overview and then a complete example. You can model your own mail merge application after this example, and should have to read little else of this section.

CONTENTS

Mail Merge Procedure

PURPOSE

What Is Mail Merge? Mail Merge enables you to create multiple versions of a document more easily than you could create them manually, one-by-one. You could, for instance, create a dozen individual letters from a general form letter (Main File) and a dozen addresses (Data File). To print the dozen letters, you issue a single command (TYPE+), which merges the addresses and form letter as it prints them.

Why Use Mail Merge? Mail Merge removes the tedium of typing many versions of a document. Mail Merge is useful when creating many nearly identical documents which are personalized at critical points such as name, address and salutation. These can include letters, contracts, notices, and statements.

Mail Merge requires you to learn only two new commands: Put Field (PF) and Field Identification (FI). The following instructions should allow you to use Mail Merge when you want to, not just when you need to.

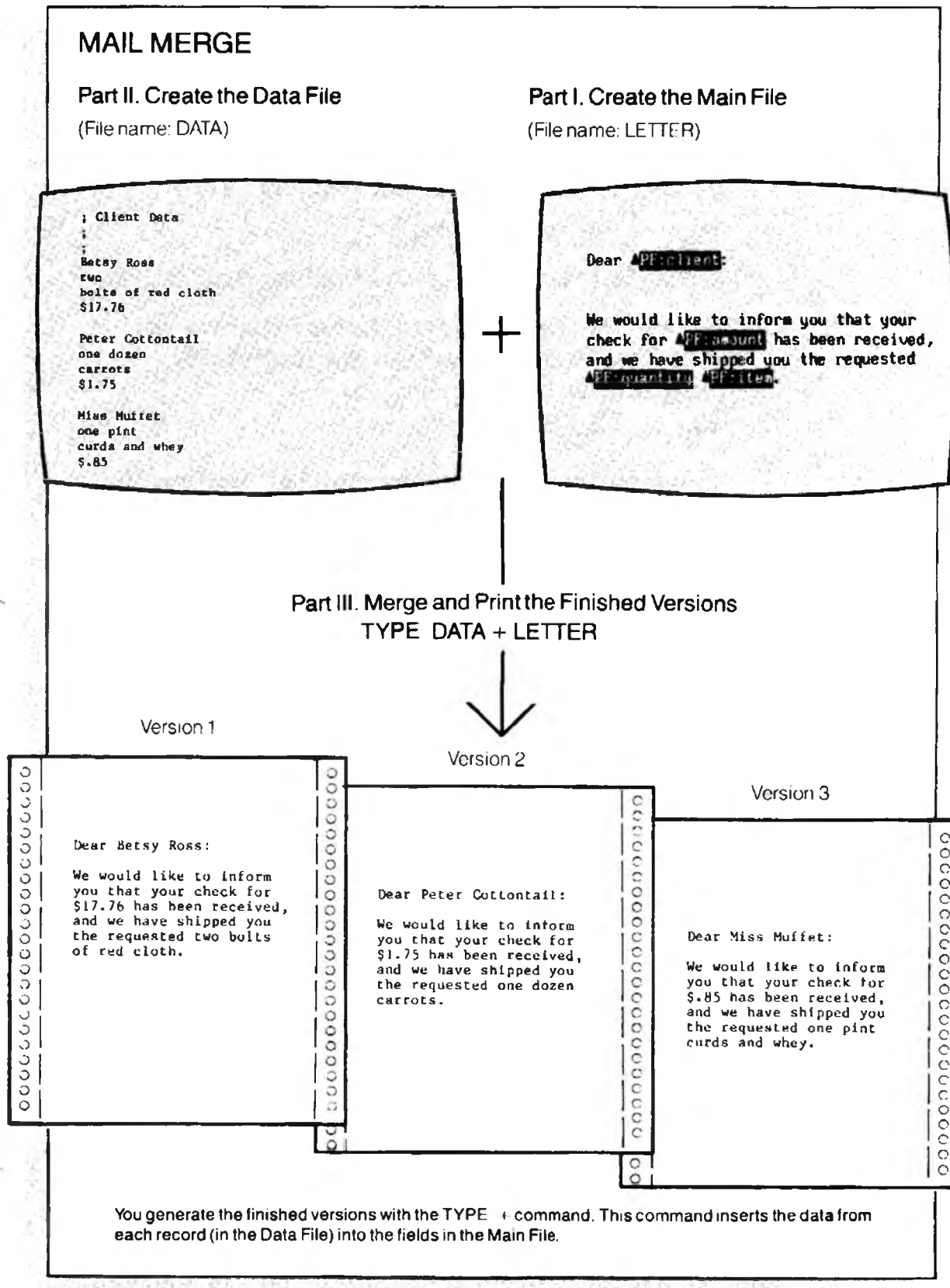
The procedure for Mail Merge has three parts, as illustrated in the accompanying figure.

- Part I. **Create the Main File** — this is the form letter — it contains the text which is common to all finished versions, along with fields that indicate where records data should be inserted.

- Part II. **Create the Data File** — it contains the records. This information is inserted into the fields of the Main File to make each version individualized.

- Part III. **Merge and Print the Finished Versions** — This is where you combine and print the file created in the two previous steps. You use only the simple command:

  TYPE *datafile+mainfile*



You generate the finished versions with the TYPE ‹ command. This command inserts the data from each record (in the Data File) into the fields in the Main File.

ACTION

Creating Mail Merge Documents

This procedure has three separate parts to it, which we list here as one sequence.

PART I

1. **Create the Main File.** To begin, open a file and give it a name (we'll use the name LETTER):

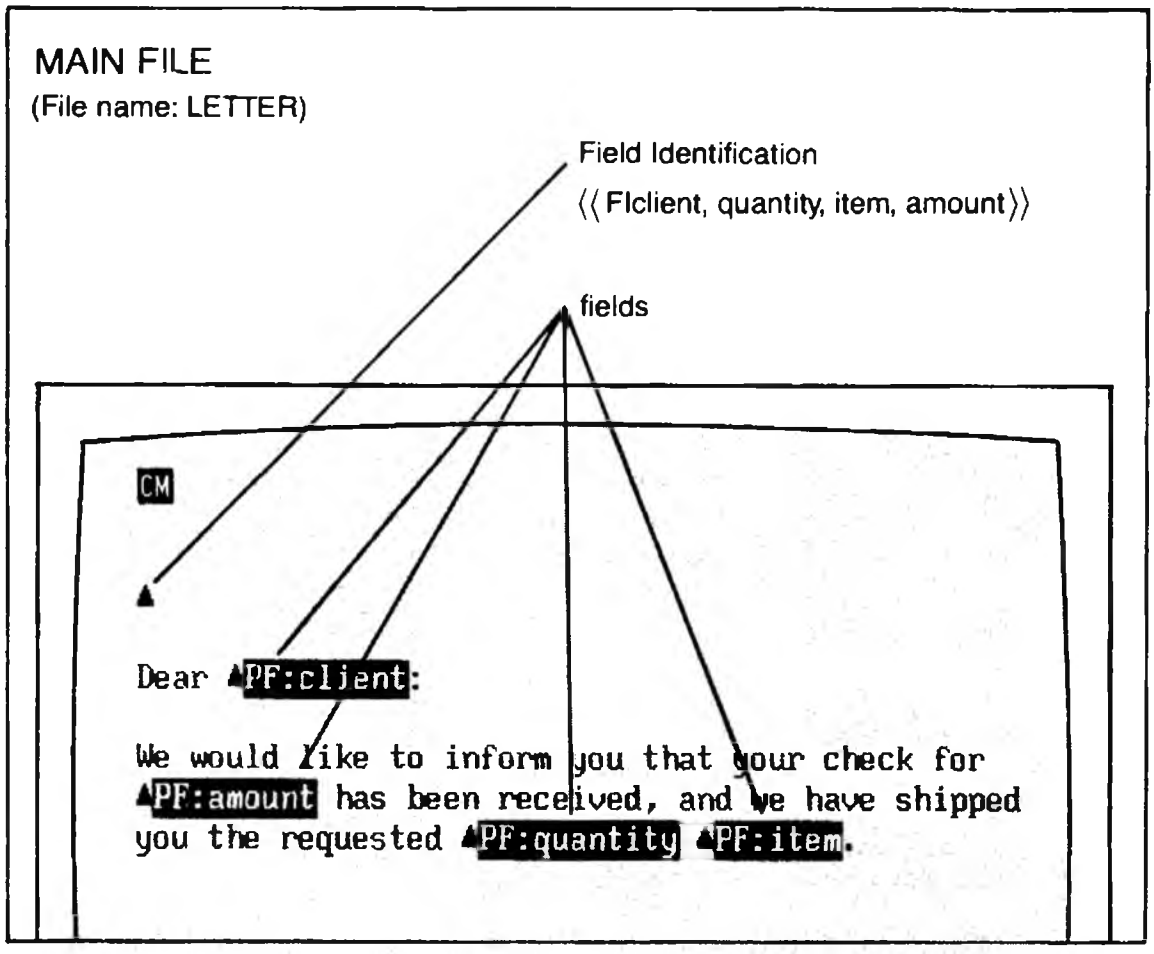
   Type: F5 new letter ↵

   Write the text you want, as in the illustration below.

   Use PF (Put Field) to place fields wherever you want data to be inserted. To insert the field "client", for instance:

   Type: F5 pf client ↵

   You may use a field in more than one place within the letter.



2. **Add the Field Identification.** This command links the field *names* (created in the previous step) to the order of fields in the Data File.

   Move the cursor to the top of your letter, anywhere ahead of the first field.

   Enter the FI command and the field names in the order that they will appear in the Data File (rather than their order in the letter). This may take some forethought — you won't be creating the Data File until Step 5 below.

   Type: F5 `fi client,quantity,item,amount` ↵

   Result: This FI command is embedded in the letter as a triangle.

3. **Store the Main File.** You have now completed the Main File, so let's store it.

   Type: F5 `store` ↵

PART II

4. **Create the Data File.** Decide on a name for yourData File. We'll use the name DATA, since it will contain our client's data. (This data will be inserted later into the fields of the Main File we just finished writing.)

   Type: F5 `new data` ↵

5. **Type in the Data.** Our example starts with the comment "Client Data" (a semi-colon at the start of a line makes it a comment). Here is one record:

   Type:
   ```
   ;Client Data↵
   ;↵
   ;↵
   Betsy Ross↵
   two↵
   bolts of red cloth↵
   $17.76↵
   ```

When entering the data, use these rules:

- Type each *field* of data on a separate line; follow each line of data immediately with a carriage return, without extra spaces.
- End the last field in a *record* with two carriage returns — this puts a blank line between records. (When printing, each record results in another version of the letter.)
- Identify each comment line with a leading semicolon (;). Any such line is ignored later by the program.
- End the the last line of data with exactly a single carriage return (no more, no less) after the last entry. (Don't end with two carriage returns, or it assumes another record is to follow. Don't omit the single carriage return, or the last field is ignored.)

6. **Store the Data File.** You have now completed the Data File, so let's store it.

   Type: F5 store ↵

DATA FILE

; Client Data	comment	
;		
;		
Betsy Ross	field 1	
two	field 2	record 1
bolts of red cloth	field 3	
$17.76	field 4	
Peter Cottontail	field 1	
one dozen	field 2	record 2
carrots	field 3	
$1.75	field 4	
Miss Muffet	field 1	
one pint	field 2	record 3
curds and whey	field 3	
$.85	field 4	

PART III

7. **Preview the Finished versions.** If you wish, you may display the three finished versions of the letter on the screen, before printing them on paper. Use the command:

   Type: F5 types data+letter ↵

   where DATA and LETTER are the names of the Data File and Main File created in Parts II and I, respectively.

8. **Print the Finished Versions.** If the displayed versions look fine, then print them with the command:

   Type: F5 type data+letter ↵

   A different version of LETTER is printed for each record it finds in the Data File. The following is the first of the three finished versions.

FINISHED VERSION

```
Dear Betsy Ross:

We would like to inform you that your
check for $17.76 has been received,
and we have shipped you the requested
two bolts of red cloth.
```

NOTE #1 **Importing Data.** The Data File could just as well originate from some other mail list manager or data base program, such as dBase III or R-Base, as long as it's an ASCII file. If it's not already in the format of one field to a line, you can either: (1) modify the Mail Merge Separators to accept the data, or (2) do a global search-and-replace on the data file to make it conform to the Mail Merge format.

NOTE #2 **Changing the Separators.** If you have a special application, you can change the characters which separate fields, records and comments. You would use FS (Field Separator), RS (Record Separator), and CO (Comment Separator) in your Printer File. These are described later in this section.

FORMAT

CM **PF** *field* Put Field

field is a name or number (a name is generally preferred over a number).
PF is an embedded command.

PURPOSE

The **PF** (Put Field) command places a field within the text of the Main File. Refer to the illustrations earlier in this section for examples. You enter the PF command into the Main File, specifying either a number or a name:

CM`pf client` Creates a field named "client" in the Main File.

CM`pf 1` Creates a field in the Main File. This field receives its data from the first field in each record. Thus, the number refers to the position of the field in each record.

ACTION

Inserting Fields into the Main File
To insert a field into the Main File:

1. As you type the Main File, when you reach the point in the text where you want to place a field, enter the PF command. Select a general name which identifies the field, such as "client", "item" or "quantity"; for example:

   Type: F5`pf client`↵

   Result: This command is embedded in the file as ▲`PF:client`. On printout, the effect of the PF command is as follows (using the example given in the previous Mail Merge Procedure): The field named "client" is replaced in successive documents by: Betsy Ross, Peter Cottontail, and Miss Muffet.

2. When using a field name, be sure to include that name in a Field Identification command at the beginning of the Main File.

NOTE #1 **Field Name.** Field *names* are generally preferred over field *numbers* because they can make the Main File far easier to read. You can use field names that describe what the field is. See the illustration below for an example. When a field name is used, the FI command must also be used to link those names to the fields in the data records.

NOTE #2 **Field Number.** Use the field number when it is easier to refer to a field by its numbered place in the record (than to assign it a name). However, field numbers generally make a document harder for others to read than a name would. The illustration below makes a comparison between field names and field numbers.

FIELD NAMES

```
CM
▲

Dear ▲PF:client:

We would like to inform you that your check for
▲PF:amount has been received, and we have shipped
you the requested ▲PF:quantity ▲PF:item.
```

FIELD NUMBERS

```
CM
Dear ▲PF:1:

We would like to inform you that your check for
▲PF:4 has been received, and we have shipped
you the requested ▲PF:2 ▲PF:3 .
```

The fields obtain their numbers from the order in which they appear in the Data File. Thus, a record to fit the previous illustration (Field Numbers) would have its fields listed in the following order:

client
quantity
item
amount

Using field numbers is a shortcut, since no Field Identification (FI) command is needed. The Field Identification command is needed only when using field names.

The previous example using field numbers is given simply to illustrate the equivalence of field numbers and field names. A more practical example of field numbers would be one where you were using only a few fields from a record of enormous length. It might be easy to refer to an item in the record as field number 45. (The alternative would be to give field 45 a name with the Field Identification command; however, the FI command would require 44 commas ahead of the specified field name.)

FORMAT

CM **FI** *field1 field2, . . .* Field Identification

field1 **is the name you specify (with the PF command) for the first field in each record,**
field2 **is the name you specify (with the PF command) for the second field in each record, and so on.**
FI is an embedded command.

PURPOSE

FI (Field Identification) assigns descriptive names to fields for use in the Main File. It links the order of fields in each record to the names used in Put Field commands in the Main File. FI is required if you use field names, but not if you use field numbers.

ACTION

Entering the Field Identification

Insert the FI command into the Main File as follows:

1. Position the cursor at the beginning of the Main File, ahead of any Put Field command.
2. Enter FI along with all fields used in the Main File, but ordered as they appear in the Data File. Using the example given in the earlier Mail Merge Procedure:

   Type: F5 `fi client,quantity,item,amount` ↵

   Result: This command is embedded in the text, displayed as a triangle. The above statement would be the proper Field Identification statement for the following record:

   ```
   Betsy Ross              (client)
   two                     (quantity)
   bolts of red cloth      (item)
   $17.76                  (amount)
   ```

   Notice that the field names are listed in the order that the fields appear in the record, rather than the order they appear in the Main File. These four names are the names which you have used in the Main File — names which identify the fields.

NOTE #1

Skip a Name. All fields in a record need not be named in the FI command. You can skip a name by leaving its position blank but keeping the commas. If, in the previous example, your letter made use of the first and third fields but not the second and fourth, you could specify:

```
CM fi client,,item
```

NOTE #2

Field Numbers. A Field Identification is not required in a Main File if you refer to the fields by number rather than by name. Refer to the Put Field command for further details.

TYPE + Printing Mail Merge Documents

FORMAT

CM TYPE *datafile*+*mainfile*,,P	To the printer
CM TYPES *datafile*+*mainfile*	To the screen
CM TYPEF *datafile*+*mainfile*,*targetfile*	To a file

datafile is the Data File (containing the records).
mainfile is the Main File (form letter).
,,P (optional) causes the printer to stop after each page; resume with +.
These are immediate commands.

PURPOSE

TYPE + merges data from records into a Main File and outputs the results to a printer. The records are listed in a Data File; one document is printed for each record. See the previous Mail Merge Procedure for illustrations. These three commands operate like the normal TYPE, TYPES and TYPEF commands you are likely already familiar with.

If the Data File is currently displayed, you can omit the *datafile* name and type:

TYPE +*mainfile*

Similarly, if the Main File is currently displayed, you can omit the *mainfile* name and type:

TYPE *datafile*+

If a block of text is defined, you can use that as the Data File.

ACTION

Printing Mail Merge Documents

When you are ready to print the Mail Merge versions, do the following:

Enter TYPE using the format given above. For example, with a Data File named DATA and a Main File named LETTER:

Type: F5 `type data+letter` ↵

Result: This statement merges the records from DATA into the LETTER as it prints. Once printing begins, you are free to continue editing other files while printing continues in the background.

FORMAT

FS<*separator*	Field Separator
RS<*separator*	Record Separator
CO<*separator*	Comment Separator

separator is the new string of characters.
FS, RS and CO are Printer File settings.

PURPOSE

FS, **RS** and **CO** re-define the characters which separate fields, records and comments in a Data File. You enter these definitions into the Printer File. All characters between the less-than sign (<) and the carriage return become defined as the new separator.

A typical use is as follows: You want to allow your Data File to use a field named "address" which inserts *three* lines of data (street, city, state). The default Field Separator (carriage return) would not allow this — it prevents a field from being more than one line. You could re-define the field separator to be a slash (/) — that would free up carriage returns to be used within the field.

FS (Field Separator) re-defines the separator between *fields* in the Data File. The default is a carriage return/line feed — each field on its own line.

RS (Record Separator) re-defines the separator between *records* in the Data File. The default is two carriage return/line feed combinations — records are separated by a blank line.

CO (Comment Separator) re-defines the character used to designate comment lines in the Data File. The default is a semi-colon (;).

ACTION **Changing the Mail Merge Separators.** You can enter all three separators using this same procedure.

1. Call your Printer File; for example, 3EPSONFX.PRN:

   Type: F5 call 3epsonfx.prn ↵

2. Enter on its own line FS<, RS< or CO< followed by the new character(s) you want to define. For example, to make the Field Separator a slash (/):

   Type: fs</ ↵

3. Type: F5 store ↵

4. Type: F5 ldprn 3epsonfx.prn ↵

Result: Step 4 loads the new separator into memory, in order for it to take effect. See Note #2.

NOTE #1 **Printer File.** You enter FS, RS, and CO into the Printer File. Each of these must appear on its own line in the Printer File. After you change a value, you must reload the table using the LDPRN command (Step 4 above).

If you use different Mail Merge separators for different Data Files, you might create a "Printer File" which contains nothing but the three field separators. Keep it with its Data File, and load it only when needed (using LDPRN). You might call it SEPARATR.PRN (without the "O", to keep it eight letters).

The FS, RS, and CO commands may not be included in the Printer Files on the original XyWrite disk. These commands do not need to be explicitly entered into the Printer Table until you want to change them. (The default definitions are internal to XyWrite.)

NOTE #2 **Example Records.** A record which uses the default field separator looks like:

Betsy Ross
$17.76
two
bolts of red cloth

If you changed the field separator to a slash (/), the record would now appear like:

Betsy Ross/$17.76/two/bolts of red cloth

NOTE #3 **Comment Line.** A comment takes up an entire line; that is, you cannot put a comment on the same line as a field, nor can you insert a comment into a field or record. In other words, a comment is recognized only before the first field or after any field or record has been formatted. A comment line begins with the semicolon (;) unless you use CO to change it.

NOTE #4 **Carriage Return / Line Feed.** To use a carriage return or line feed as part of the definition, you must enter them as follows:

- Carriage Return: Alt Shift 13
  This code appears as a ♪
- Line Feed: Alt Shift 10
  The code appears as a ◙

NOTE #5 **Typical Uses.** The Mail Merge Separator commands may be useful when you transport a data file from another program to be used as a Data File for Mail Merge. For example, if the Data File you transport uses different field separators, simply redefine FS in your Printer File. This way you can change Mail Merge to accommodate the Data File rather than vice versa.

You may also use these commands to allow a field to be more than one line; that is, change the Field Separator (FS) to allow a field to contain a carriage return.

Table of Contents and Index

INTRO

With long documents such as detailed reports and books, at times you will want to include a Table of Contents or Index. In this section we discuss how to generate them. This discussion begins with basic procedures and ends with a reference of all the commands.

CONTENTS

Table of Contents & Index Procedure (cont'd)

PURPOSE

The following procedures will enable you to generate a Table of Contents or Index in a step-by-step fashion. The procedure is in three parts and begins after a general overview and introduction of our terminology.

We use the term Table of Contents to mean any list that is sorted by *page number*. We use Index to mean any list sorted *alphabetically*. XyWrite will produce a Table of Contents or Index for you from the text that you mark in your document. There are three basic parts to generating an Index or Table of Contents. Refer to the following illustration.

- Part I. **Mark the Text** — Mark the words or phrases you want included in the Index or Table of Contents. Use one of the text markers (X1 through X9).

- Part II. **Specify the Format** — Specify the format you want using an Index command (I1 through I9) or a Table of Contents command (T1 through T9).

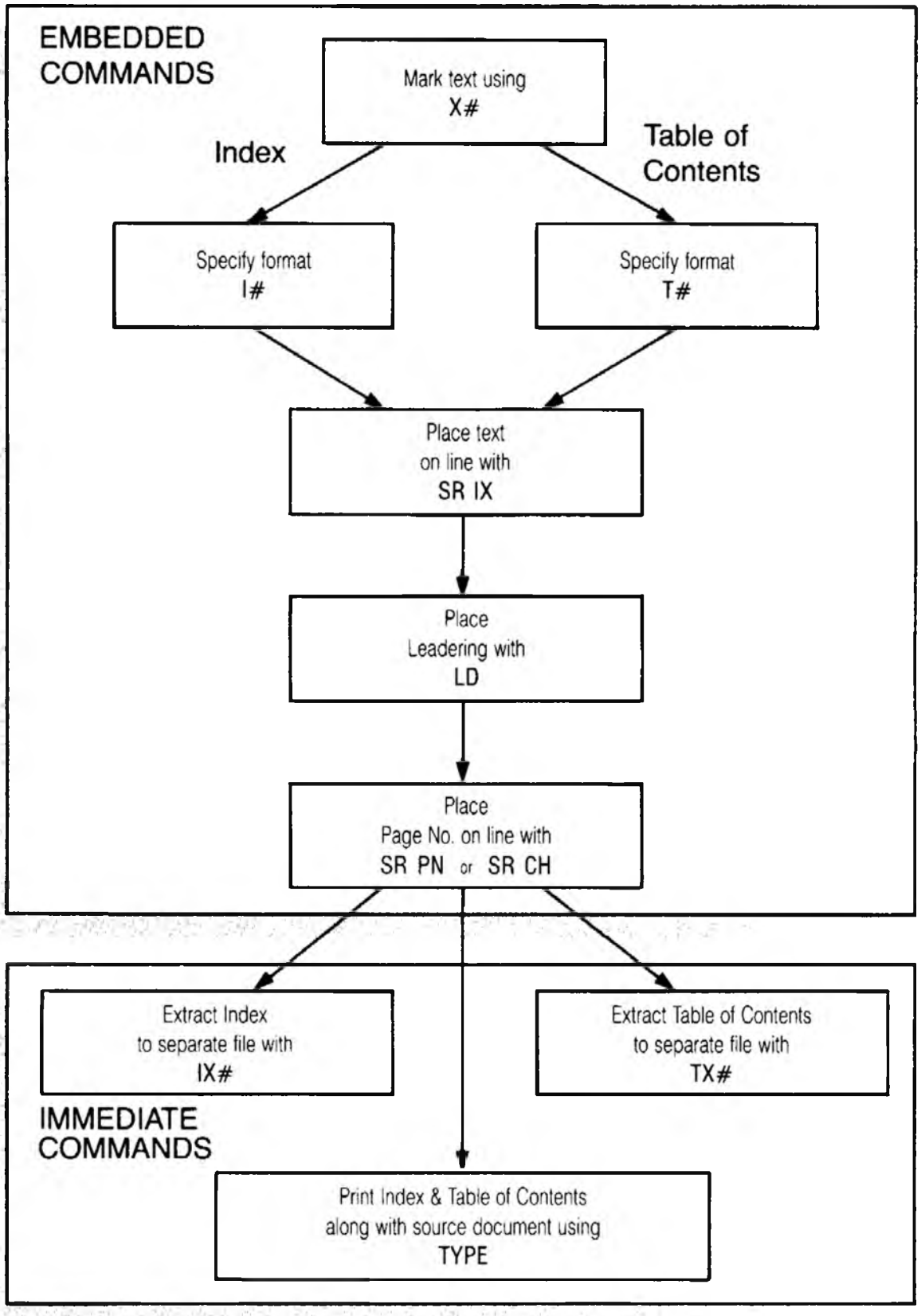
- Part III. **Extract the Table of Contents or Index** — Extract the marked text and either save or print it:

  a. Save the marked text to a separate file using IX (Index Extraction) or TX (Table of Contents Extraction). From this you can print just the Index or Table of Contents.

  b. Print the file with the Index or Table of Contents included at the end of the printout. (Use TYPE.)

Let's examine each of these steps in more detail.

Table of Contents & Index Procedure



Terminology. We use several terms with meanings specific to this section:

- Source File — This is the main document — the document from which you are extracting a Table of Contents or Index.
- Target File — This is the file to which you save the Index (using IX) or Table of Contents (using TX).
- Marker — Any one of the text markers X1 through X9. You mark text in the source document for inclusion into a Table of Contents or Index.
- Marked Text — The text which you mark for inclusion in a Table of Contents or Index. You choose one of the markers (say, X3) and then mark each entry with it.
- Extract — To copy the marked text from a source file into a target file. The text is sorted either alphabetically (using IX) or by page number (using TX).

ACTION **Generating a Table of Contents or Index.**
This procedure is composed of three parts. For more details on any command, refer to the latter part of this section.

PART I **Mark the Text**
To mark words or phrases for entry into your Table of Contents or Index:

1. Call your document to the display (the source document, that is). Let's call it CHAPTER.

   Type: F5 call chapter ↵

2. You can mark text in three different ways:

   a. **Single Word.** For each word you want included in the Table of Contents or Index:

      Move the cursor to the point immediately following the word.

      Type: F5 x3 ↵

      Press: F3

      Result: The X3 triangle is inserted in the text, marking the word ahead of it. There must be no space (or tab) between the X3 marker and the word.

      (Note: Go to Expanded Display and make sure there are no mode commands — such as «MDNM» — between the text and the X3 marker. If there are, move the X3 marker next to the text.

   b. **Entire Line.** You can mark any *phrase ending with a carriage return* — such as a title on its own line. The whole phrase, from marker to carriage return (up to 200 characters including embedded commands), is picked up. The cursor must either be at the start of a line or have a space (or tab) ahead of it:

      Move the cursor *ahead* of a phrase ending with a carriage return.

      Type: F5 x3 ↵

      Press: F3

   c. **Any Phrase.** You can also enter a phrase that doesn't appear explicitly in the text (or is in the middle of a line). This is especially useful for alternate entries in an index, where "red wagon" appears in the text, but you also want to include "wagon, red." You must type in the "wagon, red" yourself, as follows:

      Type: F5 x3 wagon, red ↵

3. Repeat step 2 until you mark all text you want included in the Table of Contents or Index.

PART II **Specify the Format.**

At the end of the source document you must provide a format such as the one in the following illustration.

1. Move the cursor to the end of your source document.

   Type: [Ctrl] [End]

   We move the cursor to the *end* because only the marked text *above* this point will be included in the Table of Contents or Index.

2. Make sure the cursor is at the beginning of the line. Enter a command T1 through T9 for a Table of Contents or I1 through I9 for an Index. In our example, the text was marked with X3 and we want a Table of Contents, so we will use T3.

   Type: [F5] t3 [↵]

3. Enter the two commands SR IX (to place marked text) and SR PN (to place page numbers) in the order you will want them positioned on a line in the Table of Contents. Also include the leadering (LD) between them, if you wish, and any margin (LM, RM) or other format command.

   In our example we want the marked text placed at the left, a leader composed of periods, and the page number on the right. (Be sure to include a space between the LD command and the period.)

   Press: [F5] sr ix [↵]

   Type: [F5] ld . [↵]

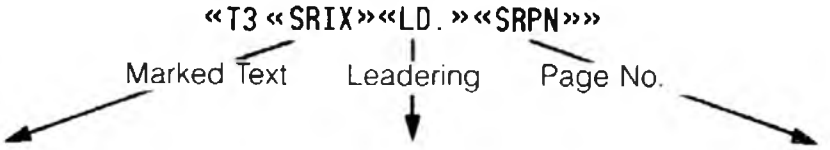
   Type: [F5] sr pn [↵]

   Another SR command, SR CH, gives you the option to set the chapter number and page number together, such as 2-35. See the Set Record command later in this section.

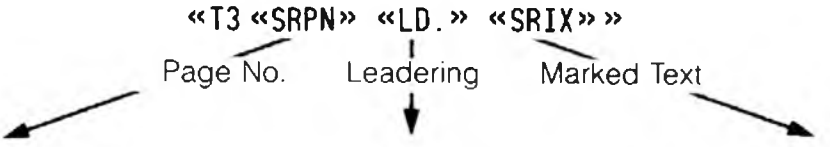
4. Press: [F3] [↵]
5. Type: [F5] store [↵]

EXAMPLES OF SPECIFYING FORMATS (T# and I#)

These first two examples show how you might list a Table of Contents (using T3). Notice in the second figure that the spaces around the LD command creates spaces at either end of the leader.

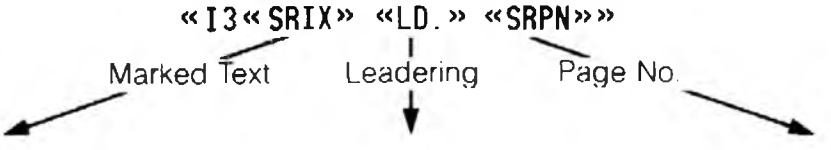



```
Penguin.....5
Baboon.....18
Zebra.....25
Aardvark.....32
Gorilla.....47
```




```
5 ..... Penguin
18 ..... Baboon
25 ..... Zebra
32 ..... Aardvark
47 ..... Gorilla
```

Here is an example of the same marked text listed as an Index (alphabetically, using I3).




```
Aardvark ..... 32
Baboon ..... 18
Gorilla ..... 47
Penguin ..... 5
Zebra ..... 25
```

PART III

Extract the Table of Contents or Index

Let's review what we have done so far to make a Table of Contents or Index. In Part I we used the X3 marker to mark the words and phrases that we wanted included. Then in Part II we provided a format for the Table of Contents, using the T3 command (for an Index we would have used the I3 command).

Now that everything is set up, we have a choice of two ways to output the Table of Contents or Index.

a. **Extract and Save to a Separate File**

If you wish to revise your Table of Contents or Index before printing it, you may want to save it in a separate file.

Enter the appropriate command TX1 to TX9 (Extract Table of Contents) or IX1 or IX9 (Extract Index) — in our example we will use TX3, which extracts only the text marked by X3. To extract text from CHAPTER and save it to a file we call CONTENTS:

Type: F5 tx3 chapter,contents ↵

Result: XyWrite extracts the marked text and page numbers according to the format we specified in Part II. It saves this text to the file named CONTENTS. If you were to omit a name for the target file, XyWrite would place the Table of Contents into a file called TABLE3 (corresponding to marker X3). To view the Table of Contents, type CALL CONTENTS.

b. **Print the Source File and Table of Contents**

To print the entire document including its Table of Contents, simply print the source document (here, named CHAPTER). (Unlike Step a above, this procedure does not save the Table of Contents or Index to a file.)

Type: F5 type chapter ↵

To *suppress* the Table of Contents or Index while printing the source document, enter the NI (No Index) command at the top of your source document.

NOTE #1 **Index Entry Without Page Number.** You can create an index entry that has no page number by using the EX (End X-Marker) command. This is a convenient way to refer your reader to another entry in the index. For example, you might have an entry like "Dog...See Animals" in the index. You might also use the EX command when you want to have an index entry that is only a header under which you list subentries (see Note #2). For details on how to use the EX command, refer to "Suppress Page Numbers" later in this section.

NOTE #2 **Subentry Under Another Entry.** You can place one entry under another by using an IL (Index Label) command. For example, you can place "Dog" under "Animals." You can even have sub-subentries like "Collie" under "Dog." For details on this procedure, refer to "Index Label."

NOTE #3 **Automatic Separators.** The IB (Index Break) command automatically places letters in your index to separate words starting with one letter of the alphabet from words starting with the next. The IB command also lets you control the format of these letter headings or insert other separators. Refer to "Index Break" later in this section for more information.

NOTE #4 **Concise Sorting.** After sorting your index, XyWrite removes duplicate entries (with same word or phrase and page numbering) and combines multiple page numbers on one line, separated by a comma and space. For example, if the entry "Computer" is indexed on pages 7, 16 and twice on 24, its listing will read:

```
Computer.....7, 16, 24
```

NOTE #5 **Customized Sorting.** If you wish to have your index sorted in a different order than that provided by XyWrite default, you can create your own sorting table. Refer to Chapter 6 for details.

FORMAT **CM X#** Text Marker

is any digit between 1 and 9.
This is an embedded command.

PURPOSE Use Text Marker commands **X1** through **X9** to mark text for a Table of Contents or Index. There are nine markers to allow you to create up to nine sets of marked text. You can mark one set of text using X1, a second set with X2, and so on. You might use X1 for a Table of Contents, X2 for an Index, and X3 for a List of Figures.

You have the option of sorting the marked text in *alphabetical* order as an Index or in *page order* as a Table of Contents.

You can mark any amount of text as a single entry in a Table of Contents or Index. You do this by typing in the text along with the X command. If you enter the X command with no text, it marks the single word to its immediate left or up to a carriage return to its immediate right.

ACTION **Marking Text**
To mark text, follow the procedure given earlier in "Part I: Mark the Text."

NOTE #1 **Two Lists From One Marker.** You can generate a Table of Contents *and* an Index from the same set of marked text. That is, from X3 you could use both I3 and T3.

NOTE #2 **Temporary Files.** When you TYPE, TYPEF or TYPES a file, separate temporary files are created to accumulate the text for building an Index or Table of Contents.

EX Suppress Page Number

FORMAT **CM EX**

EX is an embedded command.

PURPOSE The **EX** (End X-Marker) command allows you to create an index entry that has no page number. You can use such entries as titles under which you list other entries (see "Index Label") or as cross-references to other entries.

ACTION **Create Cross-Reference**

When you want to refer your reader to a different entry in your index:

1. Enter a Text Marker command followed by the index entry and the cross-reference information:

   Type: F5 x2 ↵

   Type: `Folios, See Page Numbers`

2. Enter the EX command to suppress the page number.

   Type: F5 ex ↵

3. Close the window.

   Type: F3

   Result: When you print your index, the entry "Folios, See Page Numbers" will appear without a page number.

FORMAT CM IL Index Label

IL is an embedded command.

PURPOSE The **IL** (Index Label) command allows you to have an index entry appear as a subentry under another entry. The text that appears within the IL command is not printed; it is used only for sorting.

Include the IL command as part of the X command when you are marking text.

ACTION **Creating Subentries.**
To create an index that contains the entry "Animals" with a subentry of "cat" and a sub-subentry of "Siamese":

1. Move the cursor to the start of the document and enter the marker for the major entry "Animals."

   Type: F5 x2 ↵

   Type: Animals

2. To suppress the page number for this entry, enter the EX (End X-Marker) command before you close the window.

   Type: F5 ex ↵

   Press: F3

   Result: When you process the index, the entry "Animals" will appear without page numbers.

3. Move the cursor to the point in the text that contains the first subentry and enter the marker for it.

   Type: F5 x2 ↵

4. Enter the Index Level command along with the text of the entry under which you want this subentry to appear:

   Type: F5 il Animals ↵

5. Type a tab and then the text of the subentry:

   Type: [Tab]Cat

6. Close the screen by pressing [F3].

7. Move the cursor to the point in the text that contains the next subentry or sub-subentry. Let's assume that this time we want to mark the sub-subentry "Siamese."

   Type: [F5]x2[↵]

   Type: [F5]il Animals[Tab]Cat[↵]

   Type: [Tab][Tab]Siamese[↵]

   Press: [F3]

8. Repeat this procedure until all entries and subentries have been marked.

Result: When your index is processed, the following entries will appear:

```
Animals
   Cat, 7, 11, 15
        Siamese, 11
```

FORMAT

CM NI No Index

NI is an embedded command.

PURPOSE

The command **NI** (No Index) prevents the printing of an Index or Table of Contents when you use TYPE or TYPEF. It has no effect on Index Extraction (IX) or Table of Contents Extraction (TX) to a separate file.

You use this command when you want to print your document *without* the Table of Contents or Index also being printed. The NI command overrides any Table of Contents command (T1 - T9) or Index Command (I1 - I9).

ACTION

Using the NI Command

To suppress a Table of Contents or Index when printing the source document (using TYPE or TYPEF), enter the NI command as follows. At the top of your source document:

Type: F5 ni ↵

NOTE

Embedded Command. NI is an embedded command. It is embedded in text, visible as a triangle in Normal Display. In Expanded Display it is shown expanded as «NI».

T1 thru T9 Table of Contents Command

FORMAT **CM T#** Table of Contents Command

is any digit between 1 and 9.
T1 through T9 are embedded commands.

PURPOSE Each Table of Contents command **T1** through **T9** is an essential part of generating a Table of Contents. The T1 command produces the Table of Contents for text marked by X1, T2 for text marked X2, and so on up to T9.

Let's use T3 here as an example. When you enter a T3 command in your document, the Table of Contents is printed *along with* the document when you use TYPE (or TYPEF). You use the T3 command to do the following:

- To specify the format. You type in the format of your Table of Contents as part of the T3 command. For example, you can define your format as being marked text on left, leadering consisting of periods, and page number on the right. This would require using SR IX, LD and SR PN, as follows:

  ```
  «T3«SRIX»«LD.»«SRPN»»
  ```

- To place the Table of Contents into your document. When you print your document (using TYPE or TYPEF):

  a. The marked text is accumulated from the start of the document down to the location of T3 (and no further). Thus, the *position* of the T3 command in your document is important — normally you place it at the *end* of your document.

  b. This accumulated text is kept in its page-order sequence (proper for a Table of Contents).

  c. On printout, the Table of Contents is printed at the point that the T3 command is located in the document. (Unlike TX3, no separate file is created.)

ACTION **Formatting a Table of Contents**
To format a Table of Contents using T1 through T9, follow the procedure given earlier in this section under "Part II: Specify the Format."

FORMAT CM I# Index Command

is any digit between 1 and 9.
I1 through I9 are embedded commands.

PURPOSE Each Index command **I1** through **I9** is an essential part of generating an Index. The I1 command produces the Index for text marked by X1, I2 for text marked X2, and so on.

Let's use I3 here as an example. When you enter an I3 command in your document and print it using TYPE, the Index is printed *along with* the document.

The Index commands I1 through I9 parallel the Table of Contents commands T1 through T9 except that the marked text is sorted alphabetically rather than by page number. You use the I3 command to do the following:

- To specify the format. You type in the format of your Index as part of the I3 command. You do this the same as you do for the Table of Contents. For example, you might specify marked text on the left, followed by a comma, a spaceband, and the page number:

  ```
  «I3«SRIX», «SRPN»»
  ```

- To place the Index into your document. When you print your document (using TYPE or TYPEF):

  a. The marked text is accumulated from the start of the document down to the location of I3 (and no further). Thus, the *position* of the I3 command in your document is important — normally you place it at the *end* of your document.

  b. This accumulated text is sorted *alphabetically* (proper for an Index).

  c. On printout, the Index is printed at the point that the I3 command is located in the document. (Unlike IX3, no separate file is created.)

ACTION **Formatting an Index.**
To format an Index using I1 through I9, follow the procedure given earlier in this section under "Part II: Specify the Format."

NOTE **Changing the Sorting Order.** If you wish to have your index sorted in a different order than that provided by XyWrite default, you can create your own sorting table. Refer to Chapter 6 for details.

FORMAT

CM **SR IX**	Place Marked Text
CM **SR PN**	Place Page Number
CM **SR CH***n*	Place Chapter Number

n is the separator between chapter and page values. These are embedded commands.

PURPOSE

The **SR** (Set Record) commands allow you to position the page number (using SR PN or SR CH) and the marked text (using SR IX) in a Table of Contents or Index. You enter these three commands as part of the commands T1 to T9 and I1 to I9.

ACTION

Using the SR CH Command

We illustrate use of SR IX and SR PN in the procedure "Specify the Format" earlier in this section. Let's now take a look at the SR CH command, which produces page references in chapter number-page number format.

1. Be sure that you have used Counter 0 (C0) as the counter for chapter numbers.
2. Mark the words and phrases that you want included in your Table of Contents or Index.
3. Move the cursor to the end of the document and enter the appropriate Index or Table of Contents command. We'll use I3 as an example:

   Type: F5 i3 ↵
4. Enter the command to place the marked text on the left, followed by a spaceband to separate the entry and the page references, and then the command to place the page references in chapter-page format.

   Type: F5 sr ix ↵

   Type: Space Bar

   Type: F5 sr ch- ↵ *(Note the hyphen)*

   Type: F3

   Result: Your index entries will appear in the following format: "museums 3-5."

NOTE

Chapter-Page Separator. You can define any character as the chapter-page separator by entering it immediately after the SR CH command.

FORMAT **CM LD** *n* Leadering

n is any character you specify.
LD is an embedded command.

PURPOSE The **LD** (Leadering) command provides a quick means of inserting a row of characters on a line. Typically you would use LD in a Table of Contents or Index to generate a row of periods or dashes between the name on the left and the page number on the right.

The LD command does two things:

- It pushes all text which follows it (on that line) flush against the right margin.
- It repeats the character given with the LD command (*n*) across the line.

ACTION **Inserting a Leader**
The procedure for inserting a leader is given earlier in "Part II: Specify the Format."

NOTE **Spaces as a Leader.** If you enter LD with no character, it will produce text flush left and flush right on the same line with spaces as the leadering character.

Index Break IB

FORMAT **CM IB** *n* Index Break

n (optional) defines the separator.
IB is an embedded command.

PURPOSE The **IB** command inserts a separator between words starting with one letter of the alphabet and words starting with the next. This separator can be one or more blank lines or a heading that you specify with the IB command *(n)*.

The IB command also lets you control the format of the separator. You can specify the amount of space above and below a heading, its mode (bold, underline, etc.), and how it is placed on the line (e.g., flush left, centered, flush right). These formatting instructions affect only the separators, or headings, not the text of the index.

A special variation of the IB command automatically places capital letters into your index. To use this function, you simply define the separator to be a pound sign (#); XyWrite interprets the # to mean "insert capital letters as index separators." If there are no entries for a particular letter, XyWrite skips the heading for that letter. See the description below for details on how to use this variation of the command.

ACTION **Specifying Letters as Separators.**
If you want your index to contain capital letters as separators and to have them be bold, centered and separated from the entries above and below by one blank line:

1. Go to the top line of the text file.
2. Type: F5 ib ↵
3. Enter the formatting commands.

   Press: ↵

   Type: F5 fc ↵

   Type: F5 md bo ↵

4. Enter the pound sign (#) to activate the letter separators. Follow it with two carriage returns.

   Type: #↵↵

   Press: F3

Result: When you process your index, it will contain bold, centered capital letters as headings. They will be separated from the index entries by one carriage return.

Table of Contents Extraction TX1 thru TX9

FORMAT **CM TX#** *sourcefile,targetfile* Table of Contents Extraction

is any digit between 1 and 9.
sourcefile is the file containing marked items.
targetfile is where the Table of Contents is saved.
TX1 through TX9 are immediate commands.

PURPOSE Each Table of Contents Extraction command **TX1** through **TX9** allows you to extract a Table of Contents from your document and save it to a file separate from the original file. It extracts text from the *sourcefile* and saves it to a *targetfile*. When you use the command TX2, for instance, the source file accumulates text marked with marker X2 and formats the text as specified by the T2 command.

If you omit the *sourcefile*, XyWrite will extract a Table of Contents from the document currently displayed. If you omit the *targetfile*, XyWrite will save the Table of Contents to a file it names TABLE1 (for X1) or TABLE2 (for X2), and so on.

ACTION **Extracting Tables of Contents**
To extract a Table of Contents from a source file, follow the procedure described earlier in this section under "Extract the Table of Contents or Index."

NOTE #1 **Index of a List of Filenames.** To create a single Table of Contents across several files, place a command T1 through T9 at the end of the last file, and use:

CM TX# *@parentfile,targetfile*

Refer to TYPE @ for more information.

NOTE #2 **NI Command.** The command NI (No Index) does *not* inhibit execution of this command.

NOTE #3 **Requirement.** Each TX command extracts text according to the format established by the corresponding Table of Contents command. For instance, if T1 is not present in your source document (or if it is at the top), TX1 will extract nothing.

NOTE #4

Unnumbered Text. There may be times when you want to produce documents that contain no chapter or section numbers in the text, but do contain the numbers in the Table of Contents.

When you are preparing such documents, use the LV0 through LV9 commands *instead of* the C0 through C9 commands. The LV commands work like the C commands except that the LV commands do not output numbers when you print.

When you have completed your document and are ready to create the Table of Contents, use the TX command to save the Table of Contents to a separate file. TX converts the LV commands to C commands. You can then enter the appropriate DC commands to define the counter values so the printed Table of Contents will contain section numbers. For more information on the C and DC commands, refer to "Numbering" in Chapter 4.

Index Extraction IX1 thru IX9

FORMAT **CM IX#** *sourcefile,targetfile* Index Extraction

is any digit between 1 and 9.
sourcefile is the file containing marked items.
targetfile is where the Index is saved.
IX1 through IX9 are immediate commands.

PURPOSE Each Index command **IX1** through **IX9** allows you to extract an Index from your document and save it to a file separate from the original file. It extracts text from the *sourcefile* and saves it to a *targetfile*. When you use the command IX2, for instance, the source file accumulates text marked with marker X2 and formats it according to the I2 command. The target file is created especially to hold this Index.

If you omit the *sourcefile*, XyWrite will extract an Index from the document currently displayed. If you omit the *targetfile*, XyWrite will save the Index to a file it names INDEX1 (for X1) or INDEX2 (for X2), and so on.

ACTION **Extracting An Index.**
To extract an Index from a source file, follow the procedure described earlier in this section under "Extract the Table of Contents or Index."

NOTE #1 **Index of a List of Filenames.** To create a single Index across several files, place a command I1 through I9 at the end of the last file, and use:

CM IX# *@parentfile,targetfile*

Refer to TYPE @ for more information.

NOTE #2 **NI Command.** The command NI (No Index) does *not* inhibit execution of this command.

NOTE #3 **Requirement.** Each IX command extracts text according to the format established by the corresponding Index command. For instance, if I1 is not present in your source document (or if it is at the top), IX1 will extract nothing.

User Programming

INTRO

Above and beyond all of the functions described earlier in this manual, XyWrite provides the power of User Programming. This feature allows you to combine any commands, in any order, in a program for automatic execution.

This section starts with a description of the general procedures and then covers the commands NEP, CAP, P, RUN, and LDPM.

CONTENTS

Programming Procedures

PURPOSE

What is Programming? Programming is a powerful technique which enables you to record any sequence of keystrokes for later execution. You can write your own programs to perform complicated operations, load them to Save/Gets, and then run them with as few as two keystrokes, e.g., Alt X. You can perform any operation which can be planned out in advance. What makes programming so powerful is: **Any keystrokes you can type manually from the keyboard within XyWrite, you can also record in a program file and run automatically.** There are basically two parts to programming:

1. You record a sequence of keystrokes in a *program file*.

2. You then *run* this file. This automatically executes the stored keystrokes one after another, as if you had typed them from the keyboard.

You can record and execute virtually any combination of commands in this manual. Once you grasp the concept, you can be quite innovative devising your own programs.

Practical Uses of Programming. Here are a few examples of simple programs. In each example, the program is assigned to a Save/Get key.

Saving a File. You can write a program which performs the SAVE command. If you assign the program to Save/Get key S, you can then save a file simply by pressing Alt S. This program file would contain the keystrokes:

F5 save F9

- **Displaying a Partial Directory** — Display in the directory only the files which start with a particular letter — for instance, the letter "D":

  F5 dir d*.* F9

  You might assign it to Alt D. Similar programs could display files containing other characters.

- **Calling a File by Pointing** — Call a directory, move the cursor to the filename you want, and then run this program with one keystroke to call the file:

  F5 call F9

  This program might be assigned to Alt C.

- **Making a Word Bold** — Place the cursor on the first letter of the word and strike Alt B to make it bold. The program is:

  F3 F1 F5 search /S/ F9 ← F1 Ctrl 2 F3 F5 F10

  You enter the character S by pressing Alt Shift S. (The ← key shown is the Cursor Left key.)

- **Run Another Program** — Suspend XyWrite and start another program. For example, you could start WordProof (which is abbreviated WP) and assign the program to Alt W:

  F5 do b:wp F9

You can find the program files for these applications in the section called "Programming Examples."

ACTION

Creating and Running a Program File

This procedure is divided into six parts. As an example, we will write a program to save a file with the keystrokes Alt S.

1. **Create the Program File.**

   Create a new program file using NEP along with a filename — for example, SAVEFILE:

   Type: F5 `nep savefile` ↵

   Result: A new (empty) file is created with the name SAVEFILE (this name appears at the top of the screen).

2. **Write the Program.** For more details, see "Writing A Program" which follows.

   Type: Scroll Lock F5 `save` F9 Scroll Lock

   Result: The display should show: **BC** `save` **XC**

   Analysis:

   The first Scroll Lock turns on the record mode (the "S" appears at the top right of the screen).

   F5 displays as **BC** (begin command).

   SAVE is the command.

   F9 displays as **XC** (execute).

   The last Scroll Lock turns off the record mode (the "S" disappears).

3. **Store the Program**

   Type: [F5]store[↵]

   Result: The program file is stored on the disk and disappears from the screen.

4. **Run the Program.** Let's test our program on the text file HELP1.

   Type: [F5]call help1[↵]

   Result: The file is displayed on the screen. (We will test the program by saving this file.)

   Type: [F5]run savefile[↵]

   Result: The file is saved to disk by the program and the prompt line says DONE.

5. **Load the Program onto a Save/Get Key.** *(Optional)*

   Type: [F5]ldpm savefile,s[↵]

   Result: The program is saved to key [Alt]S. The prompt line says DONE.

6. **Run the Program from a Save/Get Key.** *(Optional)*

   Type: [F5]call help1[↵]

   Press: [Alt]S

   Result: The HELP1 file is saved to disk, and the prompt line says DONE, exactly the same as in step 4. This shows you can run a program with only two keystrokes.

NOTE #1

Record Mode. Use the [Scroll Lock] key to record commands in a program file. [Scroll Lock] switches the record mode *on* and *off*. An "S" appears at the top right of the screen when the record mode is *on*:

[Scroll Lock] *on* — Record mode *on*
Records all keystrokes as key codes in the program (for later execution). For example, [F5] is recorded as **BC** (begin command).

[Scroll Lock] *off* — Record mode *off*
Keystrokes function normally and are not recorded in the program. For example, [F5] moves the cursor to the command line, and clears the command line.

NOTE #2

Additional Save/Get Keys. In Step 5 of the previous procedure you are not limited to the 26 letter keys A-Z and 10 numbers keys 0-9. You can also use any keys to which you assign Function Calls &A to &Z and &0 to &9 in the Keyboard File. These additional Save/Get keys are reserved for programming.

ACTION

Writing a Program.
The procedure for writing a program is as follows.

1. **Create or Call a File.** Use NEP or CAP to open the program file.

2. **Begin the Record Mode.** Turn on the record mode. Press [Scroll Lock] so that the S is displayed at the top right of the screen.

3. **Record the Keystrokes.** Type the exact keystrokes for the procedure you want. This procedure constitutes the program.

   The key codes that are displayed represent the keystrokes; for example, **BC** means Blank the Command Line, and corresponds to function key [F5]. The complete list of key codes is located in Chapter 6 under "Function Calls."

4. **Correcting Mistakes.** If you make a mistake, press [Scroll Lock] to turn the record mode off. Then correct the mistake. The entire keyboard functions now work normally — you may use the Backspace key or move the cursor around.

   After correcting the mistake, press [Scroll Lock] again to continue recording keystrokes.

   To Improve Readability. For commands executed from the command line, your program will be more readable if you use a carriage return [↵] as a replacement for [F9]. To do this, follow the command with [Scroll Lock] [↵] [Scroll Lock] rather than [F9]. The file STARTUP.INT is written this way; look at it to see how much easier it is to read such a program file.

   Finish the Record Mode. When done typing the program:

   Turn off the record mode by pressing [Scroll Lock] so that the **S** is no longer displayed.

5. **Store the Program.** When the program is as you want it:

   Turn record mode off as shown in the previous step. Store the program on disk by typing:

   [F5]store[↵]

Program File. A program file is a file in which you store commands (and text) for later execution. Each keystroke you type is stored as an individual key code. All key codes are listed under Function Calls in the Keyboard File section of Chapter 6.

Program files require that you use NEP or CAP to create or edit them (rather than NEW or CALL).

You can identify a program file by the presence of key codes (as shown in the following example).

The program files for the programs listed earlier under "Practical Uses" are shown on the following page.

Saving a File. You type [Alt]S to save the current file (rather than [F5]SAVE[↵]):

Keystrokes: [F5]save[F9]
As displayed: **BC save XC**

1. Create the Program File (filename: SAVEFILE)

2. Load the program onto Save/Get key: LDPM SAVEFILE,S

3. Run the program: To save the file, place the cursor anywhere within the document you want to save, then press [Alt]S.

Displaying a Partial Directory. You press [Alt]D to display only the files in the directory which start with the letter "D" (those of current interest).

Keystrokes: [F5]dir d*.*[F9]
As displayed: **BC dir d*.*XC**

1. Create the Program File (filename: PARTIAL)

2. Load program onto Save/Get key: LDPM PARTIAL,D

3. Run the program: To display the partial directory, move cursor onto the command line with the current window empty, then press [Alt]D.

Calling a File by Pointing. You call a directory, move the cursor to the filename you want, and then press [Alt]C to call the file.

Keystrokes: [F5]call[F9]
As displayed: **BC callXC**

1. Create the Program File (filename: CALLFILE)

2. Load program onto Save/Get key: LDPM CALLFILE,C

3. Run the program: To call a file, first list the directory, move the cursor to the file you want, then press [Alt]C.

NEP Creating a New Program File

FORMAT **NEP** *newfile* New Program File

newfile is the name of the new program file.
NEP is an immediate command.

PURPOSE **NEP** (New Program file) creates a new *program* file with the name you specify. To see how NEP fits into the overall programming procedures, refer to "Programming Procedures" earlier in this section.

ACTION **Creating a New Program File.**
To create a new (empty) program file:

1. Decide on a name for your new program file — for example, BRANDNEW.PM. You may choose any filename extension you want, but we recommend you use .PM to distinguish it as a program file.

2. Type: F5 nep brandnew.pm ↵

Result: This command creates a new program file named BRANDNEW.PM. (The file will not be created if one with the same name already exists.)

NOTE #1 **Saving the File.** NEP creates a new program file the same as NEW creates a new text file. Both create a file in memory; you must SAVE or STORE the document before it is safely stored on disk.

NOTE #2 **Option.** You can create a new program file and copy an existing file into it. For example:

F5 nep brandnew.pm,oldfile.pm ↵

Refer to the NEW command in the Filing chapter for details.

Calling a Program File — CAP

FORMAT

CM **CAP** *programfile*

programfile is the name of the program file you want to display.
CAP is an immediate command.

PURPOSE

CAP (Call Program File) loads a copy of the specified *program* file from the disk to the display for you to view and revise.

ACTION

Calling a Program File.
To call an existing program file to the display:

1. Start with the display cleared of any document. If you need to clear the display, use STORE or ABORT.

2. Enter CAP along with the filename you want to call. Let's say the name of the program file you want to call is EXERCISE.PM:

   Type: F5 `cap exercise.pm` ↵

Result: This example calls the program file named EXERCISE.PM to the display.

NOTE

CALL vs. CAP. CAP loads a program file to memory from the disk the same way that CALL loads an ordinary file.

To see how CAP fits into the overall programming procedure, refer to the earlier section "Programming Examples."

P Pausing During Execution

FORMAT **CM P** *comment* Pause

comment (optional) is any message you write, and is displayed on the command line during the pause.
P is an immediate command

PURPOSE When entered in a program file, **P** (pause) causes the program execution to pause for about one second (6/10 second on the IBM PC/AT personal computer). For longer pauses, execute the pause multiple times with F9.

You may find the pause useful for slowing down the program sequence, or for viewing intermediate results before they flash by. In addition, you may add your own message after the P, to be viewed on the command line.

ACTION **Pausing During Executiom.**
As an exercise, let's add a pause and a comment to the beginning of the XyWrite startup file STARTUP.INT (which is a program file).

1. Use CAP to call the program file.

   Type: F5 cap startup.int ↵

2. Add the Pause Command at the first line.

   Press: Scroll Lock

   Type: F5 p Startup File F9 F9 F9

   Press: Scroll Lock

   Result: The first line of the display should show:

   BC p Startup FileXC XC XC

   Type: F5 store ↵

3. Store the program.

   Type: [F5] store [↵]

4. Run the program. To observe the pause:

   Type: [F5] run startup.int [↵]

Result: Notice that the sequence pauses at the beginning and shows "p Startup File" on the command line.

Analysis. The keystrokes in Step 2 do the following:

The first [Scroll Lock] turns on the record mode (the "S" appears at the top right of the display).

[F5] displays as **BC** (begin command).

"p" is the Pause command. "Startup File" is the comment to appear on the command line.

[F9] displays as **XC**. The three [F9] commands cause a three-second pause.

The last [Scroll Lock] turns off the record mode (the "S" disappears).

RUN — Running a Program File

FORMAT

CM **RUN** *programfile,n*

programfile (optional) is the name of the program file you want to run.
n (optional) is any number you specify (up to 28 digits), to be passed into the program. This number is assigned to the command AS (Argument Insert). (See Advanced User Programming.)
RUN is an immediate command.

PURPOSE

RUN causes the specified program file to execute. This means the commands (and text) stored in the program file are executed automatically, as if typed from the keyboard. To see how RUN fits into the overall programming procedure, refer to the earlier section "Programming Procedures."

ACTION

Running a Program File
To run a program file — for example, EXERCISE.PM:

Type: F5 run exercise.pm ↵

Result: This command runs the program file named EXERCISE.PM — the keystrokes stored in that file are automatically executed.

To stop the program (if necessary):

Press: Ctrl Break

ALSO SEE

Related Commands. You can also run a program file with an Alt key. This requires first using LDPM (load program) to load the program file onto a Save/Get key. See LDPM.

NOTE

Shortcut. XyWrite remembers the name of the last file run. Thus, if you enter the RUN command without a *programfile*, XyWrite re-runs the most recently run file.

FORMAT

CM **LDPM** *programfile,#* *(Option 1)*
CM **LDPM** *programfile* *(Option 2)*

programfile is the existing program file you want to load.

 # (optional) is the single letter (A-Z) or number (0-9) or the two characters (&A-&Z or &0-&9) where you want to save the program file. You then run the file using Alt#.

LDPM is an immediate command.

PURPOSE

LDPM (Load Program) loads a program file either onto the specified Save/Get Key *(Option 1)* or into memory *(Option 2)*. Option 1 allows you to run the program file with an Alt key (rather than with the RUN command). Option 2 allows you to RUN a program directly from memory rather than from disk.

To see how LDPM fits into the overall programming procedure, refer to the earlier section "Programming Procedures."

ACTION
(Option 1)

Loading a Program onto a Save/Get Key.
To load a program onto a Save/Get Key:

1. **Load the program.** To load the program file EXERCISE.PM onto the AltX key:

   Type: F5 `ldpm exercise.pm,x` ↵

   Result. The program file is copied to the Save/Get key (in memory). You can now run the program file EXERCISE.PM by pressing AltX.

2. **Verify.** (Optional) To verify that the file has indeed been loaded onto that key:

   Press: Ctrl F2 X

After viewing the text, press Space Bar to return to the document.

3. **Store the Save/Get key.** (Optional) If you wish to keep this program file loaded on the Save/Get key for use at future editing sessions (after you QUIT), use STSGT (Store Save/Get Keys). Refer to STSGT in Chapter 3.

NOTE #1

Additional Save/Get Keys. You can load programs onto any of up to 72 keys. This includes the 36 standard Save/Get keys Alt A through Alt Z (Function Calls @A-@Z) and Alt 0 through Alt 9 (Function Calls @0-@9). It also includes up to 36 keys assigned to Function Calls &A through &Z and &0 through &9.

ACTION
(Option 2)

Loading a Program into Memory.
To load a program directly into memory (rather than onto a Save/Get Key), use LDPM with only the filename of the program. For example:

Type: F5 `ldpm exercise.pm` ↵

Result: The program file EXERCISE.PM is now stored in memory. When you use the RUN command, XyWrite checks to see if the program you specify is stored in memory before going to the disk.

NOTE #2

Removing a Program. To delete a program that is stored on a Save/Get key, use the REMOVE command as described in Chapter 3. To delete a program that is stored in memory, type REMOVE *filename*.

NOTES

Advanced User Programming

INTRO

User Programming, as explained in the previous section, allows you to record any sequence of keystrokes and create a program for later execution. *Advanced* User Programming is an extension of User Programming and provides a set of commands that you can use to add more power and functionality to your programs.

Refer to Appendix D for more programming examples.

CONTENTS

NOTE

This section contains advanced programming material and is not recommended for beginners. You need some programming background and need to be familiar with the features explained in the previous section, "User Programming" to get the full benefits of the advanced programming features.

Advanced User Programming

PURPOSE

Advanced User Programming extends the power and functionality of XyWrite. You can create new functions based on combinations of existing functions to bring convenience and speed to your daily work.

Here are elementary, but useful, applications for advanced programming. These examples are described in detail later in this section.

- **Character Count.** Indicates the number of characters to the end of a file.
- **Approximate Word Count.** Performs a character count as above, then divides by 7 (arbitrary) to estimate the word count.
- **Select a Printer File.** Pauses during the STARTUP.INT routine to ask which Printer File you want to load.

You can also create procedures that accept keyboard input, such as an order entry system. In addition, advanced user programs can help you to accomplish complex editing tasks which conditionally require a change to be made. For example, you can create a program that removes list entries that are dated previous to a certain date.

Features. Using the Advanced User Programming commands, you can do the following:

- Save values to 1000 Save/Get keys (000-999) during program execution
- Branch on a condition (IF Statement)
- Stop and accept keyboard input anytime during program execution (RC - Read Character)
- Pass in a value from outside the program (AS - Argument Insert)
- Determine the current cursor position (CP - Cursor Position)
- Read the current column position of the cursor (CL - Current Location)
- Jump to a label (GL - Go to Label)
- Save strings (SV), subroutines (SU), or expressions (SX) in a Save/Get
- Insert the contents of a Save/Get key at the cursor position (PV - Put Value)
- Make use of the current filename, path, page number, line number, and other XyWrite settings (VA - Value of Variable)
- Exit a subroutine (EX) or exit the program (EX1)

ACTION **Writing an Advanced User Program.**

The procedure for writing an advanced user program is as follows:

1. Create a program file using NEP (rather than NEW). We use the extension .PM merely as a reminder that this file is a program.

   Type: [F5]nep exercise.pm[↵]

2. Turn on record mode: press [Scroll Lock]. An [S] displays at the top right corner of the screen.

3. Record the keystrokes that represent the actions you want to occur when the program begins execution. Function calls on the screen represent the keystrokes.

4. When you reach a point in the program where you want to insert an *embedded* command, turn off record mode by pressing [Scroll Lock]. The [S] no longer appears in the corner.

   Type in the desired embedded commands (refer to the section "Commands and Operators" later in this section). Each time, press [F5] to move to the Command Line, enter the command and press [F9]. If the command requires an argument, a text window opens to allow you to enter it.

5. Complete the program by repeating Steps 2 through 4.

6. Be sure you are out of record mode (so the [S] no longer appears) and store the program using STORE.

ACTION **Running an Advanced User Program.**

You run advanced user programs with the RUN command. For example, to run EXERCISE.PM:

Type: [F5]run exercise.pm[↵]

Refer to the earlier section called "User Programming" for information on how to run programs from Save/Get keys.

ACTION

Example of an Advanced User Program.
We call this program Character Count. It counts the number of characters in a file and displays the result.

1. Create a program file:

   Type: [F5]nep count[F9]

2. Press [Scroll Lock] to turn record mode on.

3. Begin entering the program. Press [Ctrl][End]. A bold **BF** appears on the screen (for Bottom of File).

4. Press [Scroll Lock] to turn record mode off.

5. Type: [F5]sx[F9] A text entry screen opens.

6. In the screen, enter Save/Get 01 and then the cursor position command:

   Type: 01,[F5]cp[F9]

7. Press [F3] to close the window. The SX command displays in expanded mode as:

   «SX01,«CP»»

8. Press [Scroll Lock] to turn record mode on again.

9. Press [F5]. **BC** appears in the program file. This will cause the cursor to move to the Command Line.

10. Press [Scroll Lock] to turn record mode off again.

11. Type [F5]pv 01[F9]. This will put the value on the Command Line when the program runs. The program file looks like:

    BF▲BC▲

12. Press [F5] store [F9] to store the program file.

This program moves the cursor to the end of the file, counts the number of characters in the file, and prints the number on the Command Line.

To execute the program, call up any file:

Type: [F5] run count [F9]

ACTION **Revising Programs.**

To revise an advanced user program, you need to "expand" the commands using the Expanded Display:

1. Call up the program file you want to revise using CAP. For Example, to revise the file EXERCISE.PM:

   Type: [F5] cap exercise.pm [↵]

2. Press: [Ctrl] [F9] (for Expanded Display)

   Result: The triangles no longer appear — the commands are shown expanded within double angle brackets.

3. Insert and delete any commands you want removed from the program including the surrounding double angle brackets.

4. Press: [Ctrl] [F9] (for Normal Display)

   Result: The commands which are represented as triangles will be expanded on the screen.

ACTION **Using Expanded Display for Editing.**

You can use the Expanded Display for typing or modifying commands. For example, do the following three steps to enter the command:

«SX01,«RC»»

1. Press: [Ctrl] [F9] (to switch to Expanded Display)
2. Move the cursor where you want to insert the command, then type:

   Press: [Ctrl] [<]

   Type: SX01,

   Press: [Ctrl] [<]

   Type: RC

   Press: [Ctrl] [>]

   Press: [Ctrl] [>]

3. Return to Normal Display:

   Press: [Ctrl] [F9]

ACTION **How to Type Nested Commands.**

The SX command and IF statements both allow you to type a command within another command. These are called nested commands. Here are two examples:

«SX01,«RC»»

«IF«IS01»==«IS02»»

When nesting within an SX command or IF statement, do the following:

1. Enter SX or IF:

   Type: [F5] sx [↵]

2. Type the rest of the command, finishing with [F3] (as instructed on the display). For example:

   Type: 01, [F5] rc [↵]

   Press: [F3]

Result: This command would appear in the Expanded Display as:

«SX01,«RC»»

NOTE #1 **Running a Program from STARTUP.INT.** If you place a program into STARTUP.INT to be run on startup (with RUN *filename*), be sure to end that program with «EX», rather than editing it with «EX1» or omitting EX altogether. This will ensure that execution returns to the startup file and continues with the next statement. After all, STARTUP.INT is a program and your program is a subroutine. (If you omit the «EX», execution will stop inside your subroutine and not return to the main STARTUP.INT program.)

NOTE #2 **Accessing Function Calls.** You can use FUNC to insert function calls into your program even if they are not assigned to keys. For example: F5 func nm ↵ (No Markers), which hides all embedded triangles and paragraph-end arrows.

NOTE #3 **Improving Readability.** To make your programs easier to read, type two carriage returns after each GL statement (as we did in Example #2 which follows). This will not affect operation of the program, since the GL command skips all statements and goes directly to the specified label.

PURPOSE

Commands and Operators.
This section lists the commands available to you in Advanced User Programming. The commands in Advanced User Programming are *embedded* commands — they appear as triangles in Normal Display. They are divided into the following categories:

Commands Which Return Values
Save/Get Commands
Flow Control Commands
Relational Operators
Logical Operators
Arithmetic Operators
String Operators

Save/Gets. There are three kinds of Save/Gets:

- **Ordinary Save/Gets** - A-Z, 0-9
  These are the Save/Gets that you normally access with the Alt keys. They are "permanent" in that their contents remains intact when the program stops running. These Save/Gets are described in the "Save/Get Keys" section of Chapter 3.
- **Temporary Save/Gets** - 000-999
  These one-thousand Save/Gets are saved only while the program is running. Once the program stops, their contents vanishes. Note that Save/Gets 000-999 are separate and distinct from Save/Gets 0-9. You can abbreviate Save/Gets 000-009 with 00-09.
- **Additional Program Save/Gets** - &A-&Z, &0-&9
  These can be used only with LDPM and RUN. For example: LDPM *filename*,&A. See the LDPM command in "User Programming" earlier in this chapter for more details. Like ordinary Save/Gets, these are "permanent."

Commands Which Return Values

The following commands insert values into your program from XyWrite, from the user, or from another program. You can use any of these commands as values in expressions.

AS **Argument Insert.** Takes the string passed in from the RUN *filename,string* command, saves it to Save/Get 00, and then puts the string into the text (or expression) by performing the PV00 function. For example, if you start the program with RUN EXERCISE,1234, then AS will be assigned the string 1234 (not the number) within the program. If you load the program to a key with LDPM and then run that program from a Save/Get key, AS is assigned the string on the Command Line. Since AS returns a *string* rather than a number, to use AS within SX, use IS00 instead: «SX01,«IS00»». Note: Save/Get 00 and Save/Get 000 are the same.
Example #1: The new variable is «AS»
Example #2: «IF«AS»==«PV01»»

CL **Column Location of Cursor.** Takes on the value of the current column position of the cursor. The columns on the display are numbered starting from the left at column 0. Example: «SX01,«CL»»

CP **Character Position.** Takes on the value of the current character position in number of characters from the beginning of the file. Example: «SX01,«CP»»

Once you have saved the cursor position and moved the cursor, you can use the JMP command to return to that previous position. (The JMP command is described in Chapter 3 under "Go to Page and Line Number.")

ER **Error.** Takes on a logical value of either TRUE or FALSE. Is TRUE if there was an error in the previous command; otherwise, it has the value FALSE. Frequently used with the SEARCH command — ER is TRUE if the search returns NOT FOUND. Example: «IF«ER»==TRUE»»

RC **Read Character.** Stops the program and waits for the user to press a key. When a key is pressed, RC takes on the value of that key and the program continues with the next step. Use RC in an expression; typically you would assign RC to a Save/Get. Example: «SX01,«RC»»

VA *nm* **Value of Variable.** Obtains the current value (or string) of the variable you request (*nm*). For example, VA LM obtains the current value of the Left Margin. There are over 18 different variables you can request. Unlike the other commands in this section (except IS), you can use this command *either* in a program *or* directly in text. Refer to Chapter 6 under "Default Settings" for details on the VA command. Example: «VALM»

Save/Get Commands

In the following commands, # is a Save/Get: either an ordinary Save/Get (A—Z, 0—9) or a temporary Save/Get (000—999). Save/Gets &A-&Z and &0-&9 do *not* work with these commands. Also refer to the figure "Save/Get Commands" which follows.

GT # **Get Save/Get.** Inserts the text or invokes the program assigned to the specified Save/Get (#).

GT # is equivalent to the @A to @Z and @0 to @9 function calls used by permanent Save/Gets, which are normally assigned to Alt keys. GT inserts text all at once, while PV inserts the text one character at a time. (See keyboard function calls in Chapter 6.)
Example: «GT98»

IS # **Insert Save/Get.** In Advanced User Programming, you can use IS within an IF statement to compare the string contents of one Save/Get to another. (When used outside of IF, IS operates as just another regular formatting command. IS is further described under Save/Get Keys in Chapter 3.)
Example: «IF«IS01»==«IS99»»

PV # **Put Variable.** Inserts the characters one at a time from the specified Save/Get to the current location. May be used in two ways:

- When used outside of an expression, PV inserts the text or runs the program assigned to the Save/Get — either on the Command Line or in text.
  Example: «SX01,«VA$fi»» «PV01»
- When used inside an expression, PV combines with the expression to give a numeric result.
  Example: «IF«PV800»<«PV801»»

SU *#,string* **Save Subroutine.** Saves the string as a *program* to the specified Save/Get (#). (This is similar to SV, except the string is marked as a program.) You can run this subroutine within another program using GT #. The following example saves to Save/Get 98 a subroutine that executes the SAVE command.
Example: «SU98,BC saveXC »

SV *#,stringexp*

Save String. Saves the string expression (*stringexp*) to the specified Save/Get (#). SV can be used only for saving literals. You can compare this string with contents of any other Save/Get. The following example puts the string YES into Save/Get 99.
Example: «SV99,YES»

SX *#,numericexp*

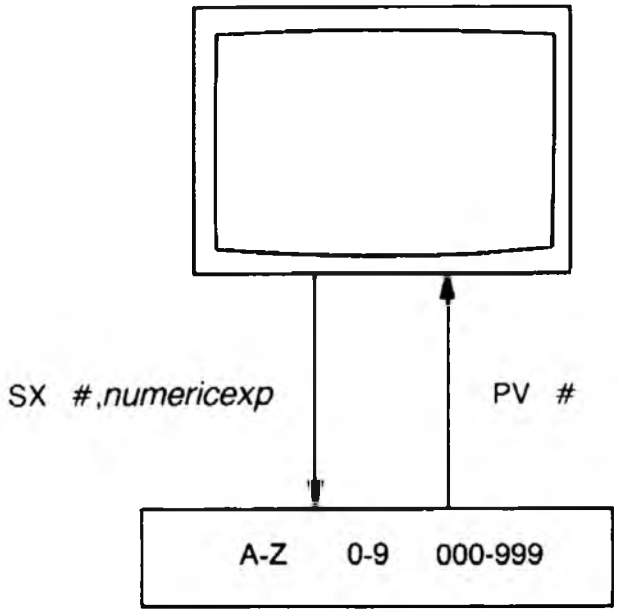
Save Expression. Saves the result of the indicated numeric expression to the specified Save/Get (#).
Examples: «SX01,25», «SXA,«IS99»»

NOTE #4 **Arguments to Commands.** Arguments (values) to embedded commands can include Save/Gets, logical expressions, numeric expressions, or additional embedded commands.

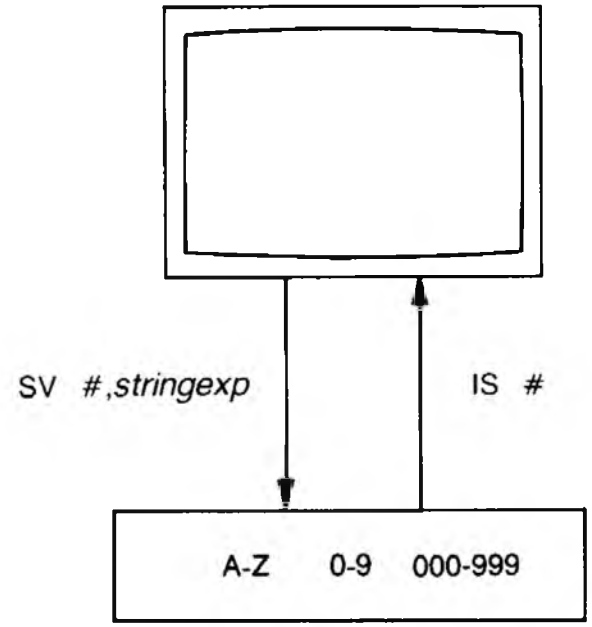
NOTE #5 **Further Reading.** For general background information on programming terminology and syntax, see Chapter 3 of the IBM "BASIC Handbook."

SAVE/GET COMMANDS

NUMERIC EXPRESSIONS



STRING EXPRESSIONS



Flow Control Commands

The following commands control the flow of the program.

EX — **Exit and Continue.** EX and EX1 are two different ways to exit a program. EX is a subroutine return. When used at the main level of a program, the EX command stops the program. When used in a subroutine, EX exits from that subroutine and continues execution at the point the subroutine returns to. (A subroutine is any program within a program — enter a subroutine with RUN or a Save/Get and end it with EX.) Example: «EX»

EX1 — **Exit and Stop.** EX1 stops the program altogether. Unlike EX, it stops regardless of whether execution is in the main program or in a subroutine. Example: «EX1»

GL *label* — **Go to Label.** Causes a jump to the label specified by the LB command. The *label* can be any length string. Example: «GLSTART»

IF *expression truebranch* **EI** *falsebranch*

IF Condition. This command evaluates a boolean expression and determines whether the expression is TRUE or FALSE. If TRUE, execution continues with the next statement. If FALSE, execution jumps to the EI command (End If) and continues from there. Examples:

Comparing Values:
«IF«PV01»==«PV02»»«GLA»«EI»«GLB»

Comparing Strings:
«IF«IS01»==«IS99»«GLA»«EI»«GLB»

LB *label* — **Label.** Marks a spot in the program that you can jump to with the GL command. The *label* can be inserted at any point in a program, and can be any length string. May also be used simply as a comment in a program. (When writing a comment that contains spaces, though, the comment must be typed in the Expanded Display.) Example: «LBSTART»

Relational Operators

These operators let you compare two numeric expressions (with PV) or two string expressions (with IS).

<	Less Than
>	Greater Than
<=	Less Than or Equal (same as =<)
>=	Greater Than or Equal (same as =>)
<>	Not Equal
==	Equal

Example: «IF(«PV01»<«PV02»)»

Logical Operators

Logical operators perform logical (or boolean) operations on numeric or string expressions. You use logical operators within IF statements.

!	Or
&	And
@XOR	Exclusive Or
@NOT	Not
@UPR	Convert letters to uppercase. (Parentheses are required.)
@CNV	Takes a key read by RC and converts it to a function call.
@SIZ	Returns a value equal to the number of characters in a string.

Example: «IF((«IS99»==«IS01»)!(«IS99»==«IS02»))»

Arithmetic Operators

Use these operators to perform arithmetic on numeric values.

+	Addition
-	Subtraction
*	Multiplication
/	Division

Example: «SX99,«PV01»*10+«PV02»»

String Operators

A string operator operates on two string expressions.

+	Concatenation
∈	String exists within a string (ASCII character 238)

Example: «SX99,«IS01»+«IS02»»
«IF(«IS01»∈«IS99»==0)»

Examples of Advanced User Programming

Advanced Programming, like any programming language, has virtually unlimited uses. The way you use it depends on your particular needs. We showed the Character Count example earlier. Following are two more examples of advanced programming. Appendix D has still more examples.

- Word Count — determines the number of words in a document.
- Select a Printer File — lets you choose which Printer File you want to load.

EXAMPLE #1 **Word Count.** You can modify the Character Count program shown earlier in this section to approximate the number of words in your document. You do this by first counting the number of characters and then dividing the result by seven. You would modify Step 6 as follows:

```
«SX01,«CP»/7»
```

EXAMPLE #2 **Select a Printer.** You can write programs that will pause in the middle of execution, ask a question and wait for you to respond. Thus you can stop and make choices. For example, you can modify your STARTUP.INT file to choose which Printer File to load. The following program demonstrates the RC (Read Character) command and the IF statement. RC causes the program to pause, allowing you to select a printer by pressing a letter (D or L).

```
«LBA» «SV01,D» «SV02,L» BC Dot matrix (D) or Letter quality (L)?« GLB»
«LBB» «SX99,«RC» «SX99,@UPR(«IS99»)» «IF(«IS99» ==«IS01»)» «GLDOTMATRIX»
«EI» «IF(«IS99» ==«IS02»)» «GLLETTER»
«EI» BC p Press 'D' or 'L'XC «GLA»
«LBDOTMATRIX» BC ldprn 3epsonfx.prnXC «EX»
«LBLETTER» BC ldprn 3diablo.prnXC «EX»
```

NOTE #6

Testing for a Carriage Return. To test for a carriage return requires two steps: (1) save a carriage return to a Save/Get and (2) make an IF statement. We will save the carriage return to Save/Get 91 and then in the IF statement, we will compare it to Save/Get 00. Type this in with Scroll Lock turned off.

Type: [F5] SV [↵]

Type: 91, [↵]

Result: The carriage return is saved to Save/Get 91. In Expanded Display, the Save/Get you just typed in looks rather odd, but nonetheless is correct:

```
«SV91,←
»
```

Then type in the following IF statement:

```
«IF«IS00»∈«IS91»==0»
```

If the contents of Save/Get 00 is a carriage return, this statement will be true.

NOTES

Using Redlining

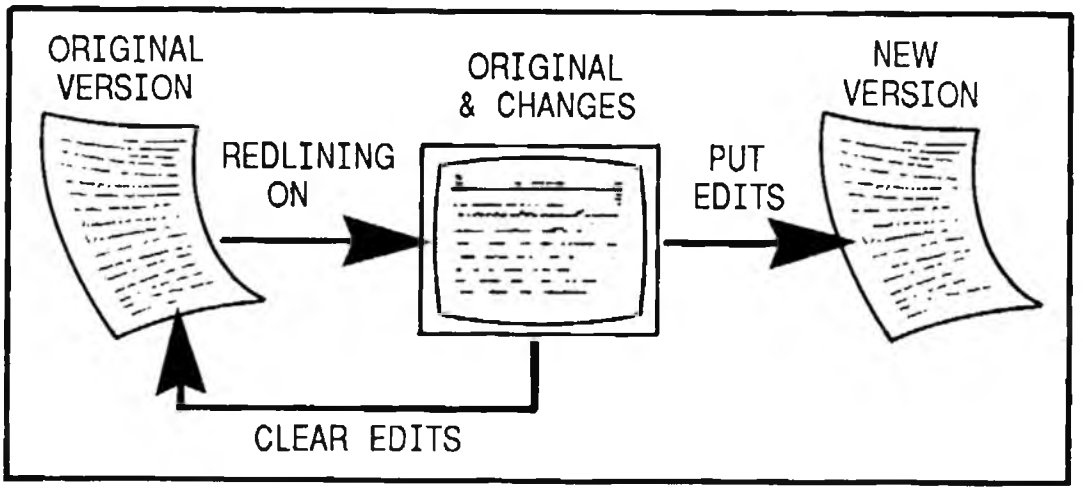
INTRO

Sometimes you want to keep track of the changes that you are making to a document. For example, if you are editing someone else's work, you may want the author to review your changes before finalizing them. Or you might be preparing a contract that you want to return for review with all changes marked.

XyWrite's **Redlining** lets you do this. It helps you keep a record of all additions and deletions. When you are ready to finalize the changes, you can use **Put Edit** to incorporate them. Conversely, you can use **Clear Edit** to remove them and restore the document to its original condition. With both commands, there is a **Verify** option to selectively review and incorporate the changes.

A Hidden Notes feature lets you make comments that are ignored by XyWrite. This makes it easier to pass along instructions that will not be printed.

CONTENTS



Turning Redlining On and Off

FORMAT

CM **RED ON**	Turns Redlining on.
CM **RED OFF**	Turns Redlining off.

This is an immediate command.

PURPOSE

The **RED** command lets you turn Redlining on and off within a window. When you turn Redlining on and then edit a document, you have a record of all of the changes you make. XyWrite uses character modes to keep track of all additions and deletions made to a file while you are in Redlining.

With Redlining ON, XyWrite automatically displays all the additions you make in a *bold* character mode and all deletions in a *reverse* mode. This lets you tell at a glance where changes have been made. It does not mean, however, that Redlining limits your use of character modes.

Each of the normally available eight character display modes (normal, bold, reverse, superscript, etc.) has both a corresponding Insert mode and Delete mode.

To insert text, simply press Ctrl and the appropriate number key as you normally would. With Redlining on, XyWrite will enter this new text with an Insert character mode that corresponds to your selected mode. Later, when you use the Put Edit command to incorporate this new text, XyWrite converts it to the appropriate character mode.

Similarly, XyWrite uses special Delete modes to keep track of deleted text. Although all deleted text appears on the screen as reverse, its original mode is preserved in the background. Thus, if you use the Clear Edit command to restore deleted text, it is restored to its original mode. When you use the Put Edit command, the text is removed from the file.

See Note #6 for more information about Redlining character modes.

ACTION

Keeping Track of Edits in a File.
To keep an on-screen record of the changes you make to a file:

1. Turn Redlining on.

   Type: F5 red on ↵

   Result: Redlining is now active. The letter "R" appears in reverse mode at the top right corner of the screen. Turning Redlining on also automatically puts you in Insert mode.

2. Call the file you want to edit. We'll use CHAPTER.DOC.

   Type: F5 call chapter.doc ↵

3. Make whatever additions and deletions you wish to the text.

   Result: Text that you add appears in bold. Text that you delete appears in reverse.

4. When you are finished editing CHAPTER.DOC, store it.

   Type: F5 store ↵

   Result: The edited file is stored to disk. You can call it back to the screen at any time to review or incorporate the edits you made. The letter "R" still appears in the upper right corner of the screen, indicating that Redlining is still active.

5. Repeat Steps 2 through 4 for each file you want to edit using Redlining.

6. When you are finished with the editing session, turn Redlining off.

   Type: F5 red off ↵

   Result: The letter "R" disappears from the top right corner of the screen. You can resume normal editing; no further record will be made of your changes.

7. If you later want to remove the editing tracks, see "Putting the Edits into a Document" or "Clearing the Edits from a Document."

NOTE #1 **Correcting Mistakes.** If you make a mistake when you are deleting original text, you can undo it with a second deletion. For example, if you are deleting by character with the Delete key and you go one character too far, you can back up by simply using the Backspace key.

Similarly, if you delete a word with Alt-Del but then change your mind, just move the cursor to that word again and press Alt-Del. The word changes from reverse mode to its original mode.

Deleting text you have just inserted is immediate. No record is made of these changes. For example, if you add the sentence "It was a very dark and stormy night," you can delete the word "very." In this case, the word disappears; it is not displayed in reverse character mode since you had just inserted it.

NOTE #2 **Window Status.** When you turn Redlining on and off, you are changing the status of the active window, not of the document you have on display. You must turn Redlining on and off in each window you want to use.

NOTE #3 **Normal Display.** Redlining works only in Normal Display. If you try to switch to Expanded Display, XyWrite displays the message "Function not allowed in Redlining mode."

NOTE #4 **No Overstrike.** Redlining works only in Insert mode. When you invoke Redlining, text entry will be in Insert mode, even if you are in one of the Overstrike modes.

NOTE #5 **Assigning a Toggle Key.** If you will be using Redlining frequently, you can assign a Redlining On/Off toggle to a key in your keyboard file. (See "Keyboard File" in Chapter 6.) The function call for this toggle is RO (Redline On/Off).

NOTE #6 **Character Modes.** XyWrite embeds special codes into the file so that it can automatically convert the Insert and Delete modes into the modes that you specified (or restore them to their original state). The following conversion table is used:

Standard Modes	Insert Modes	Delete Modes	Description
MD NM	MD IN	MD DN	Normal
MD BO	MD IB	MD DB	Bold
MD UL	MD IU	MD DU	Underline
MD RV	MD IR	MD DR	Reverse
MD BU	MD IL	MD DL	Bold Underline
MD BR	MD IV	MD DV	Bold Reverse
MD SU	MD IS	MD DS	Superscript
MD SD	MD ID	MD DD	Subscript

NOTE #7 **Changing Display Modes.** You can change the way the various character modes appear on the screen by modifying the printer file. For example, if you have a color monitor, you might want new "normal" text to appear in bright white characters on a blue background. Call up your printer file and enter the following definition:

```
MD IN=31
```

NOTE #8 **Printing a Document.** You can use the TYPE command to print a copy of your document with the Redlining markers in place. XyWrite III Plus printer files define INSERT and DELETE print attributes that enable you to identify text you inserted or deleted with Redlining on.

When your document is printed, the Insert modes (IN, IB, IU, etc.) assume the print characteristics of the corresponding standard mode with the additional effect defined by the INSERT attribute. For example, the INSERT attribute is normally defined to set off inserted text in square brackets, so text in Insert Bold mode (MD IB) would print out as [, followed by bold text, ended by].

Similarly, the Delete modes (DN, DB, DU, etc.) assume the print characteristics of the standard modes with the additional effect defined by the DELETE attribute. For example, the DELETE attribute is typically defined to strike over deleted text with a backslash (\), so Delete Bold mode (MD DB) would print text in bold mode with a backslash through each character.

If you want, you can change the way inserted or deleted text prints out by modifying your printer file. For example, you might want to take advantage of a color printer and set up your printer file to print inserted text in blue and deleted text in red. You can make this change by (1) adding mode definitions in the Printer Tables (e.g., MD IN+PICA+BLUE) or (2) changing the definition of the INSERT or DELETE attributes. When you add a mode definition in a Printer Table, it only affects text that is printed when that Printer Table is active. If you change the definition of the INSERT or DELETE attribute, you change the printed appearance of all inserted or deleted text. (See "Printer Files" in Chapter 6 for more information on modifying the Printer File.)

Keep in mind that due to the extra text that will be on edited pages, the page breaks may be different when the edits are finalized with the Put Edit command.

NOTE #9 **Using** Ctrl 0. When Redlining is on, Ctrl 0 performs the same function as Ctrl 1. That is, it activates normal mode rather than the adaptive mode.

Putting the Edits into a Document

FORMAT

CM **PE**	Put Edits	(Option 1)
CM **PEV**	Put Edits with Verify	(Option 2)

These are immediate commands.

PURPOSE

The Put Edit commands (**PE** and **PEV**) incorporate the changes made with Redlining On. The conversion begins with the file on screen from the point of the cursor to the end of the file.

Redlining uses special character modes to mark additions and deletions in a file. The Put Edit commands search for these character modes and make the indicated changes. They convert inserted text to the appropriate character modes (see Note #6 under "Turning Redlining On and Off") and erase text that is marked for deletion. (See also the companion command CE in "Clearing the Edits from a Document," the following section.)

You have two choices for incorporating the changes:

- Putting in all the edits at once (Option 1)
- Putting in the edits selectively (Option 2)

You can use these choices either on a defined block of text or on an entire file.

ACTION
(Option 1)

Putting in All the Edits.
Suppose you have reviewed the edits made to a file and know that you want to incorporate *all* of them into your original document.

1. Move the cursor to the beginning of the file.

   Press: Ctrl – Home

2. Enter the Put Edit command.

   Type: F5 pe ↵

   Result: All the editing tracks in the file are removed.

ACTION
(Option 2)

Putting in the Edits Selectively.
To search a file for editing tracks and have XyWrite stop at each one to verify that you want it incorporated into your original text:

1. Move the cursor to the point in text where you want to begin putting in the edits.

2. Enter the Put Edit, Verify command.

   Type: F5 pev ↵

   The cursor moves to the first change in the file. This could be added text (displayed in bold mode) or deleted text (displayed in reverse mode).

3. Verify whether or not XyWrite should incorporate the change. Type A, Q, S, N, Y, or U:

   A Do not incorporate the change. Abandon the process and return to the starting point.

   Q Quit. Do not incorporate the change. Abandon the process and stop at the current point.

   S Stop. Incorporate this change and then abandon the process. Return to the starting point.

   N No, do not incorporate this change but continue to the next change.

   Y Yes, incorporate this change and continue to the next change.

   U Undo this change and continue to the next change.

4. Repeat Step 3 until you have reviewed all the changes.

NOTE

Renaming Your File. After you have revised your file, you may want to store the file with a new filename. This can help you to keep track of the various versions that are generated with each new generation of edits.

Clearing the Edits from a Document

FORMAT

CM **CE**	Clear Edits	(Option 1)
CM **CEV**	Clear Edits with Verify	(Option 2)

These are immediate commands.

PURPOSE

The Clear Edit commands (**CE** and **CEV**) remove the changes made to a document during Redlining. Like the Put Edit commands, the Clear Edit commands search for the Redlining character modes, but, instead of making the indicated changes, they restore the text to its original state.

The conversion begins with the file on the screen from the point of the cursor to the end of the file. You can use either command on a defined block of text or on an entire file.

You have two choices for clearing the edits:

- Clearing all the edits at once (Option 1)
- Clearing the edits selectively (Option 2)

ACTION
(Option 1)

Clearing All the Edits.
Suppose you have reviewed the edits made to a file and know that you want to discard *all* of them and restore your document to its original condition.

1. Move the cursor to the beginning of the file.

   Press: Ctrl-Home

2. Enter the Clear Edit command.

   Type: F5 ce ↵

Result: All the editing tracks in the file are removed. Text that was in reverse is restored to its original mode, and text that was in bold is deleted.

ACTION
(Option 2)

Clearing the Edits Selectively.
To search a file for editing tracks and to have XyWrite stop at each one to verify that you want to remove the change:

1. Move the cursor to the point in text where you want to begin the cleanup.

2. Enter the Clear Edit, Verify command.

   Type: F5 cev ↵

   Result: The cursor moves to the first change in the file. This could be added text (displayed in bold mode) or deleted text (displayed in reverse mode).

3. Verify whether or not XyWrite should remove the change. Type A, Q, S, N, or Y.

   A Do not remove the change. Abandon the process and return to the starting point.

   Q Quit. Do not remove the change. Abandon the process and stop at the current point.

   S Stop. Remove this change and then abandon the process.

   N No, do not remove this change but continue to the next change.

   Y Yes, remove this change and continue to the next change.

4. Repeat Step 3 until you have reviewed all the changes.

NOTE #1

Recovering from a Mistake. Remember these commands are removing previous edits. Care should be taken to insure that you don't inadvertently lose any of your work. If you do, don't forget that you can *ABORT* the file and recall it to begin again.

FORMAT **CM LB** Label

This is an embedded command.

PURPOSE The **LB** (Label) command lets you make notes in your document that are ignored when you output the file to the printer. You can write as many notes or comments as you want – information about reference sources, ideas you want to expand on later, questions that need further research, etc. And while all of this information is conveniently stored for you, it does not clutter up your printed copy.

To enter labels, you use a procedure similar to the one used to enter footnotes or running headers. When you execute the LB command, an editing window opens so you can enter the text of your note. When you close the window, an embedded triangle appears, followed by a superior "n" and the first four characters of the Label field in reverse mode. The reverse mode text after the embedded triangle lets you easily distinguish your notes from the formatting commands represented by other embedded triangles.

You can reopen the window at any time so you can read or edit the text of a note.

ACTION **Entering a Comment.**
To enter a note for yourself as a later reminder:

1. Position the cursor in the file where you want the note.
2. Enter the Label command.

   Type: F5 lb ↵
3. Type the text of your note.
4. Close the note window.

   Press: F3

Result: An embedded triangle appears on the screen, followed by a superior "n" and the first four letters of your note in reverse mode.

ACTION **Reviewing a Note.**
To review the contents of a note:

1. Position the cursor on the embedded triangle for the note you want to review.

2. Press: Ctrl F3

   Result: The text of the note appears in a window on the screen.

3. When you have finished reviewing the note, close the window.

   Press: F3

4. Repeat steps 1-3 for each note that you want to review.

NOTE #1 **Size Limit.** There is no practical limit to the size of the notes that you can leave. However, keep in mind that you may slow down normal editing if you insert many long notes (several KB each).

NOTE #2 **Right Margins.** The labels do not affect the line breaks. Since the labels do not take up any space on output, XyWrite does not break the line until the printing text reaches the right margin. This can mean that, as you add label notes to a line, text may apparently be pushed past the right margin. This will not be the case when it is printed.

TIP **Several Origins.** If several people are making comments about the text, it's a good idea to have each reviewer start the label text with their initials. That way, the various notes from each person are easily recognized.

Memory Usage

INTRO

Information gets loaded into memory in different ways. Every time you start XyWrite from DOS, the program is loaded into memory. Then, when XyWrite runs the program STARTUP.INT, the files that you have selected to be part of your standard system configuration are loaded into memory. When you display a file on screen, it is loaded in memory too. When you use the XyWrite Spelling Checker, the program SPELL.OVR and the dictionary DICT.SPL are loaded into memory.

Depending on the options you select for your system configuration (Help files, printer files, etc.) and the size of the documents you are working with, XyWrite III Plus can occupy from 140K to 640K of memory. Because there are times when you don't need all of the XyWrite functionality that is in memory, the program lets you release some of the memory it is using and later lets you reload the memory you released without rebooting the system.

Several sections of XyWrite code can be released. Each section is associated with a major XyWrite feature. You can selectively release the sections that you are not currently using and still maintain the functionality you need. XyWrite automatically reloads the released sections of code as you give commands that use them.

The most common application of releasing a section of XyWrite code is to unload the Spelling Checker program and its dictionaries, since they occupy large pieces of memory. Later, when you give a command that requires the Spelling Checker, it reloads automatically. (If you want, you can reload it by command.) This section describes the ways of loading and unloading parts of the XyWrite III Plus program.

CONTENTS

Memory Usage Menu

FORMAT Ctrl M Display Memory Usage Menu

This is an immediate command.

PURPOSE The programming information or *code* needed to support several major features of XyWrite III Plus has been separated from the main part of XyWrite. Ctrl M displays a menu that lists each feature and tells you how much memory each one of these is using, both for its code and for its associated files. It also tells you how much memory is still available.

With the memory usage menu on the screen, you can review and change the status of each of these features. Using this menu, you can:

- Release the memory used by the code associated with a feature. To do this, select the UNLOAD PROGRAM option and press the feature number.

- Release the memory used by the data associated with a feature (for example, a printer file). To do this, select the UNLOAD FILE option and press the feature number.

- Release the memory associated with DICT.SPL. Because this spelling dictionary is so large, releasing it frees up a significant piece of memory. It is treated as a special case here because it is the part that is most commonly released. To release DICT.SPL without releasing the temporary dictionary or any personal dictionaries you have loaded into memory, select the UNLOAD FILE option and press - (hyphen).

Once you have released a file or code from memory, it must be reloaded before you can use the associated feature again. XyWrite automatically reloads the code when you execute the function associated with it. For example, if you unload the Spelling Checker and later issue the SPELL command, XyWrite loads SPELL.OVR and DICT.SPL. However, if you have released a data file from memory, such as a personal spelling dictionary, you must specifically reload it.

As a reminder, the memory usage menu contains a line that says LOAD FILE. When you select that option and press [↵], the menu clears from the screen and the LOAD command appears in the command line. Enter the name of the file you want to load and press [↵] (for more information, see "Loading Customization Files").

ACTION

Unloading a File from Memory.
Suppose that you are formatting a very long document and that you have already run Spelling Checker on it. Suppose also that you won't need to run Spelling Checker for a while, but that you could use some extra memory so you can load all of your document.

1. Display the memory usage menu.

   Press: [Ctrl] [M]

2. Use the cursor keys to select the UNLOAD FILE option.

   Result: The UNLOAD FILE line is highlighted.

3. Press: -

   Result: The 140K of memory used by the main spelling dictionary (DICT.SPL) is released. When you use one of the spelling functions, XyWrite automatically reloads the dictionary.

4. If you need more memory, select another feature that you aren't using at the moment and release it. Otherwise:

   Press: Esc

NOTE #1 **Unloading the Temporary Dictionary.** When a temporary dictionary is unloaded, its contents are deleted from memory. (See "Storing the Temporary Dictionary" in Chapter 3.)

ACTION **Loading a File Into Memory.**
Let's say you want to reload your personal dictionary PERS.SPL.

1. Display the memory usage menu.

   Press: Ctrl M

2. Use the cursor keys to select the LOAD PROGRAM option.

3. Press: ↵

   Result: The menu clears from the screen and the LOAD command appears in the command line.

4. Type: pers.spl↵

   Result: PERS.SPL is loaded into memory.

NOTE #2 **Location of the Released Information.** If you are using a floppy disk system, you may want to have the released code stored in a temporary file so you don't have to keep your program disk available. (For more information, refer to the following section "Saving Sections of Code.")

Saving Sections of Code

FORMAT

CM **SAVEC** *n,n...*

n,n... are the numbers associated with a XyWrite feature.

This is an immediate command.

PURPOSE

If you have a floppy disk system or if you run the XyWrite editor from a floppy disk, you may want to reload released code without using your program disk. The **SAVEC** (Save Code) command lets you save parts of XyWrite code to files on your current disk (see Note). You can then release those sections of code from memory. If you later execute a function that requires the released code, XyWrite calls the information from the file created by SAVEC, rather than from EDITOR.EXE on your program disk.

The SAVEC command uses the number associated with each releasable section of code. The numbers and their associated features are listed in the memory usage menu. To display the menu, press Ctrl M.

When the code is saved to disk, it is saved in file(s) named EDITOR.OV*n*, where *n* is the number of the feature you are saving. For instance, if you specify SAVEC 1, XyWrite creates the file EDITOR.OV1 on disk. The command SAVEC 0 command creates the files EDITOR.OV1 through EDITOR.OV7.

ACTION

Using the SAVEC Command.
Let's save all of the releasable code to files on disk.

Type: F5 savec 0 ↵

Result: The files EDITOR.OV1 through EDITOR.OV7 are saved on disk. You can now use the memory menu or UNLOAD command to release one or more sections of code. If XyWrite needs to reload the code to perform an function, it retrieves the code from the appropriate EDITOR.OV*n* file.

NOTE

Location of Saved Code File. When XyWrite creates an EDITOR.OV*n* file, it saves it on the drive you specify with the DEFAULT DR setting. If you aren't sure what drive you specified, display the window menu. If an EDITOR.OV*n* file exists, the letter of the drive where it is saved appears under "Disk."

Releasing Memory

FORMAT

CM **UNLOAD** *n,nF,##...*

n is a number (0-7) of a XyWrite feature.
F releases the file for specified feature.
can be DICT.SPL or the two-letter customization label (SP, PR, etc.)

This is an immediate command.

PURPOSE

The **UNLOAD** command releases code or files that have been loaded into memory. You can perform the same functions using the memory usage menu, but this command is useful when writing a program using XyWrite's programming language.

The LOAD command uses the number associated with each releasable section of code. The numbers and their associated features are listed in the memory usage menu. (To display the menu, press Ctrl M.) It also accepts the customization file labels (see "Loading Customization Files for a list of customization file labels).

ACTION

Using the UNLOAD Command.
Let's look at some examples of the UNLOAD command.

Type: F5 `unload 3,dict.spl` ↵

Result: DICT.SPL and XyWrite's Spelling Checker (feature #3) are no longer in memory.

Type: F5 `unload 3f` ↵ or
Type: F5 `unload sp` ↵

Result: These are equivalent commands. Both unload the personal dictionaries from memory.

Type: F5 `unload 0,0F` ↵

Result: The "0" refers to all features and the "0F" refers to all files (including DICT.SPL). This command releases the maximum amount of memory.

Loading Customization Files

FORMAT

CM **LOAD** *file1 file2 file3 +file4...* (Option 1)
CM **LOAD** *+file1 +file2 +file3...* (Option 2)

file1, *file2*, *file3*, and so on are names of XyWrite customization files.

This is an immediate command.

PURPOSE

LOAD lets you load all of your customization files with a single command. Each customization file must start with a special 4-character label that tells the LOAD command what type of file it is. The file types and the label associated with each are listed below.

- Printer file ;PR;
- Substitution file ;SU;
- Help file ;HL;
- Hyphenation dictionary ;HY;
- Personal spelling dictionary ;SP;
- Sort file ;SO;
- Keyboard file ;KB;

You must type the labels exactly as shown: semicolon, capitalized 2-letter identifier, semicolon. Put the label on the first line of the customization file, and end it with a carriage return.

Once your customization files are identified, you can use the LOAD command to:

- Initially load into memory one or more customization files. You can specify more than one spelling dictionary and more than one printer file within the same command by inserting a plus sign immediately before the filename of the additional files (Option 1).

- Append the information in a customization file to what is already loaded into memory. This option allows you to load a personal dictionary or printer file into memory without deleting the personal dictionaries or printer files that you loaded previously (Option 2).

ACTION *(Option 1)*

Loading Customization Files.
To load the printer files and hyphenation dictionary into memory with one LOAD command:

1. Be sure the files you want to load start with the proper identifiers.

2. Issue the LOAD command.

   Type: F5 `load 3epsonfx.prn+b-w.prn,diction` ↵

   Result: The printer files 3EPSONFX.PRN and B-W.PRN, and the hyphenation dictionary DICTION are all loaded into memory.

ACTION *(Option 2)*

Appending Printer Files or Spelling Dictionaries.
Suppose you are working on a document that uses a lot of business terms. You loaded PERS.SPL when you started your system, but now you want the Spelling Checker to reference the business dictionary BUS.SPL as well.

Type: F5 `load +bus.spl` ↵

Result: The words in BUS.SPL are loaded into memory, appended to the words in PERS.SPL. (If you omit the "+," the command would clear the temporary and personal dictionaries already in memory.)

NOTE

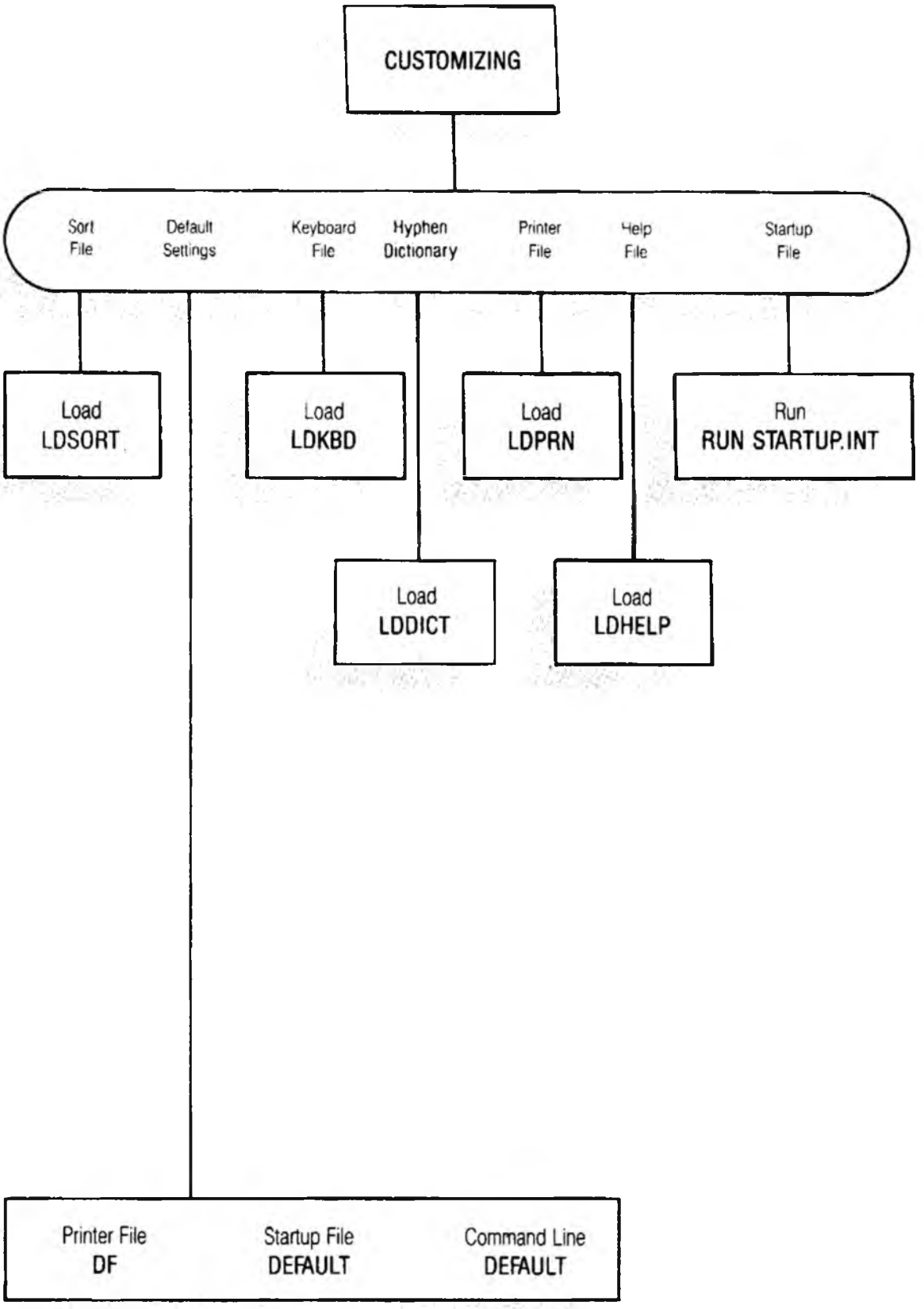
Loading DICT.SPL. Because this dictionary is loaded automatically whenever you use the SPELL command, it is unlikely you would want to load it with the LOAD command. However, the LOAD command does recognize DICT.SPL; it is a special customization file that does not require an identifying label.

Customizing

INTRO

One thing that sets XyWrite apart from other word processors is the ability to tailor the program to suit your own needs. This section describes how you can customize the help screens, the keyboard, and your printer. You can also modify how XyWrite hyphenates words and sorts indexes. And you can program XyWrite to perform any number of tasks as part of your start-up routine.

CONTENTS


CUSTOMIZING
Sort File
Default Settings
Keyboard File
Hyphen Dictionary
Printer File
Help File
Startup File
Load LDSORT
Load LDKBD
Load LDPRN
Run RUN STARTUP.INT
Load LDDICT
Load LDHELP
Printer File DF
Startup File DEFAULT
Command Line DEFAULT

Default Settings

INTRO

When XyWrite first starts up, it provides you with its own set of defaults (listed later under "Restoring Defaults"). The DEFAULT command allows you to set your own defaults, so that XyWrite will start up customized to your needs.

CONTENTS

Default Settings

PURPOSE

Default settings let you change the value of XyWrite settings. These settings are listed in the following table "Default and VA Settings." You use the DEFAULT command to enter these settings. For example, to change the default right margin to 50:

F5 default rm=5 ↵

The DEFAULT command is described at the end of this section.

Along with Default commands, the VA (Value of Variable) commands are listed in the table. A VA command inserts the current XyWrite setting into the text. The VA command is described at the end of this section.

Default settings are divided into three categories:

- **Format Settings** — To set the initial format conditions at the start of every file. For example, you might use it to set the right margin at 70 and tabs every 5 spaces.
- **Printer and Display Settings** — To set up for your particular printer and display monitor.
- **System Settings** — To set controls specific to your system, such as enabling backup of files and setting the default drive for TMP files.

DEFAULT sets these conditions for *all* files, without requiring their entry into each file you create.

NOTE #1 **Resetting the Defaults.** After changing the default settings, if you wish, you can return to the original settings built into XyWrite with the DM (Default Margins) command.

NOTE #2 **Priority of Default Settings.** A command embedded in text takes priority over the same command entered with DEFAULT anywhere else — in the STARTUP.INT file, on the Command Line, or in the Printer File. For example, RM 80 embedded into the document takes precedence over a default setting of DEFAULT RM=70 entered in the STARTUP.INT file, Printer File or on the Command Line.

NOTE #3 **Default Command in Printer File.** When inserting the DEFAULT command into the printer file, abbreviate the word default with df. For example:

```
default rm=65          (On Command Line)
df rm=65               (In Printer File)
```

DEFAULT and VA SETTINGS

	Default (Example)	Description	VA	Value Returned (Example)
Format Settings (Chapter 4)				
	default ap	Auto-Pause	(NA)	
	default fc	Flush Center	va fc	fl
	default fd=66	Form Depth	va fd	66
	default fl	Flush Left	va fl	fl
	default fr	Flush Right	va fr	fl
	default hy=1	Hyphenation	va hy	1
	default ip=5,15	Indent Paragraph	va ip	5,15
	default lm=5	Left Margin	va lm	5
	default ls=2	Line Space	va ls	2
	default md=bo	Character Modes	va md	15
	default np	No Pause	(NA)	
	default of=5,0	Offset	va of	5,0
	default pl=54	Page Length	va pl	54
	default pt=1	Print Type	va pt	1
	default rm=65	Right Margin	va rm	65
	default rt=1	Relative Tabs	va rt	1
	default sp=5	Set Page Number	(NA)	
	default tp=6	Top Margin	va tp	6
	default ts=5,10,15	Tab Set	va ts	5,10,15
	(NA)	Bottom Margin	va bt	2
Printer and Display Settings (Chapter 6, Printer File section)				
*	**default al=1**	Automatic Leading	va al	1
	default bs=1	Backspace	va bs	1
*	**default bw=1**	Black & White Display	va bw	1
*	**default bx=15**	Window Border Colors	va bx	15,1,2,3,4,5,6,9,10
*	**default cr=112,7**	Cursor Type	va cr	112,7
*	**default dd=119**	Display of Defined Blocks	va dd	119
	default dt=1	Display Type	va dt	1
*	**default du=12**	Display Units	va du	12
	default eg=1	EGA Control	va eg	1
	default ej=1	Eject Last Page	va ej	1
	default ff=0	Form Feed	va ff	1
*	**default hn=31**	Header Normal Mode	va hn	31
*	**default hr=100**	Header Reverse Mode	va hr	100
*	**default hv=6,3,2**	Hyphenation Variables	va hv	6,3,2

*	**default jl=1**	Justify Underline Chars.	va jl	1
*	**default jt=1**	Justification Type	va jt	1
*	**default mu=12**	Margin Unit	va mu	12
*	**default ms=6**	Microspace Factor	va ms	6
*	**default pd=1**	Pad Spaces	va pd	1
	default sl=43	Screen Length	va sl	43
*	**default sq=1**	Sequential Page No.	va sq	1
	default tb=0	Tab Character	va tb	0
	default ul=2	Underline Setting	va ul	2
	default vu=3	Vertical Unit	va vu	3
*	**default wo=1**	Word Overstrike	va wo	1
	default ws=1	Whole-Space Just.	va ws	1
*	**default xc=3**	Space Constant	va xc	3
*	**default xf=2**	Space Factor	va xf	2
	(NA)	Ruler Markers	va rl	▶◀
	(NA)	Window Border Characters	va wb	╔ ╗ ═ ╚ ╝ ‖

System Settings (Chapter 6, this section)

	default bk=0	Backup Files	va bk	1
	default di=1024,7,1	Directory Settings	va di	1024,7,1
	default dr=b	Default Drive for TMP	va dr	B
	default ep=1	Erase Prompt	va ep	1
	default er=1	Error Help	va er	1
	default fu=4,6	Footnote Unit	va fu	4,6
	default kc=8400	Key Click	va kc	8400
*	**default km=1**	Keyboard Mode	va km	1
	default lf=1	Line Spacing On-Screen	va lf	1
	default nc=1	Normal Carriage Return	va nc	1
	default nw=1	New Window	va nw	1
	default tf=1	Ignore Top Margin	va tf	1
	(NA)	Filename	va $fi	CHAPTER.DOC
	(NA)	Drive, Path and Filename	va $fp	B:\CLIENTS\DOE
	(NA)	Memory Available (x 1K)	va $me	411
	(NA)	Current Drive and Path	va $pa	B:\CLIENTS
	(NA)	Page No. at Cursor	va $pg	12
	(NA)	Window Number	va $wn	3
	(NA)	Window Status	va $ws	1

NA = Not applicable or not implemented

* = All of the settings in this list can be entered into the Printer File with the DF command. However, the settings marked with an asterisk (*) can also be entered into the Printer File alone, without the DF.

DEFAULT SETTINGS

The default settings are described in various places in this manual.

Format Settings.
For further descriptions of the Format Settings, see Chapter 4. Each command is described individually.

Printer and Display Settings.
The Printer and Display default settings are described in various places in this chapter:

- BW, BX, DD, DT, EG, HN, HR, PD, RL, SL and WO are described later in this chapter under "Printer File."
- DU, MU, MS, JL and JT are described in the "Microjustification" section later in this chapter.
- XC and XF are the same as the SC and SF settings described later in the "Microjustification" section.
- BS, CR, EJ, FF, HV, SQ, TB, UL, WB, WD, WS and VU are described in this section.

System Settings.
The System Settings are described in the section that follows.

DEFAULT BK

Backup of Files - Lets you turn on or off the backup of files. (The initial default is 1.)

default bk=0 *Don't* keep backup copies. (This saves room on your disk.)

default bk=1 *Keep* backup copies. With this setting, whenever you call up a file to edit, XyWrite keeps the original version of the file you called up — it does this at the first SAVE or STORE command, by changing its three-letter extension to .BAK.

BK works as follows: Let's say version A of the document CHAPTER is on your disk.

1. You CALL CHAPTER, and then edit it to version B.
2. The *first time* that you SAVE (or STORE) version B, XyWrite renames version A (which is still on disk) to CHAPTER.BAK. It then saves Version B under the name CHAPTER.

3. Any subsequent SAVE or STORE command updates CHAPTER, but does not alter CHAPTER.BAK (until you CALL CHAPTER again).

The purpose of the backup feature is to allow you to easily return to the version of your document as it was when you most recently loaded it with CALL. As you do intermediate saves, the backup copy does not change. To update the backup file throughout an editing session, STORE the document rather than SAVE it.

You can override the default BK command for any specific file by entering the BK embedded command at the start of that file. For example: F5 bk 1 ↵

DEFAULT BS **Backspace Control** - Enables your printer to print a backspace, whether or not your printer can recognize a backspace character. (The initial default is 1.)

default bs=1 Use this setting if you are using a printer which can perform a backspace — in other words, if the printer responds to the backspace character (■) ASCII Value 8 by backspacing one character.

default bs=0 Use this setting if the printer *cannot* perform a backspace (that is, if it ignores the backspace character). Then, when XyWrite encounters a backspace character, it performs the backspace function by printing that line in two passes.

DEFAULT CR **Cursor Type** - Lets you change the cursor to be non-blinking, or to change the color. This CR setting used to be called CT. (The initial default is 0,0.)

default cr=*n*,*h* Format for the CR setting. The first value *n* (normal) is the number of the display mode for the cursor normally, when you are editing text. The second value *h* (help) is the number of the display mode for the cursor when viewing help frames.

default cr=112,7 Example of the CR setting. By referring to the display table later in this chapter (in "Printer File"), you can tell that 112 is non-blinking black letters on a white background, while 7 is non-blinking white letters on a black background.

DEFAULT DI **Directory Settings** - Lets you modify the way a long directory (DIRL) is displayed. (The initial default is 0,6,0.)

DI is further described under the DIRL command in Chapter 2.

DEFAULT DR **Default Drive for TMP Files.** - Sets the default drive where you want temporary files to saved. (There is no initial default setting for DR.)

default dr=*d* General format of the DR setting, where *d* is the drive letter where you want to save the .TMP files.

default dr=c Example of the DR setting — causes all of the .TMP files to be saved to drive C.

DEFAULT EJ **Eject Last Page** - Lets you eject the very last page of a document. (The initial default is 0.)

default ej=1 This setting causes the very *last* page of a document to automatically eject from the printer. If Form Feed is off (**default ff=0**), it sends blank lines to eject the last page. If Form Feed is on (**default ff=1**), it sends a form feed character (♀) to eject the last page.

default ej=0 This setting prevents the *last* page of a document from being ejected from the printer.

DEFAULT EP **Erase Prompt** - Lets you enable the prompt "Do you wish to erase?" whenever you execute the ERASE or DEL commands, to reduce the risk of inadvertently deleting files. (The initial default is 0.)

EP is further described under the ERASE command in Chapter 2.

DEFAULT ER **Display Error Help Frame** - Enables an error screen to appear (from the help file) whenever an error occurs.

default er=1 The error/help frames are displayed whenever an error occurs. You can tailor the frames to say anything you want. This feature is useful when a user is training, because you can display information about how to continue. However, it can slow performance considerably. Refer to Help Files later in this chapter for details on how to construct help frames.

default er=0 The error/help frames are disabled.

DEFAULT FF **Form Feed Character** - Lets you insert a form feed character at the end of each page. (The initial default is 0.)

default ff=0 Causes XyWrite to send enough Carriage Return / Line Feed Characters at the end of each page to advance the paper to the top of the next page. (In the absence of any explicit setting, XyWrite defaults to FF=0.) The FD (Form Depth) setting determines the total number of lines XyWrite will send to the printer for each page.

default ff=1 Causes XyWrite to send a carriage return - line feed combination (♪◙), and a form feed character (♀), ASCII 12, at the end of each page, to advance the paper to the top of the next page. Exception: when you turn off Eject Last Page (with Default **default ej=0**), there is no Form Feed character sent at the end of the *last* page. Default FF=1 is supported on most printers.

You can substitute any string for the Form Feed character by specifying PG<*string* in your Printer File. See the Printer File for further details.

When would you use FF=1? When you want to send a form feed character (or any other string) at the end of the text on each page, rather than rely on FD (Form Depth) to insert the proper number of blank lines. Ideally, FF=0 and FF=1 would have the same effect, if FD were set properly. However, several Carriage Return / Line Feeds might not be equivalent to a Form Feed due to round-off errors or paper creep from friction feed (or to a wrong value for FD). If you notice the top-of-form creeping slightly, page after page, use FF=1. This allows the *printer* to keep track of the top of form. FF=1 also can shorten printing time.

DEFAULT FU **Footnote Unit** - Default FU takes two values, the first for footnotes and the second for reference commands. FU lets you account for the extra width of several footnote symbols or a large value in the reference command. This occurs for footnote symbols when restarting the sequence of footnotes on every page. See one of the notes under the SF (Set Footnote Number) command of Chapter 4 for more details. (The initial default is 3,5.)

DEFAULT HV **Hyphenation Value** - Lets you select the size of words to be hyphenated and the fewest number of characters allowed before and after a hyphen. For details, refer to the HV setting in the Display Settings section of "Printer File." HV used to be called HY. (The initial default is 5,2,2.)

Notice there is a related setting **default hy=***n* that simply turns hyphenation on or off.

DEFAULT KC **Key Click.** This setting controls an audible click generated by XyWrite at every press of a key. (The initial default is 0, which turns the key click off.)

default kc=*n* Format for the KC setting, where *n* sets the duration and tone of the key click as follows:

n = 256 x Duration + Tone

where Duration is 0 (short) to 64 (long duration) and Tone is 0 (high tone) to 255 (low tone). (The key click is produced from the speaker inside the computer.) You can lengthen the *duration* of the click, but keep the same tone by increasing the Duration number. You can lower the *tone* by increasing the Tone number.

default kc=8400 Example with the KC value set to 8400. More examples:

	High Tone	Medium Tone	Low Tone
Short Duration	256	383	511
	8192	8319	8447
Long Duration	16384	16511	16639

After you change the key click value, to make it take effect, clear all files and close all windows. Then execute the ABORT command.

If you put KC in your STARTUP.INT file, place it before any WINDOW command.

DEFAULT KM **Keyboard Mode** - Lets XyWrite run with memory-resident programs. (The initial default is 1.)

KM is further described in Appendix F, "Memory-Resident Programs."

DEFAULT LF **On-Screen Line Spacing.** Lets you show the line spacing on-screen as it will be printed. (The initial default is 0.)

default lf=1 Text is shown with the same line spacing on-screen as will print. For example, text which will print double-spaced is shown double-spaced on-screen.

default lf=0 All text is shown single-spaced on-screen (except for Column Tables, which are shown with line spacing as they will be printed).

DEFAULT NC **Normal Carriage Return.** Lets you choose whether to send the carriage return to the printer in normal mode or in the current mode. (The initial default is 1.)

default nc=1 XyWrite puts out carriage returns in normal mode (MD NM).

default nc=0 XyWrite puts out carriage returns in the mode that is in effect at the end of the line.

DEFAULT NW **New Window** - Enables XyWrite to open a new window when needed. (The initial default is 0.)

default nw=1 Enables automatic new windows. XyWrite opens a new window every time you execute NEW or CALL with a file on-screen. Allows you to display a directory, call a file, look it over, abort it, and then be returned to the directory where you can call up another file. It does this by automatically opening a new window when necessary and keeping the directory in an old window.

default nw=2 Same as NW=1 except XyWrite does not automatically close the current window when you execute ABORT.

default nw=0 Disables automatic windows. You must open and close windows manually.

NW is described further in a note titled "Automatic New Window" under the CALL command in Chapter 2.

DEFAULT SQ **Sequential Page Numbering.** SQ changes the way the TYPE,*a-b* command counts pages as it prepares to send them to the printer. (The initial default is 0.)

default sq=1 Sets the TYPE command to refer to the pages sequentially, starting at the beginning of the file. This setting disregards the printed page numbers (which can be changed with SP, Set Page Number). Thus, **type chapter,5** would print the fifth page in the file counting from the start of the file, regardless of its printed page number.

default sq=0 This is the way XyWrite is normally set up. This setting enables the TYPE command to honor the printed page number. Thus, **type chapter,5** would print the page which has page number 5 printed on it. This is the number the Page-Line indicator shows. See the SP (Set Page Number) command in the Header & Footer section of Chapter 4 for more details.

DEFAULT TB **Tab Character Control** – Lets you print each tab either as a tab character or as spaces. (The initial default is 0.)

default tb=0 Causes XyWrite to convert every tab to equivalent spaces on output to the printer.

default tb=1 Causes XyWrite to send tabs in the file to the printer as the tab character (^I), ASCII Value 9. This setting is valuable for printing to disk (TYPEF) when you want to retain tab characters. Use this setting when you want to produce a file (with TYPEF) that keeps its tabs (rather than converts them to spaces).

DEFAULT TF **Ignore Top Margin** – Lets XyWrite ignore the top margin (TP) embedded command. (The initial default is 0.)

default tf=1 Causes XyWrite to ignore the top margin (TP) command. You would use this if the top margin has been set by hand and the file contains a TP command that you want to override.

default tf=0 – Causes XyWrite to honor the top margin (TP) command. See the "Page Length" section of Chapter 4 for details on the TP command.

DEFAULT UL **Underline Setting** – Specifies how tabs and spaces are underlined within text that is underlined with MD UL.

For details, see the note "Controlling Underlining" in the Character Mode section of Chapter 4.

DEFAULT VU **Vertical Unit** – Specifies number of basic units in a vertical unit. Similar to DU. For details, refer to the description of the VU setting in the Printer File section later in this chapter. (The initial default is 1,1,100.)

DEFAULT WB **Window Border Characters** – Defines the characters that are used for the borders of the windows. Use the less-than sign (<) when specifying the characters. (The initial default is shown below as an example.)

default wb<*ul,ur,hz,ll,lr,vt* Format of WB setting, where *ul* is the character in the upper left corner, *ur* is upper right corner, *hz* is horizontal sides, *ll* is lower left corner, *lr* is lower right corner, and *vt* is vertical sides.

default wb< ╔╗═╚╝‖ Example of WB setting.

DEFAULT WS **Whole-Space Justification** - Lets you choose between whole-space and micro-space justification. (The initial default is 0.)

default ws=1 - When you specify justification (with JU), this setting causes XyWrite to justify text by adding *whole* spaces between words (rather than using partial spaces) and no space within words. Whole-space justification is much faster than microjustification on some printers, and is preferred in those cases.

default ws=0 - When you specify justification (with JU), this setting causes XyWrite to jusify text using *partial* spaces — what we call "microjustification".

VA SETTINGS

VA Settings. VA settings indicate the status of XyWrite. The value is inserted at the current cursor location.

VA $FI **Current Filename** - Displays the name of the current file.

VA $PA **Current Drive and Path** - Displays the current drive letter and path.

VA $PG **Current Page** - Displays the page number where the cursor is currently located.

VA $ME **Memory Available** - Shows the amount of memory (RAM) currently available, in kilobytes, after XyWrite is loaded. Thus, 312 means 312K of memory is available for more files or other programs.

VA $WN **Window Number** - Displays the number of the window number that is currently active (0-9).

VA $WS **Window Status** - Displays the status of the window where the cursor is currently located:

0 No file open, window is empty
1 File is open
2 Directory is displayed

DEFAULT CK

Spelling Checker. The DEFAULT CK (Checker) setting allows you to select certain options for Spelling Checker operation.

default ck=0 This is the default. It causes the Spelling Checker to check words that contain numbers and letters (e.g., 12th) and to ignore the automatic replacement feature with the SPELL and CORRECT commands.

default ck=1 This setting causes the Spelling Checker to ignore all words that start with a number.

default ck=2 This setting causes the Spelling Checker to use the automatic replacement feature with the SPELL and CORRECT commands.

default ck=3 This setting is a combination of the last two settings. It causes the Spelling Checker to ignore all words that contain numbers and to use the automatic replacement feature with the SPELL and CORRECT commands.

DEFAULT HS

Header Size. The DEFAULT HS (Header Size) setting allows you to set the size of the command field. This is useful if you are using a memory-resident program that puts information onto the command line. When you define the size of the command field, XyWrite ignores data that is beyond the point you specify. That means you can use the command field to do searches, compile indexes, and run programs without having XyWrite pick up data from the memory-resident program as part of its command string.

default hs=*n* Format of the HS setting, where *n* is the number of characters in the field. The default setting is 79, which means the entire command line is used.

default hs=45 Example of HS setting.

Default SW = 40, 80, 120, 132

DEFAULT ST

Show Tabs. The DEFAULT ST (Show Tabs) setting allows you to select the way tabs are displayed in expanded mode.

default st=0 This is the default. It causes tabs to be displayed in expanded mode as spaces.

default st=1 This setting causes the tab character to be displayed in expanded mode.

DEFAULT WO

Word Overstrike Editing. The DEFAULT WO (Word Overstrike) setting allows three overstrike typing modes for text entry.

default wo=0 This is the default. It causes text entered at the cursor to overwrite text characters and word separators *except* carriage return and tab characters.

default wo=1 This setting causes text entered at the cursor to overwrite all text and word separators *except* the space, tab, and carriage return characters.

default wo=2 This setting causes text entered at the cursor to overwrite text characters but not word separators.

AUDIBLE SIGNALS

Audible Signals. XyWrite uses several audible signals to notify you that a certain action has happened. You can define the tone and duration of each of the following signals by entering the setting:

DEFAULT ##=tone,duration

where ## is one of the following four settings, and *tone* and *duration* are numbers from 0 to 65534.

DEFAULT OB

Overstrike Beep. The OB (Overstrike Beep) setting allows you to define the tone and duration of the beep you hear whenever you overstrike a character. The initial setting is 0,0 which turns the overstrike beep off.

DEFAULT EB — **Error Beep.** The EB (Error Beep) setting allows you to define the tone and duration of the beep you hear whenever XyWrite detects an error. The initial setting is 1536,12288.

DEFAULT CB — **Correction Beep.** The CB (Correction Beep) setting allows you to define the tone and duration of the beep you hear whenever Spelling Checker automatically corrects a misspelling for you. The initial setting is 512,4096.

DEFAULT KC — **Key Click.** The KC (Key Click) setting controls an audible click generated by XyWrite every time you press a key. The initial setting is 0,0 which turns the key click off.

VA SETTINGS

VA $WC — **Word Count.** Displays the total number of words counted by the WC, WCB or SPELL command (whichever was most recently executed).

VA $BD — **Bad Words.** Displays the number of questionable words found by the SPELL command when it was last executed.

NOTES

VA $FS **File Status** – Displays the status of the files currently open:

0	No files are open
Non-zero	At least one file is open

VA $DT **Display Mode** – Indicates the current display mode. Similar to the setting va dt.

0 Expanded Display
1 Normal Display, no page-line numbers, no markers
2 Normal Display, w/page-line numbers & markers
3 Normal display, w/page-line numbers, no markers
4 Normal display, no page-line numbers, no markers
5 Value of VA $DT when the file is typed to screen (TYPES)
8 Value of VA $DT when the file is printed to printer (TYPE)

FORMAT

CM DEFAULT *nm=#,nm=#,...* (on Command Line or in STARTUP.INT File)

DF *nm=#,nm=#,...* (in Printer File)

nm is the two-letter name of the command
= (equal sign) separates the name from the value
is the value of the command
, (comma) separates commands
DEFAULT is an immediate command.

PURPOSE

The DEFAULT command lets you change the value of certain XyWrite settings.

For example, if you don't want backup files to be saved (because you need more room on your disk), you can turn off the backup of files with the command DEFAULT BK=0.

When you change the DEFAULT settings, the change affects *all* files — not only newly created files, but also already existing files.

You can execute the DEFAULT command from the Command Line; or you can store it in any of several files:

- Your Printer File (e.g., 3EPSONFX.PRN), where it can be kept with all the other printer commands.
- The STARTUP.INT file.
- Another file you create just for defaults (you might call it DEFAULT.PRN).

In which file do you put the default settings? To give a general recommendation: If the default setting is associated with your printer, then put the default in your Printer File. Otherwise, put it elsewhere:

- Put the Printer & Display Settings into your Printer File.
- Put the Format Settings and System Settings into STARTUP.INT or DEFAULT.PRN.

ACTION
(Option 1)

Entering Defaults into the Printer File.
To enter defaults into the printer file, do the following:

1. **Call the Printer File.** Call to the display the printer file you use. For example:

   Type: F5 ca 3epsonfx.prn ↵

2. **Find the Default Command.** Search through the Printer File for the DF command:

   Type: F5 se /df/ ↵

   If the default command is *not* present, then add it in the next step.

3. **Enter the Default Settings.** Type into the default statement the settings you want. For example, to set the tabs at 5, 10, 15, right margin at 70, and backup of files turned on, type the following statement. (Type this on its own line, with one space after the DF and no other spaces.)

   ```
   df ts=5,10,15,rm=70,bk=1
   ```

4. **Store and Load the Printer File.** To make the Default command take effect:

   Type: F5 store ↵

   Type: F5 ldprn 3epsonfx.prn ↵

   Result: *All* new documents now begin with these format conditions. Likewise, all existing documents will take on these settings unless they have explicit format settings (TS or RM) in them to override the defaults. (For example, you could override the default right margin within the document by entering the embedded command RM 80.)

5. **Verify the Default Command.** To demonstrate that the margin and tab settings you made in Step 3 are now indeed active, open a new file:

   Type: F5 new test ↵

Result: By looking at the ruler, you should see that the new file has tabs at 5, 10, and 15, and a right margin at 70 (and backup of files turned on, which is not evident until you save a file).

ACTION *(Option 2)*

Entering Defaults into STARTUP.INT.
For this procedure, refer to the section on the STARTUP.INT file later in this chapter. To enter the same settings as in the previous procedure (Option 1), you would enter the line:

```
BC default ts=5,10,15,rm=70,bk=1←
```

This statement begins with **BC** (Blank the Command Line) like other commands in the startup file do.

ACTION *(Option 3)*

Entering Defaults from the Command Line.
This method of entering default settings from the Command Line is appropriate when you want to change a default temporarily, only for the current editing session.

To set the tabs at 5, 10 and 15, right margin at 70, and backup of files turned on:

Type: F5 default ts=5,10,15,rm=70,bk=1 ↵

Result: These changes take effect immediately — whenever you create a new document or call an existing one, until you quit XyWrite or change the defaults.

ALSO SEE

Resetting the Defaults. After changing the default settings, if you wish, you can return to the original settings built into XyWrite with the DM (Default Margins) command.

NOTE #1

Priority of Default Settings. Commands embedded in text take priority over all other settings — over those entered in the STARTUP.INT file, on the Command Line, or in the Printer File. In other words, RM 80 embedded into the document takes precedence over a default setting of RM=70 in the STARTUP.INT file, Printer File or on the Command Line.

NOTE #2

Use of Equal Sign. Notice the use of the equal sign (=) — for example, DEFAULT LM=5. This is in contrast to the embedded format commands which do not use an equal sign: LM 5.

FORMAT **CM DM** Restore Defaults

DM is an immediate command.

PURPOSE After changing the default settings, you can return to the XyWrite built-in settings with the **DM** (Default Margins) command. This command sets the following initial conditions:

Format Settings

AL=0	Automatic Leading
BF=0	Bottom Footnotes
TS=8,16,24,...	Tab Set
FL	Flush Left
HY=1	Hyphenation
IP=0,0	Indent Paragraph
LM=0	Left Margin
LS=1	Line Space
NJ	No Justification
OF=0,0	Offset
OP=2	Orphan
PL=55,0,0	Page Length
RM=78	Right Margin
SF=1	Set Footnote No.
TP=0	Top Margin
WD=2	Widow

Printer and Display Settings

EJ=0	Eject Page
FF=0	Form Feed
TB=0	Tab Character

System Settings

BK=1	Backup Files

ACTION

Restoring the Defaults

As with the DEFAULT command, you can enter DM at the Command Line, in the Printer File, or in the STARTUP.INT file. In all three cases, you enter it as DM. The change affects all existing and newly created files, and does not supercede commands embedded in the file.

Value of Variable VA

FORMAT **CM VA** *nm* Value of Variable

nm is a variable (see the table earlier in this section)
VA is an embedded command.

PURPOSE **VA** obtains the current value (or string) of the variable you request (*nm*) and inserts it into the text. For example, VA LM obtains the current Left Margin setting.

The list of variables you can use with VA is given at the beginning of this section in the table "Default and VA Settings."

You can also use the VA command in User Programs (see Advanced User Programming, Chapter 5.)

ACTION **Obtaining a Value.**
To get the current value of a XyWrite setting, say tabs (TS), enter the VA command for that setting (VA TS):

1. Move the cursor to the point in text where you want to find the current tab settings.

2. Enter the VA command on the Command Line. For example, to find out the tab settings:

   Type: F5 va ts ↵

Result: The display will show the tabs settings at that location in the text. For example:

▲8,16,24,32,40,48,56

Help Files

INTRO

You might not expect Help Files to be included in a chapter on customizing. XyWrite allows you to create your own help files from scratch, or modify the ones we have provided.

CONTENTS

Displaying Help

Alt F9

FORMAT Alt F9 Display Help

This is an immediate command.

PURPOSE The purpose of a Help File, of course, is to provide help. Help Files remind you of the XyWrite commands, and put useful information at your finger tips. You can modify the Help Files, if you wish — see "Modifying Help Files" later in this section.

ACTION **Accessing the Help File.**
To use the Help File:

1. Press: Alt F9

   Result: This displays the first level of help. You may get either a single row of topics (on the ruler line) or a full screen. (If you get an error message instead of a help screen, see Note #1.)

2. Use the cursor keys to select a topic, then press ↵, or strike the first letter of a menu word.

3. At any point within Help you can press either: Esc to go back to the previous level of help or Alt F9 to return to your work.

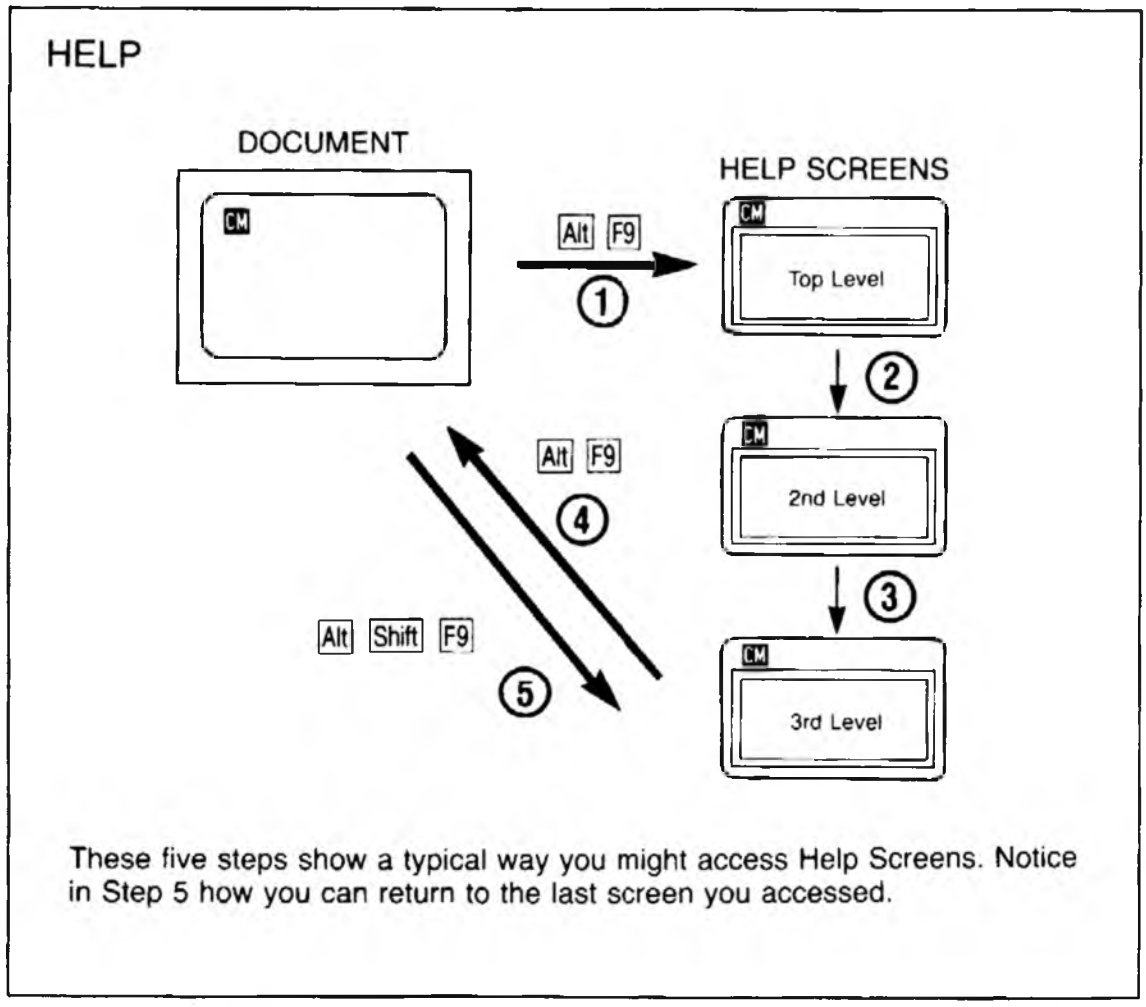
Stepping backwards through help screens is described in Note #2.

NOTE #1 **Error Message.** If you get an error message when you press Alt F9, either (1) the Help File has not been loaded into memory, or (2) the correct path name was left off the LDHELP command. To remedy this, perform the procedure "Loading the Help File" which follows. Entering the path name along with LDHELP allows the Help file to be accessed from any directory.

NOTE #2 **Returning to the Last Help Screen.** If you exited the Help File from several levels deep with Alt F9, you can return directly to that point by using Alt Shift F9.

NOTE #3 **Exiting with** Esc. At any level of Help, pressing Esc moves you back to the previous level of Help. When you are at the very first level of the Help, pressing Esc exits Help.

NOTE #4 **HELP Command.** You can enter the word HELP or just a question mark (?) on the Command Line and XyWrite displays the screen that appears when the program is first entered. This screen shows the serial number, version number and XyQuest address and phone number for assistance. It also reminds you to press Alt F9 to access the Help screens.



These five steps show a typical way you might access Help Screens. Notice in Step 5 how you can return to the last screen you accessed.

Loading a Help File — LDHELP

FORMAT

CM **LDHELP** *d:filename*

d: is the drive where the help file is located.
filename is LONG.HLP, SHORT.HLP, or a custom name
LDHELP is an immediate command.

PURPOSE

The **LDHELP** (Load Help) command loads a Help index into memory, giving you access to the entire Help File. Normally you include LDHELP in your STARTUP.INT file.

You must use LDHELP before you can access help with Alt F9. Notice that to save memory, only a small part of the Help File loads into memory — not the entire Help File. When you press Alt F9, XyWrite accesses the disk for the help information. Thus, you must keep the Help File on the disk, though you can keep it on any drive (and in any directory) as long as you specify both the drive and directory when you issue LDHELP.

XyWrite comes with two Help Files:

LONG.HLP
SHORT.HLP

LONG.HLP is a rather extensive on-line reference about most of the features of XyWrite. It is intended to help those who are just learning XyWrite.

SHORT.HLP is an abbreviated version of LONG.HLP. It is designed as an everyday help for those who know XyWrite and don't need the larger help file on their working disk.

ACTION **Loading a Help File.**
To load a Help File:

1. Enter LDHELP along with the file you want to access — LONG.HLP or SHORT.HLP. *Be sure to include the drive and path name* for the file — this allows the Help File to be accessed even if you change directories or drives. For example, if LONG.HLP is located in the directory XYW off the root directory on drive C:
2. Type: F5 ldhelp c:\xyw\long.hlp ↵

NOTE #1 **Loading Help on Startup.** To automatically load the Help File upon entering XyWrite, include the LDHELP command (with drive and path name) in the STARTUP.INT file. (See STARTUP.INT later in this chapter for the procedure.)

NOTE #2 **PATH Command in DOS.** If you specify the path to the Help File in the DOS PATH command, you need not specify the path in Step 2 above.

Modifying Help Files

INTRO

Modifying Help Files. You can modify the LONG.HLP and SHORT.HLP files, or construct your own. These files are constructed of help frames — each frame can be up to a screenful of text. The best way to learn how to modify a Help File is to call up that file and study it. The rules that control the frames are:

1. Each frame starts with a line of text enclosed in double curly braces. This line defines two things: (1) the first character defines the *type* of help frame and (2) the following words are *keywords*. For example:

   {{#*keyword1*,*keyword2*, . . . }} Format
   {{6margins,top margin,bottom margin}} Example

   This example defines a Type 6 frame, with three keywords: "margins," "top margin" and "bottom margin." You can access this help frame by *any* of the keywords — they are all *synonyms*. In this example, you could call this help frame by the name "margins," "top margin," or "bottom margin." Separate the keywords with commas.

2. The frame can be any of 12 types — Type 0 to Type 9, Type A or Type B. Some types display help only on the Ruler Line, while others display full-screen. For details, refer to "Types of Help Frames" which follows.

3. The *keywords* link frames together. They allow you to move directly from one help frame (say, frame X) to another (say, frame Y). You set this up by including a keyword of frame Y in **bold** somewhere in the text of frame X. When you press Alt F9 to view help frame X, you can move the cursor onto the bold keyword, and press ↵ (Enter) to view help frame Y.

4. If there is more than one bold reference in a frame, you can use the cursor keys to move from one to another. As each reference is selected, it appears highlighted.

NOTE #1 **Wildcard Keywords.** You can use the asterisk (*) as a wildcard in a keyword. It means that *any* characters can occupy the remaining positions. For example:

```
{{2char*}}
```

This frame would be found by any of the following:

character
characters
character string

NOTE #2 **Keyword Length.** There is no limit to the length of a keyword. Shorter keywords are preferable, but not required.

NOTE #3 **Special Keywords.** The dollar sign ($) indicates keywords that control when a specific help frame can appear. They are defined as follows:

$N Call this frame when *no file* is open on the screen
$F Call this frame when a *file* is open on the screen
$D Call this frame when a *directory* is on the screen
$B Call this frame when a *block is defined* in the current file
$* Call this frame for any of the above
$0 Call this frame directly from a key (see Note #4)
— similarly for $1 through $9 and $A through $Z

Call up the LONG.HLP or SHORT.HLP file and find these special keywords to see how to use them.

NOTE #4 **Direct Access to a Help Frame from a Key.** You can access frames with keywords $0 through $9 and $A through $Z directly from the keyboard by assigning the corresponding Function Calls in the Keyboard File. (See the Keyboard File later in this chapter.)

NOTE #5 **Ordering the Frames.** The frames do not have to be listed in any particular order within the Help File. However, if there are two frames with the same keyword, the first occurrence is selected.

NOTE #6 **Missing Keywords and References.** You can have keywords that are not referenced and you can have references for which there is no keyword. In either case, when the user selects a keyword for which no frame is defined, a wildcard frame is called, such as {{8**}}.

HELP FRAMES

Types of Help Frames. Following are descriptions of the twelve types of frames available in a help file. For examples of the different types of frames, call up a Help File and study its frames.

All twelve types of frames start with: {{#,*keyword*}}. The comma is optional. The *keyword* gives the help frame a name so that it can be accessed by that name. You can include several *keywords* so that the frame can be accessed by different names. (For example, you might name the SAVE menu both SA and SAVE.)

TYPE 0 FRAME

Ruler Line Help Menu.
A Type 0 help frame displays a menu of keywords on the Ruler Line (the line below the Prompt Line). The user selects a menu item with the cursor, then presses ↵ (Enter) to open up that help frame.

Constructing the Frame: A Type 0 help frame can be any number of lines. The first line contains the number 0 and keywords enclosed in curly braces. Each subsequent line contains a keyword that is displayed on the Ruler Line when this help frame is used, followed by a comma and and text (usually a description of the keyword) which appears on the Prompt line whenever the keyword is selected.

TYPE 1 FRAME

General One-Line Help.
A Type 1 help frame can be used in three ways: to enter a command on the Command line; to insert text in a file; to execute programs.

Constructing the Frame: A Type 1 help frame is three lines. The first line contains the number 1 and keywords enclosed in curly braces. The second line contains the message that appears on the Ruler line. This message can include underlined (MDUL) areas where the user enters text. The third line can be defined in three ways, as follows. In each variation, you can incorporate the text from the underlined areas on line two by using variable %1 to denote the first underlined area, %2 to denote the second underlined area, and so on. You can define the third line as:

```
{{1,OFFSET}}
Offset for odd pages __ for even pages __
OF %1,%2
```

- Text that is entered directly into the displayed file at the current cursor location. To construct this type of frame, you must start line 3 with an ASCII 1 (☺). For example:

```
{{1AUTHOR}}
Enter the name of the author _____
☺Written by: %1
```

- A program that is executed from the cursor location of the displayed file. To construct this type of frame, you must start line 3 with an ASCII 2 (☻). For example:

```
{{1STYLE}}
Enter the name of your style _____
☻BC US %1 XC
```


TYPE 2 FRAME

One-Line Help Bar for ASCII Characters.

A Type 2 help frame displays a string of characters on the Ruler Line and the ASCII value of the selected character on the Prompt Line. Selecting a character with the cursor puts the ASCII number for that character on the Prompt Line. Pressing ↵ (Enter) inserts this character into the text wherever the cursor was when the help was selected.

Constructing the Frame: A Type 2 help frame is two lines. The first line contains the number 2 and keywords enclosed in curly braces. The second line consists of a string of characters with no spaces between them. Each character is interpreted as a menu item on the Ruler Line.

TYPE 3 FRAME

One-Line Help Bar for Keywords.

A Type 3 help frame displays one line of text on the Ruler Line. This text can have an underlined area where the user can type. Pressing ↵ calls the help frame that matches the string of text entered.

Constructing the Frame: A Type 3 help frame is two lines. The first line contains the number 3 and keywords enclosed in curly braces. The second line is the text you want displayed on the Ruler Line. An underlined area on this line permits text entry.

TYPE 4 FRAME

Help for Key Assignments.

A Type 4 help frame displays a message on the Ruler Line. When the user presses a *letter* key, the character is displayed on the Prompt Line. If the user presses a key that has *function calls* assigned to it, those Function Calls are displayed (such as CR for Cursor Right). If there is a match between the Function Call and any Type 8 keyword, then that frame is displayed for a minimum of two seconds. To exit this frame, press Esc.

Constructing the Frame: A Type 4 help frame is two lines. The first line contains the number 4 and keywords enclosed in curly braces. The second line of the frame is the message to be displayed on the Ruler Line. When you want to call a full-screen frame from a Type 4 frame, call a Type 8 frame (rather than a Type 6 frame).

TYPE 5 FRAME

General Full-Screen Help.

A Type 5 help frame can be used to enter text into the displayed file or to execute a program. It can contain underlined areas for the user to enter text. When the frame is displayed, the user can move from one underline to the next with the cursor keys.

Constructing the Frame: A Type 5 help frame can be any number of lines. The first line contains the number 5 and keywords enclosed in curly braces. The next lines contain the text, with underlined fill-in areas that you want to display on the screen. At the end of the text to be displayed, insert an ASCII 1 (☺) character or an ASCII 2 (☻) character on a line by itself. An ASCII 1 character tells XyWrite that the lines that follow are to be inserted as text into a displayed file. An ASCII 2 character tells XyWrite that the lines that follow are programs to be executed. Use %1, %2, %3, and so on, to represent the values from the first underline area, second underline area, third underline area, and so on.

TYPE 6 FRAME

General Full-Screen Help.

A Type 6 help frame displays general help not specific to any particular command or Function Call. The user uses the cursor keys to select any word or phrase in bold (the word under the cursor appears in reverse mode) and presses ↵ to select further help. Call a Type 6 frame from a Type 0 frame.

Constructing the Frame: A Type 6 help frame can be any number of lines. The first line contains the number 6 and keywords enclosed in curly braces. All following lines display on the screen. In the text, make keywords bold.

TYPE 7 FRAME

One-Line Help Using a Directory.

A Type 7 frame allows you to execute commands with a directory on-screen. You might use commands such as CALL, ERASE, LDHELP and CHDIR — these are all commands that you can execute while pointing to a filename with the cursor.

Constructing the Frame: A Type 7 can be any number of lines. The first line contains the number 7 and keywords enclosed in curly braces. Each subsequent line starts with the word you want to appear on the Ruler line, followed by a comma, the command you want to execute, and then a space. After the space, you can include a description of the command. All the text following the comma will appear on the Prompt line.

If the first character after the comma is a hyphen (-), Xywrite will call up the help frame with that name to select more commands. For example:

```
{{7Directory}}
Call,Call a file for editing
Help,LDHELP (Load the Help File)
More,-More commands

{{7-More}}
Printer,LDPRN (Load Printer File)
```

TYPE 8 FRAME

Full-Screen Help for Function Calls.

A Type 8 help frame displays full-screen help particular to Function Calls (as opposed to commands). Otherwise, it is constructed and operates the same as a Type 6 frame. Call a Type 8 frame from a Type 4 or Type 6 frame.

Constructing the Frame: A Type 8 help frame can be any number of lines. The first line contains the number 8 and keywords enclosed in curly braces. All following lines display on the screen. In the text, make keywords bold.

TYPE 9 FRAME

Help for Error Messages.

A Type 9 help frame calls up a full-screen help frame for the last error number that the user received. For example, you might have the keyword "Error" in bold in a general help screen (Type 0 or 6), and use the same word "Error" as the keyword in your Type 9 frame. Then, when the user selects the word "Error" from the general help screen, the help frame corresponding to the last error number displays. (Strictly speaking, there is no Type 9 frame that displays — it is a dummy frame that simply holds a name for error message frames.)

Constructing the Frame: A Type 9 help frame is only one line, which contains the number 9 and keywords enclosed in curly braces. (If you set one of the keywords to $E, you can access this feature with one keystroke.) The message you want to appear associated with an error should be entered as a Type 6 frame, with internal XyWrite error numbers as keywords — for example, the Type 6 message frame for errors 12, 17 and 133 would be defined by {{612,17,133}}. Error numbers must be entered without leading zeros. You can look in the LONG.HLP file for the current error assignments. All error numbers are not included, as they are subject to change.

TYPE A FRAME

Keycode Help for Key assignments.

This frame is similar to a Type 4 frame. When you access a Type A frame, a message appears on the Ruler Line. Then when the user presses a key, XyWrite calls up a help frame with that keycode as its name. For example, if you press the F1 key (which is key #59), the help frame named "#59" is called up. The user presses ↵ to exit from this type of help frame.

Constructing the Frame: A Type A help frame is two lines. The first line contains the letter A and keywords enclosed in curly braces. The second line contains the message to be displayed on the Ruler Line.

TYPE B FRAME

Executing Commands by Number.
When you call a Type B frame, a list of numbered items (a "menu") is displayed. The user then presses a number (1 through 9) and the command associated with that number is executed.

Constructing the Frame: A Type B help frame can be up to 19 lines long. The first line contains the letter B and keywords enclosed in curly braces. What follows is up to nine *text* lines (numbered), which are displayed on the screen. The text lines are followed by the same number of *command* lines. Each command line must start with an ASCII 1 (☺) (XyWrite enters the line as text in the displayed file) or an ASCII 2 (☻) (XyWrite executes the line as a program).

NOTE #7

Accessing a Help Frame from the Command Line.
You can assign the function call HL (Help) to a key (via the Keyboard File) to pick up the first word on the Command Line and call up the help frame with that name.

NOTE #8

Displaying Help Automatically on Errors. By specifying ER=1 in the Printer File, a full-screen help frame is displayed whenever an error occurs (a different frame appears for each error). Because this feature can slow performance, it should only be set when a new user is being trained. (You can also set ER with the DEFAULT command.)

NOTE #9

Checking Help Fields. You can include the HF embedded command in Type 5 help frames to compare the original values in an underlined area (a help field) with the current values. If the values have not changed, the line that follows is ignored. The format of the command is:

CM **HF** *n1,n2,n3*

where *n1*, *n2*, and *n3* are the help field numbers. You must end this line with an EI (End If) command.

NOTES

Automatic Hyphenation

INTRO

Automatic hyphenation is really very easy to implement. In Chapter 4 we described how to use the HY command. In this section we describe the LDDICT command and the hyphen exception dictionary DICTION.

CONTENTS

PURPOSE

XyWrite has an automatic hyphen capability that breaks words according to an internally defined set of rules. Since English has exceptions to hyphenation rules, an *exception* dictionary, DICTION, is provided to let you customize the automatic hyphenation process.

You add words to DICTION and force them to break wherever you like — or keep them from breaking at all. Since it is an exception dictionary, it does not need to have all of the words of the English language in it.

To switch on automatic hyphenation, you need to use the LDDICT command. This command loads the hyphen dictionary into memory. The HY embedded command then enables you to switch hyphenation off (and on) for any documents or parts of documents. (HY is described in the Alignment section of Chapter 4.) You can also modify the rules which specify how words are broken with the HY printer setting found in the Printer File (which is distinct from the HY format command).

LDDICT Loading the Hyphen Dictionary

FORMAT **CM LDDICT** *filename*

filename is the name of a hyphen exception dictionary
LDDICT is an immediate command.

PURPOSE The **LDDICT** (Load Dictionary) command loads the hyphen exception word dictionary. This turns on the automatic hyphenation (since the default of the HY command is HY ON).

Automatic hyphenation works as follows: At the end of each line, XyWrite tests the last word as you type it in, to determine if the word should be hyphenated. It first looks in the hyphen exception dictionary to see if the word is present — if not, it uses its internal rules to break the word.

The file that you load with LDDICT contains the *exceptions* to the hyphenation rules. This is a standard text file that you can edit. In this file you can control how you want words to break or not break. Refer to the "Rules for the Dictionary" listed later.

ACTION **Loading the Hyphen Dictionary.**
To load the hyphen dictionary:

Enter the LDDICT command along with the filename of the hyphen exception dictionary — let's use DICTION:

Type: F5 `lddict diction` ↵

Result: Automatic hyphenation is now turned on. The file DICTION is loaded into memory. If you normally use automatic hyphenation, you should include this command in your STARTUP.INT file.

NOTE #1

Rules for the Dictionary. When adding a word to DICTION, use the following rules:

1. If you want to prevent the word from breaking, type the word with no hyphens.

2. Include hyphens where you want the word to break.

3. If your word is 7 characters or longer, you might want to include an asterisk (*) at (or near) the end of the word. (See Note #2.)

4. You don't have to include every form of a long word you want checked. One form, with the asterisk in the right place, might suffice for forms that end differently. (See Note #3).

ACTION

Adding a Non-Breaking Word to DICTION.
The standard hyphen dictionary file that comes with XyWrite is called DICTION. To add a word to this file:

1. Type: F5 `call diction` ↵
2. Let's add the word "XyQuest" so that it does not break. Since this would fall near the end of the file, move to the end:

   Press: Ctrl End

3. Enter the word "XyQuest" with no hyphen: (Notice the word does *not* need to be capitalized.)

   ```
   wor-ship
   wor-thy
   xyquest
   xy-write
   yard-age
   ```

4. Type: F5 `store` ↵
5. Type: F5 `lddict diction` ↵

Result: The word "XyQuest" has been added to your hyphen dictionary and will no longer hyphenate at the end of a line.

ACTION **Adding a Breaking Word to DICTION.**
The word "parapsychology" breaks badly by the internal rules (parap-sychology). To add this word to the dictionary, we call the file DICTION as before and move the cursor to just after the word "palate":

1. Enter the word, including hyphens and an asterisk (see Note #2):

   Type: para-psy-chol-ogy*

   Any hyphenation points after the 7th letter of a word are overlooked unless you put in an asterisk.

2. Type: [F5]store[↵]
3. Type: [F5]lddict diction[↵]

NOTE #2 **The Asterisk.** When an asterisk (*) is *not* present in a word, LDDICT loads only the first 7 characters of that word into its hyphen dictionary in memory — only these characters are compared to the text. To load more than 7, insert an asterisk after all the letters you want included. Only the letters *ahead* of the asterisk are loaded into memory.

NOTE #3 **Saving Dictionary Space.** The dictionary memory buffer is 64K; to save space, you can place the asterisk so one root word represents several forms. This is because the part of a word after the first 7 letters (or after the asterisk) is handled by the same internal rules that hyphenate most words.

Take, for example, the root word "approximate." The dictionary contains "ap-prox-i-m*ate". Here's how other forms will be broken:

approximately	is treated as	ap-prox-i-mate-ly
approximation	is treated as	ap-prox-i-ma-tion
approximating	is treated as	ap-prox-i-mat-ing

Before you enter a word and place the asterisk in it, jot down all the alternate forms you want covered. Then put the asterisk where it correctly handles all or most of them. Save and load the dictionary and try each form. If one doesn't break correctly, switch back to the dictionary and experiment with the placement of the asterisk, or list the errant form separately.

NOTE #4 **Ordering the Words.** You don't need to order the words alphabetically in the dictionary . However, we recommend that you insert the words alphabetically, to help you locate words easier.

NOTE #5 **Loading Dictionary on Startup.** To automatically load the dictionary on entering XyWrite, enter the LDDICT command into the STARTUP.INT file. (See STARTUP.INT later in this chapter for this procedure.)

NOTE #6 **Turning Off Hyphenation.** You can turn off hyphenation three ways:

1. Give the LDDICT command with *no filename*. This clears the dictionary from memory and turns hyphenation *off*.
2. Use the HY OFF embedded format command within a document to turn it off (and HY ON to turn it on).
3. On the Command Line, type DEFAULT HY=0. (This is an immediate command.)

ALSO SEE **Related Command.** Refer to the Printer File for the HV setting. This setting controls three hyphenation settings:

- The shortest word allowed to break.
- The fewest letters allowed *before* the hyphen.
- The fewest letters allowed *after* the hyphen.

SHOHYP Show Hyphenation

FORMAT

CM **SHOHYP** *filename* Show Hyphenation

filename is the name of the file you want to check
SHOHYP is an immediate command.

PURPOSE

The **SHOHYP** (Show Hyphenation) command enables you to see all of the hyphenation points in a list of words. It does this by creating a new file (HY.TMP) which lists the words with all hyphens showing.

You would use SHOHYP, for instance, if you were a lawyer and wanted to check the hyphenation on a set of words peculiar to your profession. You would type up a list of these words, store the file, and execute SHOHYP on that file. You would then look over the hyphenated words—if any were exceptions to the rules and did not break properly, you could single them out and add them to the DICTION file along with the proper hyphenation.

The file you check should not contain embedded commands—thus, SHOHYP is not a command you would run on just any file. (See the note below.)

ACTION

Viewing Your File's Hyphenation Points.
To view all of the hyphenation points in a list of words:

1. Create a file with the list of words whose hyphenation you want to check. Be sure there are no embedded commands in the file.
2. Store (or save) this file.
3. Type SHOHYP along with the name of this new file:

   Type: F5 `shohyp list` ↵

Result: XyWrite creates a file HY.TMP containing the words as they are hyphenated by XyWrite. Short words are omitted. Call up HY.TMP and view this file. If any words are improperly hyphenated, correct them and add them to the hyphenation exception file DICTION.

NOTE

Omitting Embedded Commands. You should omit any embedded commands from the file you are checking — they are likely to cause problems. This includes character modes (such as «MDBO») and format commands («RM50»).

NOTES

Keyboard File

INTRO

The keyboard is your connection to XyWrite — you can modify the keyboard functions to your liking. You can re-arrange keys, and you can reduce the most complicated function to a single keystroke. You can make the keyboard *help* you rather than fight you.

This section also covers the Function Calls, which form the foundation of XyWrite. Once you become familiar with the Keyboard File, you may find yourself inventing new combinations of functions to best serve your purpose.

PURPOSE

Did you ever wish you could change the meaning of the keys on the keyboard? In this section you will learn how to change a key or even the whole keyboard to suit your own needs.

Here are some simple ways you can change the Keyboard File:

- **Add convenience.** For example, move the troublesome left shift key (on the standard IBM keyboard) to a more convenient key. You might swap it with the backslash key.
- **Create shortcuts.** For example, assign the key combination Ctrl S to be a shortcut for F5 store ↵, to store a file.
- **Assign new functions to keys.** You can alter the Keyboard File to do things you could otherwise not do. You can assign any of the Function Calls you prefer to a key. For example, you can change the cursor up and down keys to Linear Up (LU) and Down (LD). Linear Up and Down allow the cursor to move directly up or down without ever switching columns.

You generally assign basic functions to the Keyboard File. From these you can construct more elaborate routines in Program Files.

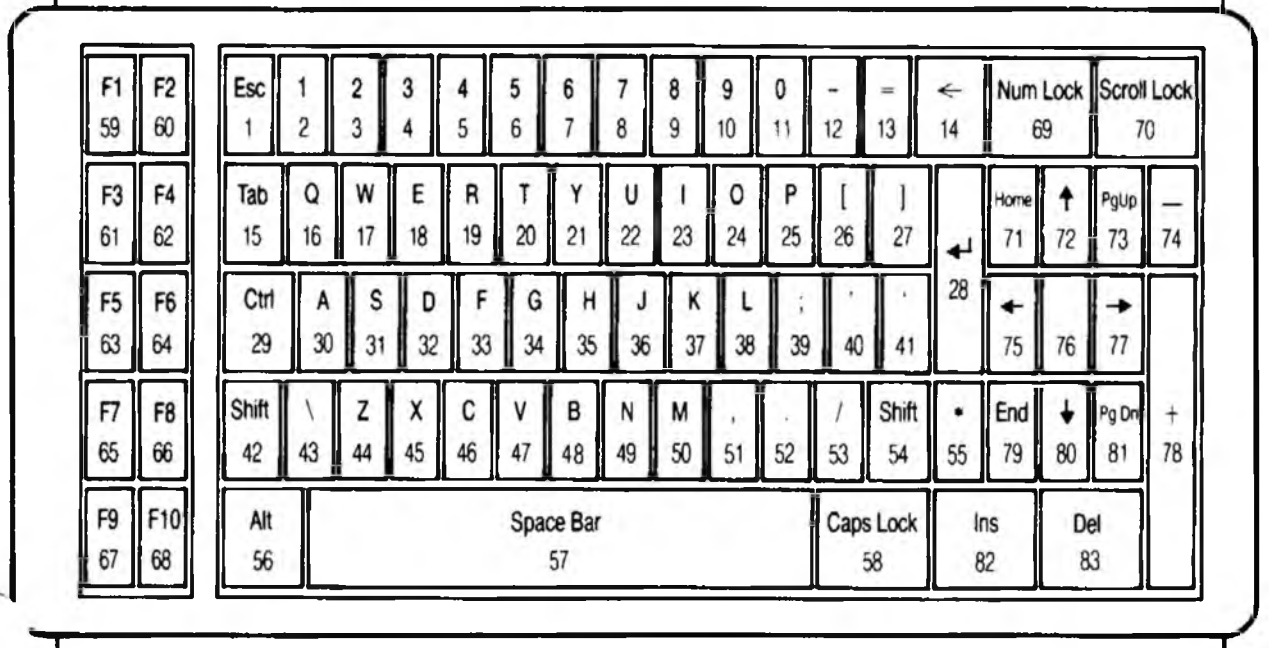
Before you learn key assignment, you should know how XyWrite takes keyboard input and puts the desired character on the screen. This is done with key codes in a keyboard file called IBM.KBD contained on the original XyWrite disk.

The figure on the facing page shows how each key is numbered. For example, when you strike key number 30, the Keyboard File translates code 30 to mean the letter 'A'.

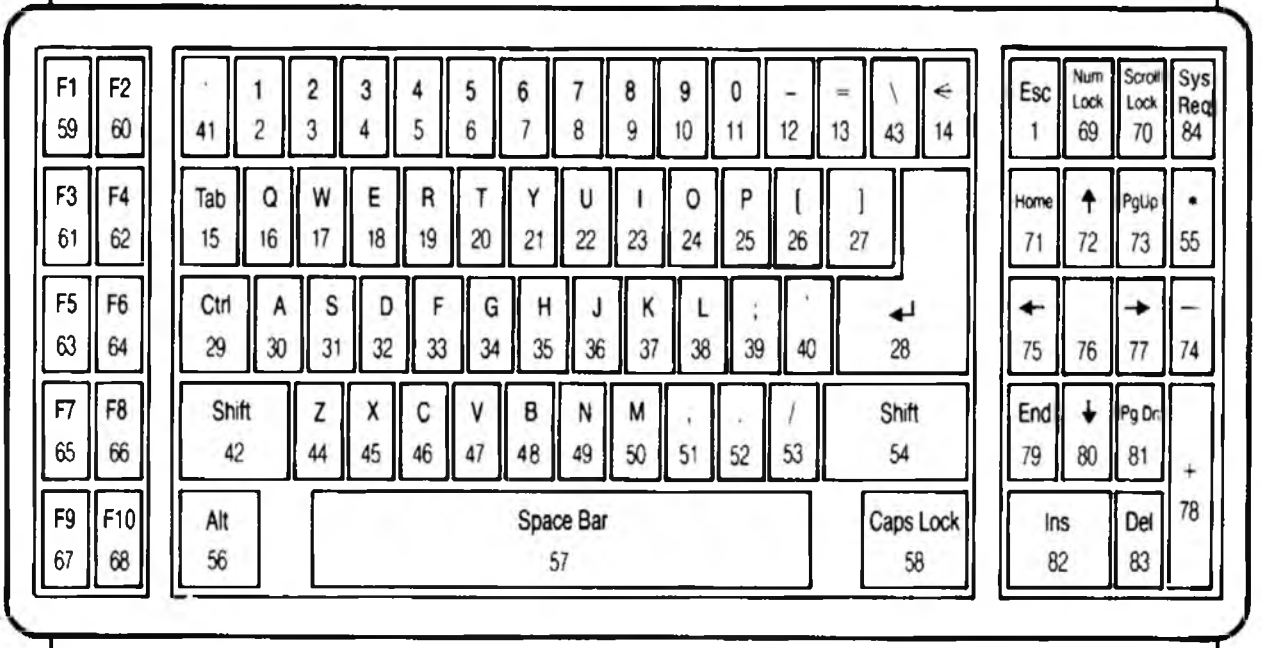
You use the LDKBD (Load Keyboard) command to load a new keyboard meaning into XyWrite. It is a feature of XyWrite that you can load a Keyboard File any time that XyWrite is running. This command is described near the end of this section.

KEYBOARD DIAGRAM

STANDARD IBM KEYBOARD



IBM PC/AT KEYBOARD



How to Modify Your Keyboard File. This section describes how a Keyboard File is constructed. The IBM.KBD file is organized into five part:

1.	Comments	Example: ;END OF TABLE
2.	Number of Keys	Example: KEYS=84
3.	Shifting Keys	Example: CTRL=29
4.	Tables	Example: TABLE=CTRL
5.	Key definitions	Example: 30=a

Each of these parts requires some explanation:

1. **Comments.** Every line that begins with a semicolon is a comment — it has no effect on key assignments.

2. **Number of Keys.** The first definition in the Keyboard File is the KEYS statement. There are 83 keys on a standard IBM keyboard. However, to accommodate the IBM PC/AT keyboard, which has 84 keys, we use the statement:

   ```
   KEYS=84
   ```

3. **Shifting Keys.** The IBM.KBD file comes with five shifting *keys* which define four shifting *states*. (Keys 42 and 54 are both assigned the same state: SHIFT.) Each shifting key shifts the keyboard to a new set of key definitions. The shifting states are defined by:

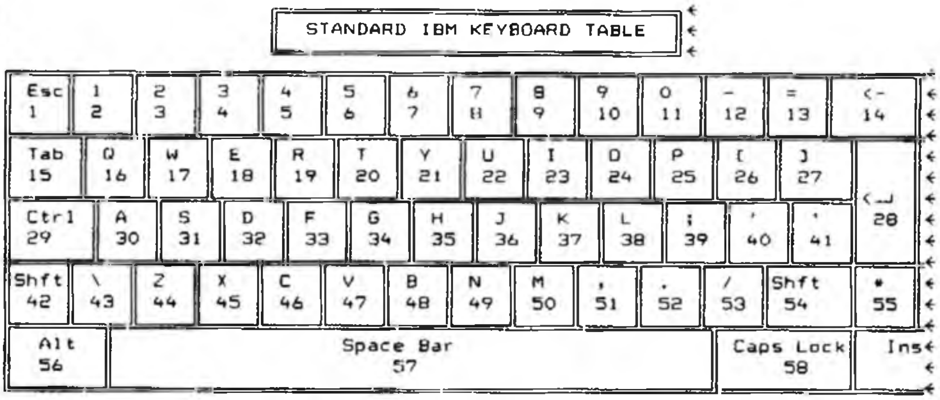
   ```
   CTRL=29
   ALT=56
   SHIFT=42,54,N
   CAPS=58,T:C
   ```

   For example, when you press Alt and the letter A, you get a Save/Get, defined in the Alt table as @A.

   XyWrite can have up to a total of six shifting states (using any names you want — no numbers allowed). That is, if you keep these four states, you can define two more (or you could delete these four and define six new ones). You can use these keys alone (TABLE=CTRL) or in combination (TABLE=CTRL+ALT) to create up to 20 tables. For example, you could define the Escape key Esc to be a shifting key, to change the keyboard to Greek letters.

   The order of these definitions defines their priority. In the above example, if you press both the CTRL and CAPS LOCK keys, the CTRL table prevails.

KEYBOARD FILE — IBM.KBD




```
;<
KEYS=84<
CTRL=29<
ALT=56<
SHIFT=42,54,N<
CAPS=58,T:C<
;<
TABLE=<
1=<<
2=1<
3=2<
4=3<
.......................................................
Y<
TABLE=CTRL<
1=»<
2=m1<
3=m2<
4=m3<
.......................................................
TABLE=CTRL+ALT,SHIFT+ALT,CTRL+ALT+SHIFT,CTRL+SHIFT<
1=<<
2=1<
3=2<
4=3<
.......................................................
Y<
TABLE=SHIFT<
1=«<
2=!<
3=@<
4=#<
.......................................................
Y<
TABLE=CAPS<
1=<<
2=1<
3=2<
.......................................................
Y<
TABLE=SHIFT+CAPS<
1=«<
2=!<
3=@<
4=#<
.......................................................
TABLE=ALT<
1=<<
2=@1<
3=@2<
4=@3<
.......................................................
83=RW<
84=<
;<
;END OF KEYBOARD TABLE<
```

(2) Number of Keys

(3) Shifting Keys

(4) Tables

(5) Key Definitions

Lines beginning with semi-colons (;) are comments.

Options for shifting key definitions.

,N	Numeric Lock
,T:*n*	Toggle
,S:*n*	Single-Shot

n is any letter you specify. It is displayed in the upper right-hand corner of the screen. It indicates that the Toggle Key is *on*, or that the Single-Shot key is armed.

N (Numeric Lock) indicates that when you press that key, the sense of the Numeric Lock key changes. In the example SHIFT=42,54,N statement, if Numeric Lock is on, pressing either Shift key changes the number pad back to a cursor pad.

T (Toggle) defines a key as a toggle shifting key — that is, it switches *on* the first time you press it and *off* the next time. You can define up to four toggle keys — CAPS LOCK and three others (Num Lock, Scroll Lock and Automatic Uppercase do not count — they are not toggle *shifting* keys). You can display any letter in the header when a toggle key is on. For example, CAPS=58,T:C causes the letter 'C' to appear when you press the Caps Lock key. The 'C' disappears when you press the Caps Lock key again.

S (Single-Shot) specifies that the shifting key does not have to be held down when pressing the next key. For example, if you define CTRL=29,S:A you could execute Ctrl Z by pressing the Ctrl key, releasing it, then pressing the letter Z. The letter 'A' would appear in the header when you press Ctrl, and would turn off when you press the Z. Single-Shot shifting is especially useful for people with a typing handicap.

4. **Tables**. The Keyboard File IBM.KBD is made up of seven tables. (You can create up to 20 tables.) Each table is like having an entire new keyboard. The seven tables are:

```
TABLE=
TABLE=CTRL
TABLE=CTRL+ALT,SHIFT+ALT,CTRL+SHIFT,CTRL+ALT+SHIFT
TABLE=SHIFT
TABLE=CAPS
TABLE=SHIFT+CAPS
TABLE=ALT
```

Each table begins with a TABLE= statement. Following the TABLE=CTRL statement, for example, you will find the table which applies when the CTRL key is pressed.

5. **Key Definitions**. Following the TABLE= statement you will find the key definitions.

Format: *#=f1,f2,f3,...*

is the keycode — a number from 1 to 83
f1,f2,f3,... are Function Calls or characters

Examples:

`30=a`	The letter 'a' is assigned to key number 30.
`72=CU`	The function CU (Cursor Up) is assigned to key 72.
`80=MU,LD`	The two functions MU, LD (Move Up, Linear Down) are assigned to key 80.
`31=BC,s,a,v,e,XC`	The command SAVE is executed by the series of keystrokes BC (Blank Command Line), spelling SAVE, and XC (Execute)

Notice in this last example that when you enter text, each character is separated by a comma.

ACTION

Re-Assigning a Key

To demonstrate how to re-assign a key, we will show how to change the Ctrl-Up Arrow key combination to move the cursor up five lines at a time.

1. Call the Keyboard File to the screen:

   Type: F5 `ca ibm.kbd` ↵

2. Use the SEARCH command to find the table for the Ctrl key.

   Type: F5 `se /TABLE=CTRL/` F9

   You may have to continue the search with F9 two or three times until you find a list of keycodes that begins with this statement.

3. Referring to the illustration, you can see Cursor Up is 72.

   Type: F5 `se /72/` ↵

   Result: The cursor stops on the line reading:

   `72=CU←`

   CU is a *function call* —it means Cursor Up. (Each time it is executed, it moves the cursor up one row.)

4. Change this line to read:

   `72=CU,CU,CU,CU,CU←`

5. Store the Keyboard File.

   Type: F5 `store` ↵

6. Load the altered Keyboard File into memory.

   Type: F5 `ldkbd ibm.kbd` ↵

Now, while holding down Ctrl, hit the Up-Arrow key. The cursor moves up five lines. With a similar procedure, it is easy to create a Ctrl-Down Arrow function which will move the cursor down five lines.

EXAMPLE

Command Keys. There are many commands you use in XyWrite. You store or save files and probably call the same file many times during a session. Let's see how you can create a command key so you can save a file by pressing Ctrl S.

1. Type: F5 `ca ibm.kbd` ↵
2. Note that the code for S is 31. Search for the CTRL table.

   Type: F5 `se /table=ctrl/` F9 F9 F9

   Result: The cursor should now be positioned at the top of the CTRL table.
3. Find code 31:

   Type: F5 `se /31/` ↵
4. Change the line to read:

   `31=BC,s,a,v,e,XC` ↵
5. Type: F5 `store` ↵
6. Type: F5 `ldkbd ibm.kbd` ↵

Analysis. Before trying out this new command (in Step 4), let's see just what it does step-by-step:

BC clears the Command Line (the same as key F5). The four letters **SAVE** are typed on the Command Line.
XC executes the SAVE command (the same as key F9).

To try out this new command, call any file to the screen. Then press Ctrl S — this should save the file to disk.

EXAMPLE

Overcoming the IBM Left Shift Key. The IBM PC keyboard presents an interesting challenge for new users who are used to an ordinary typewriter. Since the left Shift key is not next to the Z key, you probably found yourself hitting the backslash key whenever you wanted to use the Left Shift. Wouldn't it be easier if the backslash key could be switched with the left Shift key?

ACTION

Moving the Troublesome Left Shift Key.
Here's how you can reverse the codes for these two keys so you can use the backslash key (43) as the left Shift key (42). Let's swap keys by changing every occurrence of 42 to 43 and every occurrence of 43 to 42.

1. Type: F5 ca ibm.kbd ↵

2. Type: F5 se /42=/ ↵

   Result: The cursor should stop at a "42=" with a "43=" nearby.

3. Change "42" to "43".

4. Change the nearby "43" to "42".

5. To find the next occurrence of code 42:

   Press: F9

6. Repeat Steps 3, 4 and 5 until you receive the NOT FOUND prompt on the PRMPT line.

7. Change the name of the SHIFT function — for example:

   Type: F5 ch /SHIFT/XSHIFT/ ↵

8. Type: F5 store ↵

9. Type: F5 ldkbd ibm.kbd ↵

Try out the new Keyboard File and see if the changes work. Press the old left Shift key — this should now make a backslash (\). Try the old backslash key as a shift key by typing a capital 'A.'

NOTE #1 **Executing Function Calls as Commands.** The FUNC command allows you to execute any of the two-letter Function Calls directly from the Command Line. The Function Call is executed as though it were assigned to a key and you pressed that key. This is a great way to execute seldom-used functions that are not assigned to keys.

To execute a Function Call from the Command Line, enter FUNC followed by the Function Call you want to execute. For instance, to execute the function NM (No Marker):

Type: F5 func nm ↵

In this case, the format triangles and carriage-return arrows normally displayed on-screen are hidden from your view (although they remain embedded in the file).

NOTE #2 Ctrl **vs.** Alt **Key**. The Alt key is used for Save/Get keys A-Z and 0-9. Therefore, when using the Keyboard File to assign new functions to keys A-Z or 0-9, you might use the Ctrl key instead, and leave the Alt key free for Save/Gets.

NOTE #3 **Keep Old Lines as Comments**. When you modify a line in the Keyboard File, it's often a good idea to keep the old line by making it a comment. This way you can remember what the key used to do in case you want to revert it to its previous function. Put a semi-colon (;) in front of the old line. Some people use three semi-colons to emphasize that it was intentionally removed. For example, if you were to change key 72 from CU to LU:

```
;;;72=CU
72=LU
```

NOTE #4 **Multiple Character Assignment.** Whenever you use more than one character in a key assignment, you must start the assignment with a function call. If no other function call is appropriate, use **CS** (Clear Scroll Lock). For example, the assignment 74=J,o,h,n would not work, whereas 74=CS,J,o,h,n would work. (If a key assignment does not begin with a function call, only the first character is read.)

Load Keyboard File — LDKBD

FORMAT

CM **LDKBD** *filename* Load Keyboard File

filename is the Keyboard File to be loaded.
LDKBD is an immediate command.

PURPOSE

The **LDKBD** (Load Keyboard File) command loads the Keyboard File you specify into memory for use. A Keyboard File does not take effect until it is loaded with LDKBD.

ACTION

Loading a Keyboard File
To load a keyboard file into memory:

Enter the LDKBD command along with the name of the Keyboard File you want to load. For example:

Type: F5 `ldkbd ibm1.kbd` ↵

Result: Keyboard File IBM1.KBD is now in effect.

NOTE #1

Requirement. It is not *mandatory* that you load a Keyboard File. XyWrite comes with its own keyboard file built-in. You only have to load a Keyboard File if you wish to *modify* the keyboard from its standard settings.

NOTE #2

Startup. In order to use a Keyboard File you have altered, you must load it each time you run XyWrite. You can have your Keyboard File automatically loaded at startup by adding the LDKBD command to your STARTUP.INT file.

Function Calls

PURPOSE

Function Calls are two-letter instructions that represent basic Keyboard actions. Function Calls are assigned to keys in the Keyboard File. For example, 77=CR means that Cursor Right is assigned to the cursor right-arrow key (key #77). Similarly, DL means define a line of text, CP means copy the defined block of text, and M2 means set bold mode.

There are two ways you can execute these Function Calls: (1) in the Keyboard File, and (2) with the FUNC command on the Command Line. These are both described in the previous section "Keyboard File." Function Calls also appear in Program Files -- for example **BC** appears in the STARTUP.INT file.

To execute several Function Calls in sequence with one keystroke, assign them to a key, separating them with commas. For example, 77=CR,CD would move the cursor right and then down.

NOTE

User Programming. All of these Function Calls can be used in programs except TS (Toggle Scroll Lock). TS cannot be entered into a program file, since you use it to begin and end the recording of a program. The Function Calls are arranged in the following categories:

ASCII Characters
Character Modes
Columns
Command Line
Copying and Moving Text
Counters/Numbering
Cursor Movement in Text Area
Defining a Block of Text
Deleting Text
Foreign/Overstrike Characters
Help
Math
Normal/Expanded Display and PG-LN
Save/Gets
Search
System
Toggle Keys
Windows

FUNCTION CALLS — Listed by Type

The keys listed in the right-hand column are those provided in the original IBM.KBD file. You can change these assignments.

FUNCTION CALL		KEY
ASCII CHARACTERS		
R0	**ASCII 0** – Enter the single digit ASCII 0	Alt Shift 0
R1	**ASCII 1** – Enter the single digit ASCII 1	Alt Shift 1
R2	**ASCII 2** – Enter the single digit ASCII 2	Alt Shift 2
R3	**ASCII 3** – Enter the single digit ASCII 3	Alt Shift 3
R4	**ASCII 4** – Enter the single digit ASCII 4	Alt Shift 4
R5	**ASCII 5** – Enter the single digit ASCII 5	Alt Shift 5
R6	**ASCII 6** – Enter the single digit ASCII 6	Alt Shift 6
R7	**ASCII 7** – Enter the single digit ASCII 7	Alt Shift 7
R8	**ASCII 8** – Enter the single digit ASCII 8	Alt Shift 8
R9	**ASCII 9** – Enter the single digit ASCII 9	Alt Shift 9
CHARACTER MODES		
M0	**Mode 0, Reset** – Text which you type is entered in the mode that exists at the cursor location	Ctrl 0
M1	**Select Normal Mode** – Text which you type is entered in the normal mode (not highlighted)	Ctrl 1
M2	**Select Bold Mode** – Text which you type is entered in bold	Ctrl 2
M3	**Select Underline Mode** – Text which you type is entered in underline	Ctrl 3
M4	**Select Reverse Mode** – Text which you type is entered in reverse mode	Ctrl 4
M5	**Select Bold Underline Mode** – Text which you type is entered in bold underline	Ctrl 5
M6	**Select Bold Reverse Mode** – Text which you type is entered in bold reverse	Ctrl 6
M7	**Select Superscript Mode** – Text which you type is entered in superscript	Ctrl 7
M8	**Select Subscript Mode** – Text which you type is entered in subscript	Ctrl 8

COLUMNS

TL	**Table Column Left** – Move the cursor one column to the left in the column table.	Shift ←
TR	**Table Column Right** – Move the cursor one column to the right in the column table.	Shift →
TE	**Table Entry** – Create a new row of entries in the column table.	Shift Ins
HC	**Home Column** – Move the cursor to the top of the current entry in the column table.	Shift Home
EE	**Erase Entry** – Delete a row of entries in the column table.	Shift Del
EC	**End Column** – Move the cursor to the bottom of the current entry in the column table.	Shift End
ED	**Entire Row Define** – Define the current row of cells in the column table.	(none)
MC	**Mark Cell** – Define the cell where the cursor is located in the column table.	Shift F1

COMMAND LINE

BC	**Blank the Command Line** – Clear the Command Line and move the cursor next to the start of the Command Line.	F5
CC	**Change Cursor** – Move cursor between the Command Line and text.	F10
XC	**Execute** – Execute the command which is currently on the Command Line.	F9
CH	**Clear Header** – Erase the text on the CM line without moving the cursor	(none)
GH	**Go to Header** – Move cursor from the text to the previous cursor position on the Command Line without clearing the Command Line.	(none)
GT	**Go to Text Area** – Move cursor from the Command Line to the previous cursor position in the text area.	(none)

COPYING AND MOVING TEXT

CP	**Copy Defined Block of Text** – Copy the currently defined block of text to the cursor position	F7
MV	**Move Defined Block of Text** – Move the currently defined block of text to the cursor position	F8

COUNTERS/NUMBERING

C0 to **C9**	**COUNTER 0** – Insert the format command C0 in the text.	(none)
	COUNTER 9 – Insert the format command C9 in the text.	(none)

CURSOR MOVEMENT IN TEXT AREA

CR	**Cursor Right** – Move right one character; wrap to next line	→
CL	**Cursor Left** – Move left one space; wrap to previous line	←
CU	**Cursor Up** – Move cursor up one line	↑
CD	**Cursor Down** – Move cursor down one line	↓
LR	**Cursor Linear Right** – Move cursor right one character; allowable to move past the carriage return	(none)
LL	**Cursor Linear Left** – Move cursor left one character; do not wrap to previous line	(none)
LU	**Cursor Linear Up** – Move cursor directly up one line	(none)
LD	**Cursor Linear Down** – Move cursor directly down one line	(none)
PW	**Previous Word** – Move cursor to previous word	Alt ←
NW	**Next Word** – Move cursor to start of next word	Alt →
PT	**Previous Tab** – Move cursor to previous tab position	Shift Tab
NT	**Next Tab** – Move cursor to the next tab without moving the text	Ctrl Tab
EL	**Express Left** – Move cursor to far left of line and then to left end of previous lines	Ctrl ←
ER	**Express Right** – Move cursor to far right of line and then alternately to start and end of subsequent lines	Ctrl →
LB	**Line Begin** – Move cursor to far left of line and no further	(none)
LE	**Line End** – Move cursor to far right of line and no further	(none)
PL	**Previous Line** – Move cursor to start of previous line	(none)
NL	**Next Line** – Move cursor to start of next line	(none)
PS	**Previous Sentence** – Move to start of previous sentence	(none)
NS	**Next Sentence** – Move cursor to start of next sentence	(none)
PP	**Previous Paragraph** – Move to start of previous paragraph	(none)
NP	**Next Paragraph** – Move cursor to start of next paragraph	(none)
HM	**Home** – Move cursor to the top of the screen	Home
BS	**Bottom of Screen** – Move cursor to bottom of screen	End
MD	**Move Down** – Scroll text and cursor down one line.	Ctrl ↓

MU	**Move Up** – Scroll text and cursor up one line.	Ctrl ↑
PD	**Page Down (Next Screen)** – Scroll down one screen	PgDn
PU	**Page Up (Previous Screen)** – Scroll up one screen	PgUp
PF	**Previous Formatted Page** – Move cursor to first line of previous printed page	Alt PgUp
NF	**Next Formatted Page** – Move cursor to first line of next printed page	Alt PgDn
TF	**Top of File** – Move cursor to top of the file	Ctrl Home
BF	**Bottom of File** – Move cursor to the bottom of the file	Ctrl End

DEFINING A BLOCK OF TEXT

DF	**Define Block** – Begin or end defining a block of any size	F1
DW	**Define Word** – Define the word the cursor is on	Alt F4
DL	**Define Line** – Define the line of text the cursor is on	F4
DS	**Define Sentence** – Define the sentence the cursor is on	Ctrl F4
DP	**Define Paragraph** – Define paragraph the cursor is on	Shift F4
DC	**Define Column** – Begin defining a column of text	Alt F1
XD	**Cancel Define** – Release the defined block of text	F3
YD	**Cancel Define** (Variation) – Similar to XD — releases defined text but don't close a footnote screen or other such screen. (Used in programming).	(none)

DELETING TEXT

RD	**Rubout Defined Block** – Erase defined block of text	Alt F6
BD	**Backspace Delete** – Delete the previous character	Backspace
RC	**Rubout Character** – Erase character at point of cursor	Del
RW	**Rubout Word** – Erase the word the cursor is on	Alt Del
RE	**Rubout to End of Line** – Erase text from cursor end of line	Ctr Del
RL	**Rubout Line** – Erase the line the cursor is on	Alt F5
RS	**Rubout Sentence** – Erase the sentence the cursor is located in	(none)
RP	**Rubout Paragraph** – Erase the paragraph the cursor is located in	(none)
UD	**Undelete** – Restore the last text deleted	Alt F3
UP	**Unpad Spaces** – Delete spaces between the cursor and and the first character to its left.	(none)

FOREIGN/OVERSTRIKE CHARACTERS

S1	**Acute Accent**	(none)
S2	**Grave Accent**	(none)
S3	**Umlaut**	(none)
S4	**Circumflex**	(none)
S5	**O Accent**	(none)
S6	**Tilde**	(none)
S7	**Underline**	(none)

HELP

SH	**Show Help** – Display help screen	Alt F9
OP	**Old Prompt** – Switch from your document to the previously accessed help frame	Alt Shift F9
DR	**Display Ruler** – Display menu for setting tabs and margins	Alt Tab
PM	**Prompt** – Display the help frames of the help file one at a time	(none)
HL	**Help** – Displays help frame related to first word on Command Line.	(none)
NR	**Next Ruler** – Toggles the third line in the header between the tab ruler and a solid line.	(none)
$0 to **$9**	**Call Help $0** – Call the help frame with keyword $0 **Call Help $9** – Call the help frame with keyword $9	(none) (none)
$A to **$Z**	**Call Help $A** – Call the help frame with keyword $A **Call Help $Z** – Call the help frame with keyword $Z	(none) (none)

MATH

DT	**Display Total** – Dump accumulated sum into the text at the cursor position	Alt =
SM	**Sum** – Add the number the cursor is on to the total; if block is defined, sum all numbers in block	Alt +
SU	**Subtract Value** – Subtract the number the cursor is located on from the total	Alt –

NORMAL/EXPANDED DISPLAY and PG-LN

SP	**Show Page-Line No.** – Turn on page and line number indicator (P-L) at top of display and switch to Normal Display.	Shift F9
TP	**Toggle Page Normal/Expanded** – Toggle between Normal and Expanded Display; keep Page-Line indicator *on* when returning to Normal.	(none)
CM	**Change Mode Normal/Expanded** – Switch between Normal and Expanded Display, and turn Page-Line No. indicator *off* when returning to normal.	Ctrl F10
XP	**Expanded Mode** – Switch text to Expanded Display.	(none)
WG	**Normal Mode** – Switch text in Normal Display without Page-Line indicators.	(none)
LS	**Line Spacing** – Toggle display between single line spacing and true line spacing.	(none)
NM	**No Markers** – Hide the format Triangles, carriage return arrows and Page-Line numbers. NM also switches the display to Normal if it was in Expanded.	(none)

SAVE/GETS

AD	**Append Define to Save/Get** – Adds the currently defined block of text to the end of the text in the specified Save/Get	Shift F2
SD	**Save/Get Directory** – Display directory of Save/Gets	Alt F2
SK	**Show Save/Get Key** – Show contents of the Save/Get given by next letter or number typed	Ctrl F2 #
SV	**Save Define** – Save the defined block of text to the Save/Get identified by the next key struck	F2 #
@A to **@Z**	**Save/Get Key A** – Place contents of Save/Get A at the cursor location (or run program on Save/Get A)	Alt A
	Save/Get Key Z – Place contents of Save/Get Z at the cursor location (or run program on Save/Get Z)	Alt Z
@0 to **@9**	**Save/Get Key 0** – Place contents of Save/Get 0 at the cursor location (or run program on Save/Get 0)	Alt 0
	Save/Get Key 9 – Place contents of Save/Get 9 at the cursor location (or run program on Save/Get 9)	Alt 9
&A to **&Z**	**Save/Get &A** – Run program assigned with LDPM filename,&A	(none)
	Save/Get &Z – Run program assigned with LDPM filename,&Z	(none)
&0 to **&9**	**Save/Get &0** – Run program assigned with LDPM filename,&0	(none)
	Save/Get &0 – Run program assigned with LDPM filename,&9	(none)

SEARCH

FD	**Find Difference** – Compare the two files in the current and previous windows — stop where files do *not* match	(none)
FM	**Find Match** – Compare two files in the current and previous windows — stop where files match.	(none)
WA	**Wild Alphanumeric** – Interpreted as any alphabetic or numeric character in a search	Alt Shift A
WL	**Wild Letter** – Interpreted as any letter A-Z in a search	Alt Shift L
WN	**Wild Number** – Interpreted as any number 0 thru 9 in a search	Alt Shift N
WS	**Wild Separator** – Interpreted as any word separator in a search	Alt Shift S
WW	**Wild String** – Interpreted as any string from 1 to 80 characters in a search	Alt Shift W
WX	**Wild Any Character** – Interpreted as any character in a search.	Alt Shift X

SYSTEM

BK	**Break** – Stops any command currently in progress (such as delete or search); stops any user program which is running	Ctrl Break
CO	**Comma** – Inserts a comma into the text. CO is used in a keyboard file in place of the punctuation mark (,) for commands that contain a comma (such as IP 0,5). CO is needed since the comma mark (,) is used as a separator. Example: I,P,0,CO,5.	(none)
DX	**Display Off** – Freeze the display (complement of DO)	(none)
DO	**Display On** – Turn on the display (complement of DX)	(none)
FF	**Force Fill** – Force the display to refresh.	(none)
EX	**Exit** – Quit XyWrite, return to DOS (after prompting for verification)	Ctrl Alt Del
PR	**Print Screen** – Print the contents of the full screen (all 25 lines, including header)	Shift PrtSc
RV	**Review** – Preview the document which is currently displayed (same as TYPES)	(none)
EN	**Edit Next File** – Opens the next file that matches a global filename specification	Ctrl N
EF	**Edit Footer, Header or Footnote** – Opens up a footer, header or footnote so that you can edit it without switching to Expanded Display.	Ctrl F3

TOGGLE KEYS

Insert Mode

CI	**Clear Insert Mode** – Switch to Overstrike mode (from Insert)	(none)
SI	**Set Insert Mode** – Switch to Insert mode (from Overstrike)	(none)
TI	**Toggle Insert Mode** – Switch between Insert and Overstrike	Ins

Numeric Lock

CN	**Clear Numeric Lock** – Turn off the Numeric Lock	(none)
SN	**Set Numeric Lock** – Select the Numeric Lock state	(none)
TN	**Toggle Numeric Lock** – Change the state of the Numeric Lock (TN was formerly NK)	Num Lock

Scroll Lock

CS	**Clear Scroll Lock** – Turn off Scroll Lock	(none)
SS	**Set Scroll Lock** – Turn on Scroll Lock	(none)
TS	**Toggle Scroll Lock** – Change the state of the Scroll Lock	Scroll Lock

WINDOWS

AS	**Alternate Screen** – Move cursor between the two windows last displayed	Alt F10
NX	**Next Window** – Move the cursor successively through all open windows (in numerical order)	Shift F10
SW	**Show Window Menu** – Display the window menu	Ctrl F10
#1 to **#9**	**Window No. 1** – Move the cursor to window no. 1	(none)
	Window No. 9 – Move the cursor to window no. 9	(none)

FUNCTION CALLS — Listed Alphabetically

NOTE #1 **@0 - @9, @A - @Z** The @ ("at" sign) followed by a letter or a number is used to get text from the Save/Get key letter or number. For instance @A requests the text saved as 'A' to be inserted at the position of the cursor.

NOTE #2 **M0 - M8** The letter **M** followed by a number between 0 and 8 selects the character mode of text typed in at the keyboard. If there is a block defined, the character mode of the defined block is changed to the selected mode.

NOTE #3 **R0 - R9** The letter **R** followed by a single digit (0 to 9) forms one digit of an ASCII character. The ASCII characters over 128 must be entered this way.

The first time R0 to R9 is encountered, the number following the R is accumulated as a digit of the decimal number. Subsequent keys add further digits to the code number being accumulated. (If a key is struck that is not one of these calls, the code is cleared.) If the state of the keyboard changes (such as a shift key being released) then the accumulated code is given to the program as complete.

As an example (in the IBM.KBD) when Alt Shift is held down, numbers are assigned the R values. If you hold down the Alt and Shift keys and strike the numbers 2,2,4, you get the Greek letter alpha (α) entered in the text when you release the Alt and Shift keys.

NOTE #4

S1 - S7 The letter **S** followed by a number between 1 and 7 is used for entering foreign accented characters. When you make this call, the indicated accent mark appears on the screen. If the letter that is next struck is in the IBM extended character set, the accent mark is replaced by that letter with the accent applied.

The only characters that can be considered for replacement with this scheme are the letters a, A, e, E, i, o, O, u, U, y, n, and N.

These functions approximate the use of *dead keys* on international keyboards. Further effects can be achieved for characters not available in the standard IBM set by the use of the Character Substitution File. You can set up this file to suppress escapement (forward movement of the printer) after the printing of a special character such as a tilde or an accent mark.

NOTE #5

WA, WL, WN, WS, WX, WW These function calls are used to indicate the wildcard entries that are part of the Search string. A complete description is available in the section on the Search command.

SPELLING CHECKER/THESAURUS

Q1-Q8 **Spelling Checker menu functions.** Assigned to F1-F6, Esc, and ↵.

AC **Auto-Check/Correct.** Turn Auto-Check/Correct on and off. Assigned to Ctrl A. Shows "c" in header.

AR **Auto-Replacement.** Turn Auto-Replacement on and off. Assigned to Ctrl R. Shows "r" in header.

FS **Fix Spelling.** Return cursor to last misspelled word and display spelling menu. Assigned to Ctrl F.

SO **Spell One Word.** Check the spelling of a single word. Assigned to Ctrl S.

SY **Synonyms.** Display a list of synonyms. Assigned to Ctrl T.

IR **Insert Replacement.** Display spelling menu without checking for errors and without loading the spelling dictionary. This function allows you to use the spelling menu to enter corrections to the temporary or personal dictionary. Not assigned.

REDLINING

RO **Redlining On.** Turn redlining on and off. Not assigned.

TOGGLE KEYS

TI **Toggle Insert.** Toggle between Insert mode and the active overstrike mode, as defined by the DEFAULT WO setting. Assigned to Ins.

TW **Toggle Word Overstrike.** Toggle between Word Overstrike mode and Insert mode. Word Overstrike goes into Insert mode at the end of a word. When Word Overstrike is active, a lowercase "o" appears in reverse mode in the upper right corner of the header. Assigned to Alt Ins.

TO **Toggle Overstrike.** Toggle between Overstrike mode and Insert mode. Overstrike mode causes text entered at the cursor to overwrite text characters and word separators. When Overstrike mode is active, an uppercase "O" appears in reverse mode in the upper right corner of the header. Not assigned.

MI **Momentary Insert.** Switch from Overstrike to Insert for a *single* insert. You must enter this function call at the exact location that you want to make the insert. Cursor movement cancels it. Assigned to Ctrl Ins.

MEMORY USAGE

ME **Memory Menu.** Display the memory usage menu. Assigned to Ctrl M.

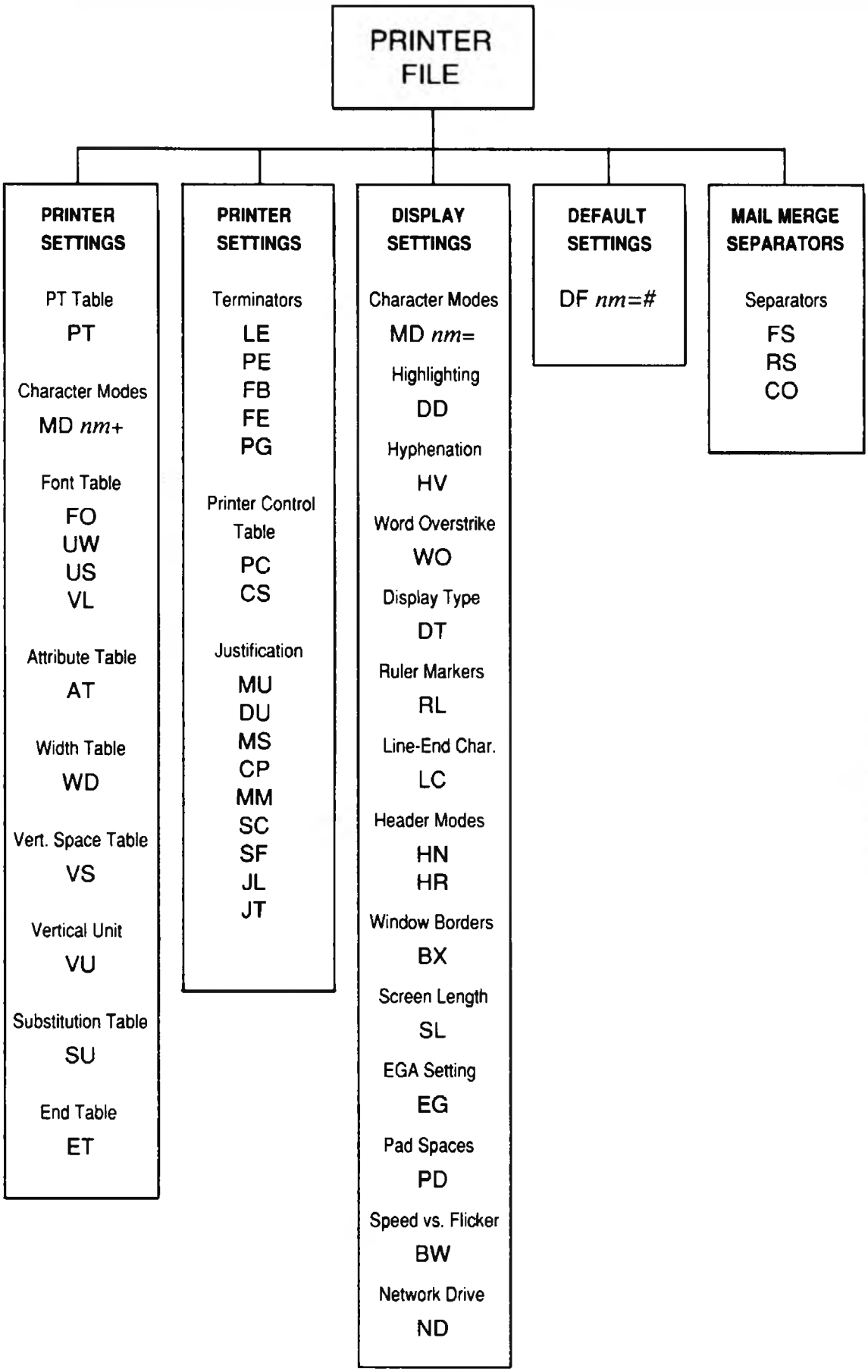
Printer File

INTRO

For the most part, printer manufacturers have adopted different control codes. As a result, software manufacturers have had to develop ways of handling printer information. XyWrite provides Printer Files which contain the codes to control printout — and also control display settings and XyWrite defaults.

For information on how to install one or more printers, refer to Load Printer File (LDPRN) at the end of this section.

CONTENTS


PRINTER FILE
PRINTER SETTINGS
PT Table
PT
Character Modes
MD nm+
Font Table
FO
UW
US
VL
Attribute Table
AT
Width Table
WD
Vert. Space Table
VS
Vertical Unit
VU
Substitution Table
SU
End Table
ET
PRINTER SETTINGS
Terminators
LE
PE
FB
FE
PG
Printer Control Table
PC
CS
Justification
MU
DU
MS
CP
MM
SC
SF
JL
JT
DISPLAY SETTINGS
Character Modes
MD nm=
Highlighting
DD
Hyphenation
HV
Word Overstrike
WO
Display Type
DT
Ruler Markers
RL
Line-End Char.
LC
Header Modes
HN
HR
Window Borders
BX
Screen Length
SL
EGA Setting
EG
Pad Spaces
PD
Speed vs. Flicker
BW
Network Drive
ND
DEFAULT SETTINGS
DF nm=#
MAIL MERGE SEPARATORS
Separators
FS
RS
CO

PRINTER FILE SUMMARY

Display Settings

Display Modes

Format: *mode*=*n* ←
Example: MD NM=7 ←

Ruler Markings

Format: RL<*abcdefghi* ←
Example: RL< ▶ ◀ C ⌂ └ ┐ ||| ←

Header Mode

Format: HN=*n* ←
Example: HN=23 ←

Format: HR=*n* ←
Example: HR=112 ←

Line Character

Format: LC=*n* ←
Example: LC= ←

Printer Settings

Printer Modes

Entering an Attribute

Format: AT<*string* ←
Example: AT< ▶U ←

Exiting an Attribute

Format: AT>*string* ←
Example: AT> ▶D ←

Overstrike Character

Format: AT#*string* ←
Example: AT# ■ ←

Non-Printing Mode

Format: AT* ←
Example: AT* ←

Line-Ending Character

Format: LE<*string* ←
Example: LE< ←

Paragraph Ending Character

Format: PE<*string* ←
Example: PE< ♪ ☺ ←

File Begin Character

Format: FB<*string* ←
Example: FB< ♪ ☺ ←

File End Character

Format: FE<*string* ←
Example: FE< ♪ ☺ ←

Page End Character

Format: PG<*string* ←
Example: PG< ▶ VP ←

Font Table

Format: FO: *name* ←
Example: FO: PICA ←

Use Width Table

Format: UW: *name* ←
Example: UW: PICA ←

Use Substitution Table

Format: US: *name* ←
Example: US: FOREIGN ←

Substitution Table

Format: SU: *name* ←
Example: SU: FOREIGN ←

Width Table

Format: WD: *name* ←
Example: WD: PICA ←

Attribute Table

Format: AT: *name* ←
Example: AT: UNDERLINE ←

End Table

Format: ET ←
Example: ET ←

Print Type:

Format: PT=*n* ←
Example: PT=1 ←

PRINTER FILE SUMMARY *(cont'd)*

Default Settings

Default
Format: DF *nm=n* ←
Example: DF TP=6 ←

Padding
Format: PD=*n* ←
Example: PD=1 ←

Word Overstrike
Format: WO=*n* ←
Example: WO=1 ←

Hyphenation
Format: HY=*n1, n2, n3* ←
Example: HY=6, 2, 3 ←

Display Type
Format: DT=*n* ←
Example: DT=2 ←

Justification Settings

Margin Units
Format: MU=*n* ←
Example: MU=12 ←

Display Units
Format: DU=*n* ←
Example: DU=12 ←

Microspace Units
Format: MS=*n* ←
Example: MS=6 ←

Character Pitch
Format: CP: ←
Example: CP: ←

Microspace Mode
Format: MM=*n* ←
Example: MM=27 ←

Space Constant
Format: SC=*n* ←
Example: SC=3 ←

Space Factor
Format: SF=*n* ←
Example: SF=2 ←

Justify Underline
Format: JL=*n* ←
Example: JL=1 ←

Justification Type
Format: JT=*n* ←
Example: JT=0 ←

Mail Merge Separators

Field Separator
Format: FS<*separator* ←
Example: FS</ ←

Record Separator
Format: RS<*separator* ←
Example: RS<" ←

Comment Separator
Format: CO<*separator* ←
Example: CO<: ←

NOTES

Basics of the Printer File

PURPOSE

A Printer File is a file which contains the settings for a particular printer. It also contains various other settings (listed below). Initially, all you need do is look to see which file applies to your printer, and then load that file with LDPRN (Load Printer File). LDPRN is described at the end of this section.

XyWrite comes with Printer Files for over 70 specific printers. To see these filenames, list the directory of your original XyWrite Word Processor diskette:

Type: F5 dir *.prn ↵

(The printer files are also listed at the end of the Installation Guide.)

To find out which model a particular Printer File supports, call up that file — the model numbers are listed at the top of each Printer File.

The Printer File contains the codes for controlling not only the printer, but also the display and several other XyWrite settings:

- Printer Settings
- Display Settings
- Default Settings (see Default Settings, earlier in this chapter)
- Justification Settings (see Microjustification, later in this chapter).
- Mail Merge Separators (see Mail Merge, Chapter 5)

Only the first two topics are described in detail here; the others are covered elsewhere in this manual, under their respective headings. Refer to the section on Character Modes in Chapter 4 for further description of mode commands (MD NM, MD BO, etc).

ACTION **Making Changes in the Printer File.**
To make changes to the Printer File, do the following:

1. Call the Printer File. Call to the display the printer file you use. For example,

   Type: F5 ca 3epsonfx.prn ↵

   (If you doubt which Printer File to use, call the file STARTUP.INT, and look for the filename having the extension .PRN).

2. **Find the Command.** Search through the Printer File to see if the command you want to enter is already present.

3. **Enter the Setting.** Type the settings you want. For example, to set XyWrite to display bold characters as red on a black background on your color monitor, look up the number in the color table — it's 4. Then enter the setting on its own line.

   Type: md bo=4 ↵

4. **Store and Load the Printer File.** To cause the setting you entered in the previous step to take effect:

   Type: F5 store ↵

   Type: F5 ldprn 3epsonfx.prn ↵

   Result: The setting now is in effect. The MD BO command will cause the characters to display as RED.

NOTE #1 **Initial Configuration.** When you start up the *original* XyWrite disk, it runs through a configuration routine which asks you to pick a Printer File. It then inserts the LDPRN command into the STARTUP.INT file for you.

How Do Printer Files Work?

When you write a document, you can indicate in text where you want attributes such as boldface, underlining and italics. The Printer File specifies exactly how these (and other) attributes are both displayed and printed. You can also change fonts anywhere within your text, using the PT (Print Type) format command — for example, to change from Pica to Elite.

If you want text to be printed **bold** (extra dark), you insert «MDBO» (Mode Bold) into the text (perhaps by pressing Ctrl 2). That command does two things — it affects the *display* and the *printer*. You can control both of these devices independently.

First, «MDBO» changes the look of the text on screen as you write; it might be extra *bright* on your mono chrome monitor, or a specific *color* on a color display. For example, the following statement sets the color of the display text:

```
MD BO=116
```

If you insert this line into a Printer File and load the file, areas marked for bold print will be red on a white background on your color screen.

Second, when you print a document using TYPE, XyWrite inserts the printer codes assigned to MD BO into the text going to the printer. These codes are specified by another line in the Printer File; for example:

```
MD BO+ELITE+EMPHASIZED+DOUBLE
```

Let's say you have loaded the Epson FX Printer File, and you send a file containing bold text to your printer. The two characters ▶G (Escape and G) are inserted into the text in place of the embedded command MD BO. This causes doublestrike to begin. In place of the next MD NM (Mode Normal), XyWrite inserts ▶H to cause the doublestrike to stop.

TERMINOLOGY

XyWrite Printer Files define *fonts* and *attributes* for printers in three general categories: dot matrix, daisy-wheel, and laser print technology.

font — Any printer typeface, such as Pica, Elite, Courier or Helvetica.
attribute — Any printer effect that modifies an existing font, such as underline, superscript or subscript.

These terms have a slightly different meaning for each type of printer:

1. **Daisy-Wheel Printer**

Courier	**PT 1**
Prestige Elite	**PT 2**
Essay PS	**PT 3**

Font — Any print wheel: Courier, Pica, Elite, Light Italic, Boldface, Proportional.
Attribute — Doublestrike, shadow strike, underline, overstrike, superscript, subscript.

2. Dot Matrix Printer

Pica	**PT 1**
Elite	**PT 2**
Proportional	**PT 3**

Font - Pica, Elite, Proportional, Pica/Compressed, Elite/Expanded, etc.
Attribute - Doublestrike, shadow strike, underline, overscore, bold, superscript, subscript, overstrike, double underline, etc.

3. Laser Printer

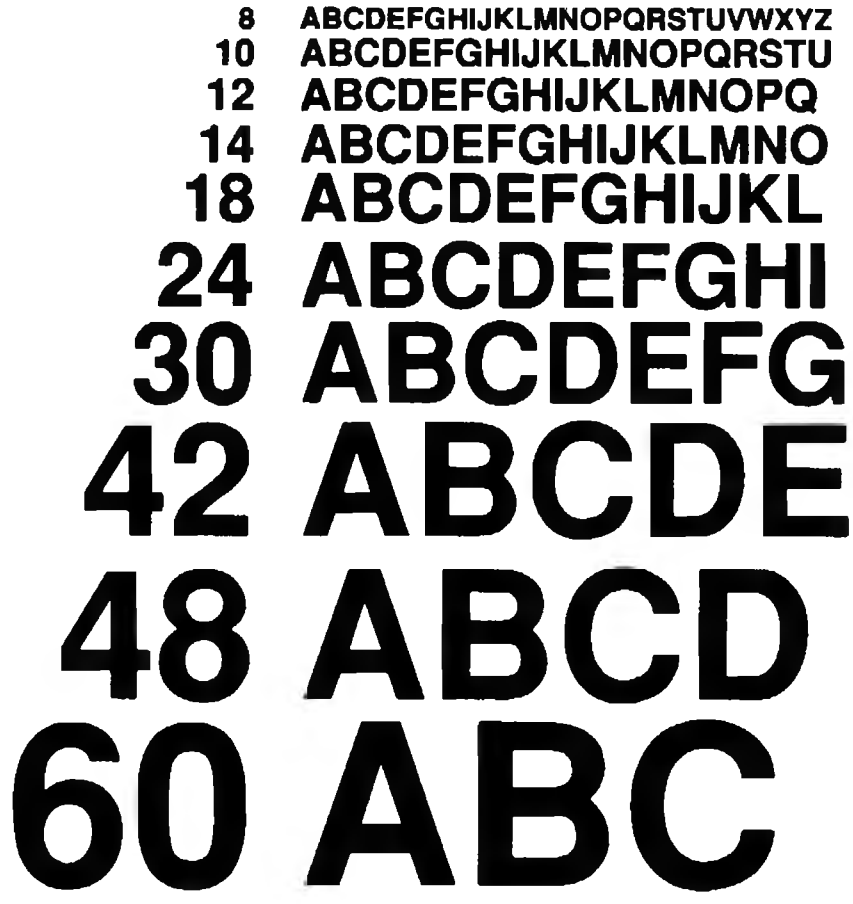
Courier	**PT 1**
Times-Roman	**PT 2**
Helvetica	**PT 3**

Font - Typeface including size. Examples: Courier Bold 10-point, Helvetica Bold Oblique 12-point, Times-Roman Italic 8-point.
Attribute - Underline, superscript, subscript

You define a *character mode* by combining one font and one or more attributes. You will find fonts and attributes for your printer listed at the top of its Printer File.

TERMINOLOGY *(cont'd)*

Point Size. The height of a character (see the figure). You can gauge the point size of a character approximately by measuring from the top of an uppercase letter to the bottom of a "y" or "g" and adding for white space above and below it.

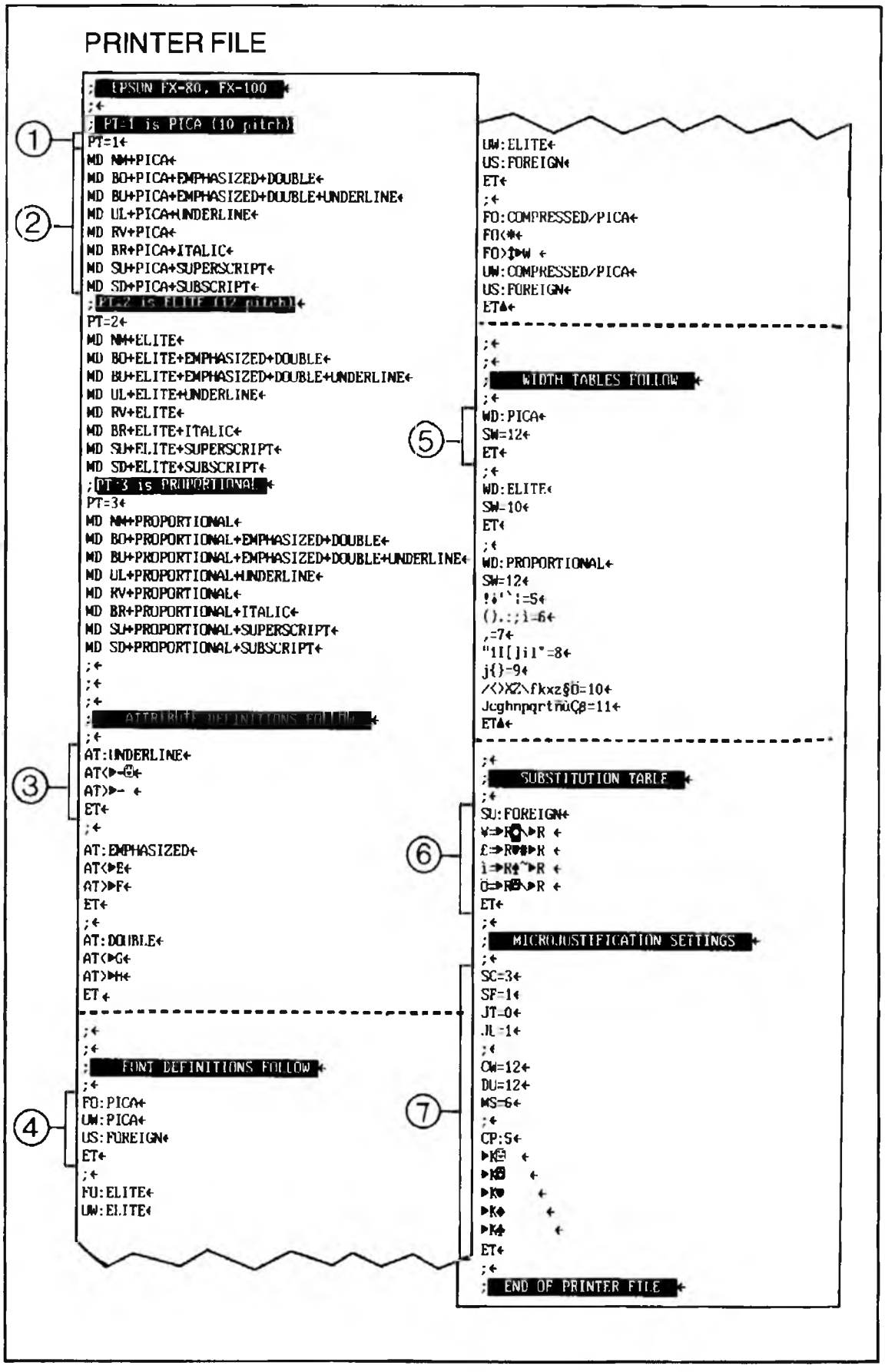


Leading. (pronounced lĕd-ing) The vertical spacing of lines of type, measured from baseline to baseline.

How to Construct the Printer File.
The Printer Files provided on your XyWrite disk are already properly set up to work. However, if you want to *modify* your Printer File, it would be good to understand how it has been constructed. Look at the EPSON FX-80 Printer File example; it has seven basic parts, as follows. All tables start with a name (AT, FO, WD, SU) and end with an ET (End Table) command. Lines starting with a semicolon (;) are comment lines.

1. **PT Tables** (Print Type) — In this example three sets of Character Modes are defined as Print Types, for easy switching between three families of fonts
2. **Character Modes** — Each mode (MD *nm*) with its font and assigned attributes
3. **Attribute Tables** — Control codes for specific attributes
4. **Font Tables** — Definitions and control codes for specific fonts
5. **Width Tables** — Width settings for fonts (note Pica and Elite each have one width for all characters; Proportional has characters of various widths)
6. **Substitution Table** — Control codes for specialized characters
7. **Microjustification Settings** — (covered in the next section of this chapter)
8. **Vertical Spacing Settings** — (covered in this section and the end of Chapter 4)
9. **Terminator Characters** — Characters to end a line, paragraph, page and file
10. **Display Settings**
    a. **Screen** - Headers, ruler, carriage return symbol, borders
    b. **Text** — Normal or expanded display, display modes, highlighting

The last three types of settings are not shown in the Epson example because the Epson printer uses the built-in default values.



Lines beginning with semi-colons (;) are comments.

Typeface Settings

The Typeface Settings section includes all of the assignments that affect the characteristics of printed type (size, weight, slant, etc.). These are: PT Tables (PT), Font Tables (FO), Attribute Tables (AT), Printer Control Strings (<,>,#,%,*), Character Modes (MD*nm*), Width Tables (WD), Vertical Spacing Tables (VS) and Substitution Tables (SU).

NOTE

Comment Lines. Any line starting with a semicolon (;) is a comment. The *entire line* is allocated as a comment — you cannot place a comment to the right of a printer setting. The line is ignored when the Printer File is loaded into memory. For example:

```
;This is a comment line.←
```

You can place comments throughout the Printer File as you wish except you cannot put comments in tables that have a specified number of lines (for example, SU:, CP:, VS:).

PT=*n*

PT Tables. You can easily *change fonts* at any point within your document. In the middle of a 10-pitch document you could print a 12-pitch phrase. To set this up, you use **PT** (Print Type) tables.

In the Epson Printer File illustration, there are three sets of definitions: PT=1 for Pica (10 pitch), PT=2 for Elite (12 pitch) and PT=3 for Proportional. (The term 10 pitch means 10 characters per inch; 12 pitch means 12 characters per inch.)

Each PT=*n* statement has a font number *n* which identifies the font. You use this number in the corresponding PT format command. Once you define the PT tables, you can switch from one font to another by entering the PT format command directly into the text: for example, [F5]PT 3[↵] for the table PT=3.

For further illustration, refer to the section in Chapter 4, "Character Modes and Print Type."

In general, we have set up the Printer Files for various printers in the following way:

PT1 is 10 pitch
PT2 is 12 pitch
PT3 is proportional

The default for PT is 1 (10 pitch). You can change the default to PT2 for instance, in any of three ways:

default pt=2	(in the STARTUP.INT file)
default pt=2	(on the Command Line)
df pt=2	(in the Printer File)

If you have a question about which PT values are defined for your particular printer, call your Printer File to the screen and take a look at the PT tables. Following is an example of the PT (Print Type) printer setting:

```
PT=1←
MD NM+PICA←
MD BO+PICA+EMPHASIZED+DOUBLE←
MD BU+PICA+EMPHASIZED+DOUBLE+UNDERLINE←
MD UL+PICA+UNDERLINE←
;←
PT=2←
MD NM+ELITE←
MD BO+ELITE+EMPHASIZED+DOUBLE←
MD BU+ELITE+EMPHASIZED+DOUBLE+UNDERLINE←
MD UL+ELITE+UNDERLINE←
;←
PT=3←
MD NM+PROPORTIONAL←
MD BO+PROPORTIONAL+EMPHASIZED+DOUBLE←
MD BU+PROPORTIONAL+EMPHASIZED+DOUBLE+UNDERLINE←
MD UL+PROPORTIONAL+UNDERLINE←
;←
```

These character mode assignments are further described next.

NOTE **Chaining Printer Sets.** You can chain printer sets in the Printer File. You would do this if your file had many PT commands in it, for a laser printer, yet you wanted to print that file out on a dot matrix printer, which does not have as many fonts available. The format for chaining printers sets is:

Format: PT=*a,b,c,...*
Example: PT=1,2,3

MD *nm*

Character Mode Assignments. Once the Attribute and Font tables have been defined, you can assign them to character modes. The assigment requires the plus sign (+) operator:

Format: *mode+font+attribute+attribute+...*←
Example: MD BO+PROPORTIONAL+BOLD+UNDERLINE←

This assigns the font PROPORTIONAL and attributes BOLD and UNDERLINE to the character mode MD BO. The *mode* can be any of the following.

MD NM	MD FL	MD 0
MD BO	MD FU	MD 1
MD UL	MD FR	MD 2
MD RV	MD SO	*up to*
MD BU	MD FN	MD 255
MD BR		
MD SU		
MD SD		

You are allowed to use only one font in a mode definition. However, you can use several attributes. If you want to change the standard assignments, check to see if your printer supports the combination you want. (For instance, the Epson printer does not support the EMPHASIZED attribute in combination with COMPRESSED font.)

FO: **Font Tables.** Each Font Table contains the printer codes to access a particular character set. It can also contain the value of vertical leading (VL) associated with that character set and references to both a Substitution Table and a Width Table. (See "Terminology" earlier for examples of fonts.)

The following example of a font table is from the CORONA.PRN file:

```
FO:BKMAN12←
VL=60←
FO<@FONT  BKMAN12←
UW:BKMAN12←
US:SUB←
ET←
```

The first line defines the name of the Font Table, BKMAN12 (Bookman 12 point typeface). You can name the Font Table anything you want, but if you change it, be sure to make the same change in the PT table.

The next line (**VL=**) defines the vertical spacing, or leading value, associated with this font. VL is most useful for printers that support a range of point sizes, because it allows you to change from one type size to another within your document and have the leading change automatically. The VL setting is optional and is included only for laser printer files. It is in effect only if you turn on Automatic Leading (see Chapter 4).

Once you have printed several files in different fonts, you may decide you prefer to have your text set more tightly or loosely than the standard values defined in your printer file. You can change the VL values in the Printer File — for one font or for all of them — but first you should understand how they were originally calculated. To determine the standard leading value associated with each font, set up the following ratio:

$$\frac{\text{Point size}}{72} = \frac{x}{\text{Minimum Vertical Unit}}$$

where "point size" refers to the size of the font being defined; "72" is the number of points in an inch; and

"minimum vertical unit" is the number of the smallest vertical movement the printer can make in one inch. (On some printers, a point is defined as 1/60 inch. Refer to the documentation supplied with you printer.) Let's calculate the leading for a 12-point font on a printer that moves in increments of 1/300 inch.

$$\frac{12}{72} = \frac{x}{300}$$

$$x = 50$$

In this example, the value of VL is 50. If the result of your calculation is a fraction, round it to the nearest whole number. You may adjust the VL value until the vertical spacing is the way you want it.

A second argument in the VL setting affects the baseline calculation for a line of type. If the second argument is 0 or is omitted, XyWrite outputs 2/3 of the first argument *before* the line of text and 1/3 *after* the line. If the second argument has a non-zero value, that value defines the amount of leading that takes place before the line of text. The difference between the first and second arguments determines the amount of leading after each line of text. (See "Automatic Leading" in Chapter 4 for more details.)

The third line (**FO<**) contains the control code that the printer needs to enter the BKMAN12 font. Some tables also contain a line of code that begins with **FO>**; this line of code tells the printer to leave the current font. For more information on control strings, see String Operators which follows.

UW:

UW: (Use Width Table) indicates that the width table named BKMAN12 is used to specify the width of the characters in this font. You may sometimes see a UW: statement that contains a number followed by an asterisk and then the name of a width table. For example:

```
UW:2*pica
```

This Use Width Table statement contains a *scaling factor*. When a scaling factor is supplied, XyWrite uses it to multiply all of the

widths in the specified table to produce a new set of widths for the fonts being defined. In the example shown above, the widths in the table PICA would be multiplied by 2. Scaling factors reduce the number of width tables that need to be created and stored in the Printer File.

US: **US:** (Use Substitution Table) indicates that the Substitution Table called SUB is used with this font.

The **ET** ends the table.

FO= **Font Value.** The FO setting controls several aspects of how your text looks — from underlining to microjustification. Enter the FO setting in the Font Table of the Printer File. The format is:

fo=*v*

where *v* (value) is a decimal number. The values that can be used are made up of combinations from the table below.

Example: If you want to use combinations 1 and 4, then the entry would be fo=5.

VALUE	USAGE
1	Do not output the exit string for the font when microjustifying. This allows underline mode on certain printers to be microjustified. (Assumes that there is an MM mode in the printer table.)
2	Do not output exit string for the font when microjustifying, and do not output the entry string when returning from microjustifying. This is used for superscript or subscripts on printers that roll the carriage up or down.
4	Print discretionary or inserted hyphens in the current mode.
8	Allow microjustification of this font (overrides the JL setting).
16	Don't allow microjustification of this font (overrides the JL setting).
32	This is a foreign language font (which is basically used with special software that recognizes character sets in extra memory such as on an EGA card.)

AT: **Attribute Tables.** An Attribute Table contains the control codes for each attribute. Typical attributes are:

Underline
Bold
Double
Superscript
Subscript
Italic
Forms
Emphasized

For example, to define doublestrike for the Epson FX printer, the Attribute Table would look like:

```
AT:DOUBLE←
AT<:▶G←
AT>▶H←
ET←
```

The **AT:** line specifies the name of the attribute — in this case DOUBLE.

The **AT<** line specifies codes that turn on this attribute. In this case, when you print a file, the two characters Escape and G (▶G) are inserted into the text to *start* doublestrike. For more information on control strings, see String Operators which follows.

The **AT>** line specifies codes that turn off this attribute. Escape H (▶H) is inserted into the text to *end* doublestrike.

The **ET** ends the table.

If you look at a Printer File, you will see that there are already a number of these attributes defined, including the most common printer effects (bold, underline, etc.).

AT= **Attribute Value.** This setting, which you enter in the Attribute Table as AT=*v*, works exactly like the Font Value setting described earlier.

<

>

#

%

*

String Operators. In both the Font Tables and the Attribute Tables you can specify the printer codes that are sent to the printer. For example:

AT<▶U or FO<▶p1

Each statement includes a name (AT or FO), an operator (<) and a string (▶U or ▶p1).

The *string* is the series of printer control codes, and can be any number of characters. The operator tells XyWrite to insert the *string* into the text in a specific manner:

< insert string upon *entering* the character mode
> insert string upon *leaving* the character mode
insert string *after* each character in the mode
% insert string *before* each character in the mode
* insert string *in place of* each character in the mode

The pound (#) and percent (%) symbols are used along with the backspace character (◘) mainly for overstrike.

Entering a Mode

Format: AT<*string*←
Example: AT<▶U←

The two characters Escape (▶) and U are inserted into the text string at the *start* of the attribute.

Leaving a Mode

Format: AT>*string*←
Example: AT>▶D←

The two characters Escape (▶) and D are inserted into the text at the *end* of the attribute.

Overstrike Character

Format: AT#◘*string*←
Example: AT#◘-←

The string "backspace, hyphen" is sent *after* each character. Thus, every character is overstruck with a hyphen. A special case of this follows.

Double-Strike Character

Format: AT#□ ←

The string "backspace, space" is an special case — it causes the printer to move back and print the same character again. (Notice that it does not move back and print a space.)

Pre-Strike Character

Format: AT%*string*←
Example: AT%–□←

The string "hyphen, backspace" is sent *ahead* of each character. Thus, the hyphen is printed, then overstruck by the next character.

Non-Printing Character

Format: AT***string*←
Example #1: AT* ←

Every character is removed and replaced with a single space. Thus, this is the non-printing mode, useful when spaces and all characters are the same size. (When using *proportionally-spaced characters*, each space is narrower than the characters they replace. This can present unexpected results.)

Example #2: AT*←

Every character is removed altogether (and replaced with nothing). This is also a non-printing mode — the displayed characters do not take up any room on the printout.

WD:

Width Tables. A Width Table defines the widths of characters. The table begins with WD (Width Define) and SW (Standard Width) and ends with ET (End Table). SW and the other lines specify the width of characters — the characters are listed on the left of the equal sign (=), and the width is given on the right. The width is given in Basic Units (also called Width Units) — the smallest incremental width of which your printer is capable.

To keep the list orderly, after SW we list the characters in order of size, narrowest first. For example:

```
WD:HELVETICA←
SW=20←
!.:=10←
l=15←
ij=16←
bpqv=26←
WXM=30←
ET←
```

SW (Standard Width) specifies the width of all characters not otherwise specified in the table. You normally assign SW to the width that applies to the most number of characters. This prevents you from having to specifically list those characters in the Width Table.

DU= The value for display units (DU) can be specified in each width table. The following example demonstrates this.

```
WD:PICA←
SW=12←
DU=12←
ijlt=9←
abcdef=15←
ghknopqrsuvxyz=18←
mw=21←
```

The DU=12 statement will cause the markers on the ruler line to be adjusted for this width table. The numbers on the ruler will reflect the current DU setting. Normally, in 10-pitch (MU=12, DU=12), the numbers are 10 dots apart. If you switch to 12 pitch (DU=10), the numbers are 12 dots apart.

You can still enter the DU value in the Printer File *outside* of the width table. When you do this, the DU value affects *all* width tables.

VS: **Vertical Spacing Table.** The Vertical Spacing Table, which is optional, allows you to specify the printer codes for incremental vertical spacing. VS: is the vertical equivalent of CP: (see "Microjustification" later in this chapter). This table starts with VS:*n*, where *n* is the number of lines of code that follow. The next lines are the code directing the printer to move one vertical unit, two vertical units, and so on. For example:

```
VS:3←
@lh 1;←
@lh 2;←
@lh 3;←
ET←
```

The VS table has an effect on the LE and PE commands. See the note "Vertical Spacing" after the descriptions of LE and PE which follow.

VU=

Vertical Unit Setting. The VU (Vertical Unit) setting in the Printer File allows you to work in the vertical unit of your choice (lines, points, etc.). The unit you choose must be a multiple of the minimum vertical movement. For example, you cannot work in points of 1/72 inch if the printer moves in increments of 1/300 inch. XyWrite Printer Files are set up to work in *lines* (1/6 inch).

The VU setting defines three values:

- The input unit multiplier (*x*)
- The screen display divider (*y*)
- The multiplier for internal storage (*z*)

Format: VU=*x,y,z*

Example: VU=3,1,10

Insert this value after any default (DF) commands in the Printer File.

The first value, the *input unit multiplier* (*x*), defines the number of minimum vertical movements the printer must make to equal one of the vertical units you have decided to work in. For example, assume you are using a laser printer that has a minimum vertical movement of 1/216 inch and you want to make your vertical measurements in *points* (1/72 inch). The input unit multiplier *x* is 3, because the printer must move 1/216 of an inch three times before it has made a vertical movement equal to one point (1/216 x 3 = 1/72).

Whenever you issue a Vertical Spacing command (see Chapter 4), XyWrite multiplies the value in that command by the "input unit multiplier." For example, if VU=50 and you issue a Line Spacing command of LS 2, XyWrite sends instructions to the printer that it must move an amount equivalent to 100 minimum vertical movements (50 x 2).

The second value, the *screen display divider* (*y*), converts the output units into screen lines. It affects the way files are displayed on the screen when you are working with column tables, when you are showing true line spacing on the screen (LF=1), or when you type to screen with TYPES.

The third value, the *internal storage multiplier* (*z*), modifies the internal system multiplier so that you can use points (rather than lines). This is necessary when the page length (PL command) is greater than 650 points.

This value converts the number you specify in the Vertical Spacing commands (see Chapter 4) into XyWrite storage units. If you are working in *lines*, the internal storage multiplier *z* is 100, so you could specify hundredths of lines (e.g., 1.25 lines). If you are working in *points*, the minimum conversion factor is 10, which means you can only specify up to tenths of a point.

SU: **Substitution Tables.** A Substitution Table specifies how displayed characters are to be printed. It allows you to construct foreign language characters, graphic symbols and math signs, for instance. This table begins with SU (Substitution) and ends with ET (End Table). Make the character assignments as follows: on the left of the equal sign put the display character; on the right side put the string of printer characters and codes. In the following example, a zero in the text prints with a line through it. See the Character Substitution File later in this chapter for more details on the assignment statements.

```
SU:ZERO←
0=0■/←
ET←
```

NOTE **Common Names.** Width Tables, Substitution Tables and Font Tables can all use the same names. The only constraint is that Attribute Tables and Font Tables should not have the same names.

Terminator Strings

You can add the following assignments in your Printer File to define printer codes at specific points in the document. (Place each statement on its own line.)

Soft return	LE<*string*←
Paragraph End (Hard Return)	PE<*string*←
Beginning of File	FB<*string*←
End of File	FE<*string*←
Between each page	PG<*string*←

LE

Line-End String. LE enables you to specify the characters inserted at the end of each *soft-return* line (that is, each word-wrapped line). These characters are inserted into the text as it is sent to the printer. (Hard-Return lines are handled under PE.)

Format: LE<*string*← Line End
Default: LE< ♪◙←

This string contains a single space, a carriage return (♪) and a line feed (◙). This is for a printer which needs a carriage return and line feed at the end of every line. You enter the carriage return character with Alt Shift 13, and the line feed character with Alt Shift 10

Example #1: LE< ←

This string contains a single space. This example inserts only a space at the end of each line — no carriage return or line feed. If you print to a file (TYPEF) with this setting, the lines within each paragraph remain wordwrapped (for further editing).

Example #2: LE<←

This string contains no characters at all. You could use this to send to the printer thousands of uninterrupted characters, as if they were all one line. For example, you'd use it when sending a long string of graphics data to a dot matrix printer in graphics mode — with no line-end characters inserted.

PE

Paragraph End String. PE enables you to specify the characters inserted in place of each *hard return*. Like all terminator characters these characters are inserted into the text as it is sent to the printer with TYPE (or TYPEF).

Format:	PE<*string*←	Paragraph End
Default:	PE< ♪◙←	
Example:	PE< ←	

The default is single space, carriage return (♪), and line feed (◙). Use the LC setting instead to change the *displayed* character which represents PE. (See "Line Character" in Display Settings.)

FB **File Begin String.** FB allows you to send characters to the printer at the *start* of a file.

Format:	FB<*string*←	File Begin
Default:	FB<←	
Example:	FB<▶G←	

This example sends an Escape-G at the start of each file. On an IBM ProPrinter, this causes the characters to print second strike.

FE **File End String.** FE allows you to send characters to the printer at the *end* of a file.

Format:	FE<*string*←	File End
Default:	FE< ←	
Example:	FE<♀←	

This example specifies that a form feed character be sent to the printer at the end of each file. This would eject the last page in most printers.

PG **Page Break String.** The PG (Page Break) setting lets you send a character string to the printer between pages.

Format:	PG<*string*←	Page Break
Default:	PG<♪◙♀←	

The default sends carriage return, line feed, and form feed to the printer at the end of each page. If you use the PG setting, it is important to include the carriage return and line feed characters if you want the first line of the next page to print correctly. PG does *not* normally send the string after the last page of a file. To do that, you must specify DEFAULT FF=1 *and* DEFAULT EJ=1.

NOTE **Vertical Spacing.** When used in conjunction with the Vertical Spacing table (VS:), LE and PE contain only the codes for a carriage return. FE contains the code to reset vertical spacing to 6 lines per inch.

Printer Control Tables

PC:

Printer Control Table. The PC table allows you to send control strings to the printer. You would use this table to control the printer at specific points within your document. For example, if your printer has two paper trays, you can change trays when you begin a new page. You could use letterhead paper for the cover page of a letter and plain paper for subsequent pages. You invoke the codes in your text file with the PC format command. See the PC command in the "Printer Control" section of Chapter 4 for an example.

You can have only one PC table in a Printer File. The table can contain any number of lines. Each string must be on its own line, and can be any number of characters. You must specify the number of lines of control strings (*n*) at the top of the table (PC:*n*).

Do not use the PC table to change characteristics of fonts (size, weight, spacing, positioning), because that information will not be integrated into XyWrite's internal calculations for horizontal and vertical spacing (for example, line breaks, justification and page breaks).

To give an example:

```
PC:3
▶&1
▶&2
*&e
```

The first line (PC:3) starts the table. The number 3 indicates that three lines of control codes follow.

The three lines of codes perform three different functions. In this case, the first code (▶&1) instructs the printer to switch to paper tray #1. The second code switches to tray #2, and the third code switches to manual feed, for feeding envelopes. These codes are not universal — they are different for different printers.

CS: **Counter String Table.** The CS table lets you specify a list of strings that will be used for counting pages, chapters, footnotes, or other counters (sections, paragraphs, outlines, lists). You invoke this table with the SF *n (Set Footnote Number), SP * (Set Page Number) or counter command DC 1=* (Define Counter).

You can have only one CS table in a Printer File. This table can have any number of lines. Each string must be on its own line, and can be any number of characters. You must specify the number of counter string lines (*n*) at the top of the table (CS:*n*).

When you use this table for counting, the list of strings will eventually be exhausted. After this point, the first string will be used twice, then the second, and so on.

To give an example:

```
CS:4
!
@
*
$
```

The first line (CS:4) starts the table. The number 4 indicates that four counter strings follow.

Let's show what happens when you invoke this table with SP *. When you use PN (Page Number) in a running header or footer, the first page number is printed with "!" in place of the page number, the second page is printed with "@", and so on.

If the CS table is omitted from the Printer File, the default sequence will be *, **, ***, and so on. Thus, the default CS table is:

```
CS:1
*
```

Display Settings

INTRO

Overview of Display Settings. All of the Printer File settings described up to this point control only the *printer*. The remaining settings in this section control only the *display*, not the printer. They can be included in the Printer File anyway (even though they do not affect the printer) or separated out into their own file (also to be loaded with LDPRN).

Display settings fall into two categories: Text Settings and Screen Settings. Text Settings affect the text: its style, color and hyphenation. Screen Settings do not affect the text; they affect the environment surrounding the text — the ruler markers, paragraph-ending arrows, header colors (at the command line and prompt line), the window border colors and screen length.

Creating a Separate Display File. If you want, you can leave all of the display settings out of your normal Printer File and instead put them in a separate file. You might do this if you switch between monochrome and color displays. You could have one display file for a monochrome monitor, another for a color monitor. Like the standard Printer Files, a display file is loaded with the LDPRN (Load Printer) command. You might call this file DISPLAY.PRN. Then you would add the line LDPRN DISPLAY.PRN to your STARTUP.INT file.

Text Settings

The text settings are those settings that affect the characteristics of the text itself on-screen — for example, underlining, color and hyphenation.

Built-In Display Settings. All of the display settings have default values built into XyWrite. For instance, the default normal mode (MD NM) is dim white characters on a black background. The statement for this would be:

```
MD NM=7←
```

Notice that you will *not* find this statement anywhere in the Printer File — it is built into XyWrite. You need to include a setting in the Printer File only if you want to *change* it from the default (for instance, if you had a color monitor and wanted the normal mode to be a color).

DISPLAY SETTINGS

```
;←
; DISPLAY SETTINGS←
;←
;←
; Character Modes←
MD NM=112←
MD BO=116←
MD UL=96←
MD RV=7←
MD BU=100←
MD BR=4←
MD SU=10←
MD SD=6←
;←
; Header Color←
HN=31←
;←
; Header Reverse Colors (for CM, PRMPT, PG-LN)←
HR=100←
;←
; Window Border Colors←
BX=96,112,112,112,112,112,112,112,112,112←
;←
; Defined Block Contrast←
DD=72←
;←
; Cursor Contrast←
CR=119←
;←
; Line-End Character:  (Character displayed for Carriage Return)←
LC=¶←
;←
; Footnote Number Display Mode←
; The value chosen should be used only by FN←
;FN=114←
;←
```

MD *nm*

Display Character Modes. In the previous section we saw how you use a mode command along with the plus sign (+) to specify how character modes are printed, such as MD BO+PROPORTIONAL+BOLD. Use a mode command along with the equal sign (=) to specify how character modes are *displayed*. Use the Color Table (shown later in this section) for the number of the mode. Use the following format in the Printer File, on a line by itself.

Format: *mode=n*←
Example: MD BO=7←

This example reads: Bold mode text is displayed as bright white characters on black background (number 15 from the Color Table). You could specify another number to instead display underline, bold, reverse, flashing, or colors (with a color monitor).

XyWrite comes with the following modes built-in (for a color monitor).

MD NM=7	white (dim) on black
MD BO=15	bright white on black
MD UL=1	blue on black
MD BU=9	black on white
MD BR=120	light blue on black
MD RV=112	gray on white
MD SU=122	light green on white
MD SD=121	light blue on white
MD FL=135	white (dim) blinking on black
MD FU=129	blue blinking on black
MD FR=240	black blinking on white
MD SO=143	bright white blinking on black
MD FN=122	light green on white

DD

Highlighting of Defined Blocks.
The DD (Display of Defined Block) setting allows you to change the highlighting of defined blocks. This setting is useful for low-contrast screens such as those found in lap portables.

Format:	DD=*n*←	Display of
Example:	DD=119←	Defined Block

The number that is given with this setting is combined with the color of the text that is defined. The number 119 seems to be as useful as any.

HV

Hyphenation Control. The HV (Hyphenation Value) setting in the Printer File gives you control over some of the hyphenation rules.

Format:	HV=n1,n2,n3←	Hyphenation
Example:	HV=6,2,3←	

The result is that only words of 6 characters or greater will be hyphenated. At least 2 characters will appear before the hyphen, and in a word that is hyphenated, at least 3 characters will appear after the hyphen. The default is HV=5,2,2.

NOTE

The HV *setting* used to be HY, which caused confusion with the HY (hyphenation on/off) *command*. If you have an HY setting in your printer file, it will be recognized by XyWrite III Version 3.1 as Hyphenation Values. If you have a *default* HY setting to control the way words are hyphenated, however, it must be changed to HV, since default HY controls which hyphenation style (if any) is on when XyWrite starts.

WO

Word Overstrike Editing. The WO (Word Overstrike) setting allows a special overstrike typing mode for text entry. This mode causes text entered at the cursor to overwrite text characters but not word separators. When a separator is encountered, subsequent text is entered in Insert mode.

This overstrike mode replaces the normal overstrike mode of XyWrite and is accessed by striking the Ins key in the usual fashion.

For instance, if you were to place the cursor on the "e" of the word "sacred", and enter the characters "ifice", the word "sacred" would be changed to "sacrifice". Any following text would be moved to fit the extra characters.

The setting is entered by typing WO=1 on a line by itself in the Printer File. The default is WO=0.

DT

Normal/Expanded Display Type. The DT setting specifies how documents are displayed when first called to the screen.

There are four choices and they are given with the following settings. The markers referred to are the line-end symbol (←) and embedded command triangles (▲).

DT=0	The file displays in Expanded Display.
DT=1	The file displays in Normal Display with Page/Line off, markers visible.
DT=2	The file displays in Normal Display with Page/Line turned on, markers visible.
DT=3	The file displays in Normal Display with markers hidden.

If no DT is explicitly called out, XyWrite defaults to DT=1.

Screen Settings

Screen Settings do not affect the text; they affect the environment surrounding the text — the ruler markers, paragraph-ending arrows, header colors (at the command line and prompt line), the window border colors and screen length.

RL

Ruler Markers. The RL (Ruler) command allows you to change the characters shown on the third line of the display - the ruler. The format is:

Format:	RL<*abcdefghi*←	Ruler markers
Default:	RL<▶◀▼⌂└¬▒‖‖←	
Example:	RL<▶◀CDP¬■LR←	

a is the Tab marker for Flush Left Tabs (TS 10)
b is the Tab marker for Flush Right Tabs (TS 10R)
c is the Tab marker for Flush Center Tabs (TS 10C)
d is the Tab marker for Decimal Tabs (TS 10D)
e is the marker for the first Indent Paragraph value
f is the marker for the second Indent Paragraph value
g is the marker for the position of the cursor column
h is the left margin marker (LM 0)
i is the right margin marker (RM 78)

You can substitute any ASCII characters for a ruler marker. If you choose to use a ruler symbol other than the original XyWrite setting, be sure to follow this exact order to replace the correct one.

PRINTER FILE

LC

Line End Character. LC (Line End Character) specifies the character to be *displayed* as a carriage return (hard return). The default setting is a left arrow (←), ASCII 27. If you want to turn off the arrows on the display, change the LC character to a space.

Format:	LC=*c*←	Line Character
Example:	LC=←	

c is any character.

HR **Header Reverse Mode.** HR (Header Reverse) specifies the display mode for CM, PRMPT, NM, P-L — the part of the header which defaults to reverse display mode.

Format:	HR=*n*←	Header Reverse
Example:	HR=100←	

HN **Header Normal Mode.** HN (Header Normal) specifies the display mode for all of the XyWrite header *except* the parts specified above in Header Reverse Mode — that is, the parts which default to normal mode.

Format:	HN=*n*←	Header Normal
Example:	HN=31←	

BX **Window Border Colors.** The BX (Box Colors) setting allows you to define the colors of the borders to the windows.

Format:	BX=*active*,*w1*,*w2*,...,*w9*←
Example:	BX=15,1,2,3,4,5,6,9,10,11←

This would set the border of the active window (*active*) to bright white, window 1 (*w1*) to blue (1), window 2 (*w2*) to green (2), window 3 (*w3*) to Cyan (3) and so on. (Refer to the Color Table for the color combination of your choice.)

The default for all borders is Reverse.

SL **Screen Length.** The SL setting allows you to use a display terminal that has more than the standard 25 lines.

Format:	SL=*n*←	Screen Length
Example:	SL=44←	

This would set the screen length to 44 lines, which would give 3 lines of header and 41 lines of text.

The XyWrite default is 25. The SL setting should not be used with any of the standard 25-line displays — you will get unhappy results if you do. If you are using an EGA adapter, refer to the EG command described next.

EG

EGA Support. If you have an IBM Enhanced Graphics Adapter (EGA), you can use the EG setting rather than SL to change the number of screen lines:

EG=0 25 screen lines
EG=1 43 screen lines

PD

Pad Spaces. When you move the cursor off to the right of a line of text and you begin to enter new text, PD (Pad Spaces) determines whether spaces are added to fill the gap to the *left* of the cursor. PD has an effect only if you use the linear cursor keys. To illustrate, imagine the following: You move the cursor to the right (enabled by Linear Right in the keyboard file), past the end of a line of text (past the line-end arrow). You then begin typing. If PD=1, the text is inserted exactly where the cursor is located, and the gap between the line-end arrow and the new text is filled in with spaces. If PD=0, the cursor jumps left to the line-end arrow and the text is inserted adjacent to the existing text. In summary:

PD=0 Spaces are *not* inserted
PD=1 Spaces are inserted to fill gap

BW

Speed vs. Flicker. BW (Black and White) lets you choose between a fast screen and a clean screen. BW=1 causes XyWrite to update the display faster (and is intended for Black & White screens). You notice the improved speed when scrolling text, switching between windows, and running programs. The cost is a noticeable flicker each time the screen is updated.

BW=0 Clean screen, no flicker (but slower)
BW=1 Fast screen (with flicker)

The built-in default is BW=0. You would use BW=0 when you have a Color Graphics Adapter. Try BW=1 and see if the speed is worth the tradeoff.

Display Tables

Color Table. To select a color (for a color monitor), pick one color for the Characters and another color for the Background. Add the two numbers together. Use this number with the MD command. For example, for bright white characters (15) on a blue background (16), use MD 31.

Characters (Foreground)

0 Black Characters
1 Blue Characters
2 Green Characters
3 Cyan Characters
4 Red Characters
5 Magenta Characters
6 Brown Characters
7 White (Dim) Characters
8 Gray Characters
9 Light Blue Characters
10 Light Green Characters
11 Light Cyan Characters
12 Light Red Characters
13 Light Magenta Characters
14 Yellow Characters
15 Bright White Characters

Background

0 Black Background
16 Blue Background
32 Green Background
48 Cyan Background
64 Red Background
80 Magenta Background
96 Brown Background
112 White (Dim) Background
128 Blinking Characters, Black Background
144 Blinking Characters, Blue Background
160 Blinking Characters, Green Background
176 Blinking Characters, Cyan Background
192 Blinking Characters, Red Background
208 Blinking Characters, Magenta Background
224 Blinking Characters, Brown Background
240 Blinking Characters, White (Dim) Background

Monochrome Table. On a monochrome monitor, the following table is in effect. Add 128 to a number to make it blink. "White on Black" means White Characters on a Black Background.

0	Black on Black (invisible)
1	White on Black, underlined
2-7	White on Black, not underlined
8	Black on Black (invisible)
9	Bright White on Black, underlined
10-15	Bright White on Black, not underlined
112	Black on White
120	Black on White
128	Black on Black (invisible)
129	Flashing White on Black, underlined
131-135	Flashing White on Black, not underlined
136	Black on Black (invisible)
137	Flashing Bright White on Black, underlined
138-143	Flashing Bright White on Black, not underlined
240	Flashing Black on White
248	Flashing Black on White

The only values that don't display are: 0, 8, 128, and 136.

ND

Network Drives. When you are working on a network, several users have access to the same files. This could create problems if more than one person decides to edit the same file at the same time. To avoid this situation, XyWrite provides a way for you to lock out other users when you have a file open for editing. You do this by including the ND (Network Drive) setting in the Printer File. The ND setting defines all the drives in a network (up to a maximum of five) in the following format:

Format: ND=*d1:,d2:,d3:,d4:,d5*
Example: ND=C:

Once you have loaded the Printer File containing this setting, XyWrite prevents anyone in the network from accessing a file while it is open on someone else's screen.

NOTE #1

Changing Disks. If the ND setting is in effect, and if your network drive has a removable disk, **do not remove the disk from the network drive while a file is open from that drive.** Doing so could result in a loss of data.

NOTE #2

Backup. If the ND setting is in effect, the BK (Backup) setting is ignored and XyWrite will not make automatic backup copies of files. You can, however, specify a *save* drive (see "Setting the Default Drive" in Chapter 2 and Note #3 below).

NOTE #3

Defining Multiple Save Drives. If you decide to define several save drives, be sure that only one of them is a network drive and that the network drive is the first one specified.

Special Printer Files

You can construct printer files to perform special tasks (for example, the file STRIP.PRN which we include).

The main use for STRIP.PRN is to produce a file stripped of all embedded XyWrite commands. You might do this to convert a file for use by another word processor, where you don't want the XyWrite double-angle brackets: « and ». Note that using this file also strips out the character modes (Bold, Underline, Reverse).

To use this file, go to the Command line and type:

F5 ldprn strip.prn ↵

Call the file you want to the display and type:

F5 typef oldfile,newfile ↵

where *oldfile* is the original file and *newfile* is the file stripped of XyWrite embedded commands. The *newfile* is the file you can send to another word processor.

STRIP.PRN does the following:

- Inserts any running headers, running footers, footnotes, the soft date, soft time and soft page numbers into the text as fixed text. Tabs are left intact.
- Converts word-wrapped line endings to hard returns. (To make them remain word-wrapped, set LE<)
- Strips out Indent Paragraph (IP) commands. (You will have to reinsert this if you want it included.)
- Removes all character display modes (bold, underline, reverse, superscript and subscript).

When through using STRIP.PRN, be sure to load your normal printer file.

Load Printer File LDPRN

FORMAT **CM LDPRN** filename.prn Load Printer File

filename.prn is the Printer File to be loaded.
LP is an abbreviation for LDPRN.
This is an immediate command.

PURPOSE The **LDPRN** (Load Printer File) command loads the Printer File you specify into memory. This sets up XyWrite to work with your particular printer. (Also see Note #1 if you have a serial printer).

When you first installed XyWrite according to the Installation Guide, the procedure copied LDPRN *filename* into your startup file (STARTUP.INT). This sets up XyWrite for your printer each time you start XyWrite. If you want to change your default printer, you must modify this statement (Step 3 below).

ACTION **Load Printer File.**
To set up XyWrite for a particular printer, load the corresponding Printer File into memory as follows:

1. Look through the .PRN files in the original XyWrite diskette to find the one for your printer. Let's say it's an Epson FX.

2. Enter LDPRN along with the name of the Printer File:

   Type: F5 ldprn epson.prn ↵

   Result: This loads your Printer File into memory, setting up XyWrite to print to that printer. See Note #1 if you have a serial printer.

3. To "permanently" set up XyWrite for your particular printer, make sure the statement in Step 2 above is entered into your STARTUP.INT file. See the Startup File section later in this chapter for instructions.

NOTE #1

Installing a Serial Printer. When installing a printer that runs off your serial port (COM1 or COM2), you must redirect the line printer port (LPT1) to the serial port. To do this, do the following prior to starting XyWrite:

1. Copy MODE.COM from your DOS disk to your XyWrite Working Disk.
2. Type the following two statement *prior* to starting XyWrite with EDITOR. Refer to your printer manual for the proper settings to include — here we've used 9600,n,8,1. The second statement re-directs LPT1: to COM1:.

   Type: `A>mode lpt1:=9600,n,8,1`↵

   Type: `A>mode lpt1:=com1:`↵

   If you have a two-floppy system, you might add these two statements to your AUTOEXEC.BAT file.
3. Start XyWrite:

   Type: `A>editor`↵

   At this point you are in XyWrite, where you can call, edit and store files.
4. After you quit XyWrite, to return LPT1 and COM1 to their original states:

   Type: `A>mode lpt1:`↵

NOTE #2

Automatically Setting Up Serial Printer. If you have a serial printer, you might create a DOS batch file to do Steps 2 through 4 automatically. (This is more appropriate for a hard disk system than a two-floppy system.) This batch file sets up LPT1, starts XyWrite, and then resets LPT1 when you quit. If you name it XY.BAT, you would start XyWrite by typing XY at the DOS prompt. XY.BAT is as follows:

```
mode 9600,n,8,1
mode lpt1:=com1:
editor
mode lpt1:
```

NOTE #3 **Switching Between Printers.** XyWrite supplies Printer Files for many different printers. If you use more than one printer, you need to run the procedure "Load Printer File" for the printer you want. Another way, for the more hearty XyWrite users, is to create a XyWrite program that you run — it might ask "Dot Matrix (D) or Letter Quality (L)?" You would respond with the appropriate letter — D or L. Refer to the example "Select a Printer" at the end of Advanced User Programming in Chapter 5 for the program.

NOTE #4 **The PRN Extension.** We recommend you use the PRN extension to Printer Files (such as EPSON.PRN) for easy identification in the directory. However, XyWrite does *not* require you to use this extension.

Character Substitution File

INTRO

You can construct a Character Substitution table as either (1) part of a Printer File (using SU) or (2) alone as a separate file. In this section we describe how to do the latter. The principles are the same — except in this case you use LDSUB rather than LDPRN.

A Character Substitution File allows you to determine how each displayed character is printed. This is useful for building foreign language characters, graphics symbols and math signs, for instance. It can work just as well for a dot matrix printer as it can for a daisywheel or laser printer.

The Character Substitution File does for your printer what the Keyboard File does for your keyboard — customize it to your liking. To use a Substitution File is quite simple, and requires learning only one new command: LDSUB.

Load Character Substitution File LDSUB

FORMAT

CM **LDSUB** *filename*

filename is the name of substitution file.
LDSUB is an immediate command.
LDSUB was named LO in earlier versions of XyWrite.

PURPOSE

The **LDSUB** (Load Character Substitution File) command loads the specified Character Substitution File into memory for use. This allows you to change any character into a string of characters when you output to a printer. This means that you can change a foreign language character or a graphics symbol to the string of codes necessary to get it to print properly on your printer. You can also use this command to reassign the printwheel characters for a daisy-wheel printer.

ACTION

Substituting Characters for Printout

The rules for building this text file are straightforward. The format is a simple assignment:

n=string

- *n* is any of 256 ASCII characters (see Appendix).
- *string* is the sequence of ASCII characters you want printed in place of *n*.

This format is used in Step 2 below. You can list as many of these assignments as you wish.

1. Create a new file (in which to store the substitutions). Let's call it CHAR1.SUB:

   Type: F5 new char1.sub ↵

2. As an example, let's tell XyWrite to print the number zero with a slash through it (Ø). In other words, substitute for the zero the three characters zero, backspace and slash.

   Type: 0=0◘/↵

   To type the backspace character (◘), you press Alt Shift 8.

3. Type: F5 store ↵

4. Load the character substitution file into memory.

   Type: F5 ldsub char1.sub ↵

Result: Now whenever you print a document, all zeros will appear as (Ø). Character substitution occurs for the commands TYPE and TYPEF.

NOTE #1 **Making Comments in the File.** Any line beginning with a semi-colon that does not have an equal sign as the second character is ignored by the program. You can use this to make comments in this text file.

NOTE #2 **Graphics Mode Example.** Let's show how you output a question mark (?) composed as a bit pattern by going into the graphics mode of a dot matrix printer. (Shown here is the control code sequence to enter graphics mode for 8 characters, followed by the eight characters that are output as a bit pattern for the question mark.) In the character substitution file, type:

?=►S0008 ☻♥QY☼♠ ←

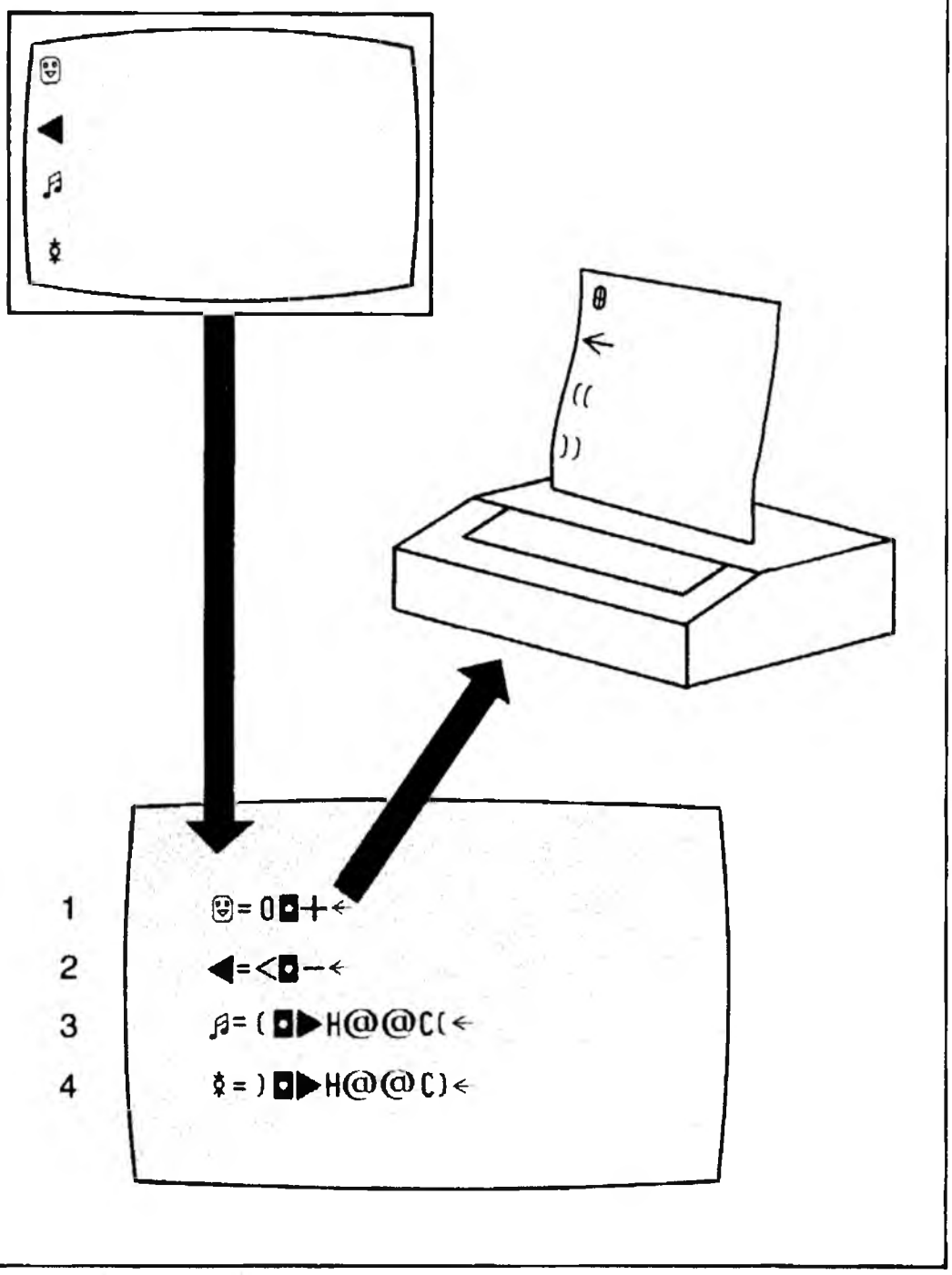
You can build your table to create your own special characters. You should refer to the ASCII table in the appendix for information on entering any of the foreign or special symbols.

NOTE #3

Character Substitution File. A sample file is shown on the facing page. The four lines do the following:

1. The character which is displayed as a smiling face (☺) prints as a plus sign overstriking a zero (⊕).
2. The displayed character ◄ prints as an arrow (←), which is made from a < (less-than) sign overstriking a hyphen (–).
3. The display character ♬ prints (on a Qume LetterPro 20P only) as a double open-parenthesis ((, which is made from one parenthesis, a backspace (■), 3/120″ forward spacing (▶ H@@C — peculiar to this model printer), and a second parenthesis.
4. The display character ♯ prints as a double closed parenthesis)).

CHARACTER SUBSTITUTION FILE



NOTES

Microjustification

INTRO

XyWrite supports the highest quality printing your printer can produce — microjustification. Printer Files supplied with XyWrite are already set up for microjustification. This section is for those who, for one reason or another, want to modify the default settings.

This section contains an in-depth discussion of how microjustification works and how to modify the settings. As such, it is relatively advanced. You can a find more elementary discussion related to Microjustification in the "Alignment" section of Chapter 4.

Microjustification

FORMAT

MU=*n* Margin Units
DU=*n* Display Units
MS=*n* Microspace Units
CP:*n* Printer Escapements

SC=*n* Space Constant
SF=*n* Space Factor

JL=*n* Justify Underline
JT=*n* Justification Type

You enter these settings in the Printer File.

PURPOSE

When you print a document, the XyWrite default is to print the text flush left, with a ragged right edge. This is unjustified text. By embedding a JU format command in your document, the text which follows will be printed justified — that is with text flush against both margins. (Refer to the JU command in Chapter 4, under *Alignment.*) XyWrite justifies text by adding spaces between characters to stretch lines to the same length.

XyWrite offers two ways to justify text:

Whole-Space Justification works by adding whole spaces between the words.

Microjustification, however, adds space in fractional pieces between words and characters, rather than whole spaces only between words. We call these **partial spaces**. The advantage is that the text has a smoother, more refined appearance than with Whole Space Justification.

We have set up each printer for the highest quality printing — microjustification — if the printer supports it. However, microjustification can slow down some dot matrix printers significantly. If this is a problem you can set the Printer File to whole-space justification. (See Note #5).

MICROJUSTIFICATION SETTINGS

The following settings control microjustification and relate format commands and screen display to proportional type. You enter them into the Printer File. These settings take effect only after you load the Printer File with the LDPRN command.

MU **Margin Units** — The MU setting specifies the number of Basic Units in a Margin Unit. Margin Units are the units used in margin commands LM, RM, TS and IP. Thus, LM 10 means set the margin to 10 Margin Units. (MU was formerly called **CW** — these two commands are equivalent.)

DU **Display Units** — The DU setting specifies the number of Basic Units in a Display Unit. Display Units are used in determining the tabs and margins on-screen. DU is usually set the same as MU (Margin Units), though it can be different.

MS **Microspace Units** — The MS setting specifies the number of printer escapement units in a Margin Unit. This setting may be different than MU for printers with character spacing different from print-head escapement, such as the HP LaserJet.

CP **Character Pitch** — The CP setting specifies the number of lines of printer codes that are to follow. The lines that follow CP specify the printer codes to create increments of printer escapements.

SC **Space Constant** — The SC setting specifies the number of partial spaces inserted between words before inserting space between characters within a word.

SF **Space Factor** — The SF setting specifies the ratio of partial spaces added between words to those added between characters within a word *after* the SC spaces have been allocated. A larger number means more spaces will be added between words before they are added between characters.

JL **Justify Underline** — The JL setting enables (JL=1) or inhibits (JL=0) justification of text printed in the underline mode (MD UL). See Note #7. (JL was formerly called JU.)

JT **Justification Type** is a setting reserved for future use. All printers currently require JT=0.

ACTION

Setting the Printer File for Microjustification.
XyWrite Printer Files come set up for microjustification (in the printer's default pitch). Follow this procedure if for some reason you need to modify your microjustification settings. (See Note #1.)

1. **Call your Printer File.** We will use an Epson printer for our example throughout this procedure.

   Type: F5 call 3epsonfx.prn ↵

2. **Margin Units**. Search for the MU setting:

   Type: F5 se /mu/ ↵

   Result: This moves the cursor to MU (Margin Units), the first justification setting in the file. (If MU is not present, look for CW. MU was formerly called CW.)

   Derive the value for Margin Units from your printer manual. MU is described in Note #2. The Epson prints at 120 dots per inch. Thus, all characters are some multiple of 1/120th inch in width. A Basic Unit is 1/120th of an inch. The widest character in the Proportional Space Width Table is 12 Basic Units, which is 12/120ths of an inch. If we set MU to 12, then LM10 will correspond to 1.0 inch, which is convenient.

   Type: MU=12←

   This tells XyWrite that, for instance, when we ask for a Left Margin of 8 we want the left margin to be moved 8 characters each with a width of 12 Basic Units.

3. **Display Units.** Set the value for DU to be the same as MU (Margin Units).

   Type: DU=12←

   Result: Margins appear on the screen similar to how they will appear on the printed page. You might set DU to a value greater than MU if your printer is set up for letters of smaller point size than normal and you want the tabs closer together.

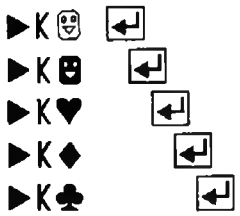
4. **Microspace Units.** Looking in the Epson manual, the escapement (horizontal motion) is given in 1/60ths of an inch. At 10-pitch, each Margin unit is .1", which is 6 printer escapement units. So we give the MS a value of 6. (The Epson printer escapement is 1/60th, *not* 1/120th, of an inch.)

   Type: MS=6←

5. **Character Pitch.** Type a value for CP equal to the number of lines you will enter in Step 6. Be sure to use a *colon*, not an equal sign.

   Type: CP:5←

6. **Codes for Partial Spaces**. Referring to the printer manual if necessary, type in the codes that direct the printer to move one partial space (1/6), two partial spaces (2/6) and so on, all the way up to 5/6. Place these lines immediately following the CP command, one code per line. You must type as many lines as the value given with the CP command (in this case, 5). The set of Epson commands looks like this:



7. **Space Constant.** Set the number of partial spaces you want added between the words before spaces are added between the characters. Let's say we want 3:

   Type: SC=3←

8. **Space Factor**. Set the ratio of partial spaces added between words to partial spaces added between characters within words. For example:

   Type: SF=2←

9. **Justified Underline**. Set JL to enable (JL=1) or disable (JL=0) justification of the underline mode (MDUL):

   Type: JL=1↵

10. Type: F5 store ↵

11. Type: F5 ldprn 3epsonfx.prn ↵

Next, in order to print a justified document, you must insert the JU command into the document — this turns justification *on* for that text. Refer to the JU command in the Formatting chapter for this procedure. Finally, print out the document using TYPE.

Width Table — For printers with varying character-widths (proportional type), the width for each character is specified in a Width Table in the Printer File. (Width Tables are discussed in the Printer File section of this Chapter.)

Basic Units — The Width Table mentioned above uses Basic Units to specify characters. Thus, the line A=5 means the letter "A" is five Basic Units in width.

Margin Units (MU) — The format commands that you use to specify margins and tabs use values we call Margin Units (for example, LM5, RM70). A Margin Unit is defined to be some number of Basic Units with the MU setting.

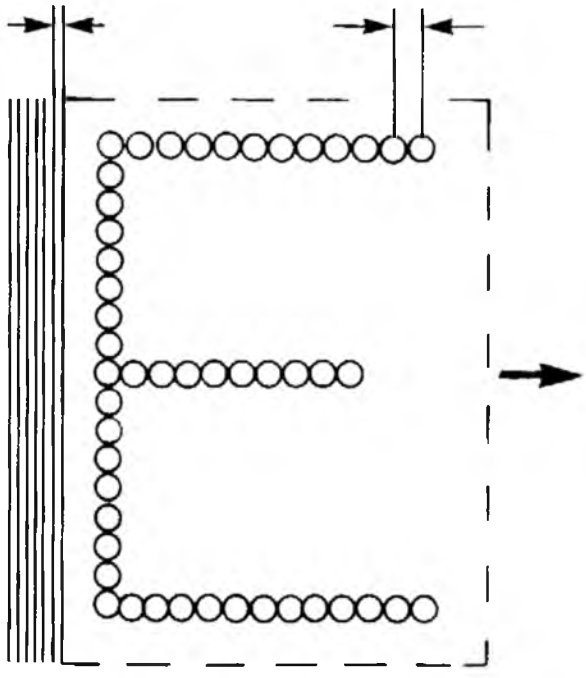
Display Units (DU) — The display of text on the screen sometimes requires an adjustment to account for differences in the size of the characters that print and the size of the characters that are displayed.

XyWrite lets you get an indication of how the text might print out — by adjusting the tabs and margins on the ruler in the display. A Display Unit is defined by the DU command to be some number of Basic Units.

MICROJUSTIFICATION

PRINTER ESCAPEMENT UNIT
Smallest movement of
the printhead

BASIC UNIT
Smallest incremental
width of a character



Printer Escapement Units — A printer escapement unit is the smallest increment of printer escapement. You can derive its value from your printer manual. (Printer escapement is the smallest horizontal distance the printer can move to position a character.)

Some printers have characters that have widths in increments that are different from the increments of printer escapement (where it places the characters). The most notable examples of this are the laser printers that have printer escapement in 1/720th of an inch and character widths in 1/300ths of an inch. (HP-Laser is one.)

Microspace Units (MS) — This setting specifies the number of printer escapement units in one Margin Unit.

NOTE #1 **Printer Files.** XyWrite includes Printer Files for many popular printers. All printers capable of microjustification come set up for it in the printer's default pitch (usually 10 or 12 characters per inch).

However, you'll need to install the proper commands if 1) you are creating your own Printer File with microjustification, or 2) you want to microjustify text in your printer's non-default pitches.

NOTE #2 **Setting Margin Units.** Determine the number of units, or width, that you want to define the numbers you give with the margin commands.

NOTE #3 **How Partial Spaces are Allocated.** Before XyWrite prints a line, it calculates the number of spaces by which the line would fall short of the right margin, and then distributes those spaces between characters according to the values given for the Space Constant and the Space Factor. With SC=5 and SF=2, the first 5 partial spaces are added between words. After those spaces have been allocated, spaces are added in the ratio of 2 between words for every 1 between adjacent characters.

NOTE #4 **Constraints.** Hardware may limit the use of microjustification. In particular:

- You can use microjustification only if your printer supports it. A printer must be able to move in increments of less than one space. Virtually all current-model printers can do this.
- Display monitors cannot move text in increments smaller than one space, so microjustified text will look ragged-right on screen.
- Some dot matrix printers must go into graphics mode each time a partial space is added. This process may be too slow for your needs; if so, switch to whole-space justification according to Note #5.

NOTE #5 **Selecting Whole Space Justification.** To disable microjustification and select whole-space justification, call up your Printer File and insert the statement DF WS = 1 on a line by itself. To switch back to microjustification, use DF WS = 0 instead.

NOTE #6 **Switching Fonts.** You can set up your Printer File to easily switch fonts — for instance, to switch from 10-pitch to 12-pitch. This requires you set up the PT setting in your Printer File and use the PT command in your text. Refer to the Printer File section in this chapter.

NOTE #7 **Justify Underline.** If you use JL=1, depending on the length of your printer's underline, there may be gaps between adjacent underlines. This is due to the partial spaces added between characters. If there are gaps, you should set JL=0 in your Printer File.

NOTE #8 **Try Hyphenation.** Microjustification is an aesthetic feature - it makes your printed copy look better. But occasionally when a long word wraps to the next line, the line can look very stretched when printed. You can turn on automatic hyphenation (see LDDICT) or add soft hyphens to the big words to break them up.

NOTE #9 **Help.** If a file for your printer is not provided on the master diskette, look at the user manual for your printer and take one of the existing files as a guide to insert the codes for the display modes. If you can't figure it out send us a copy of the manual for your printer and a diskette and we'll send you an input file for the Load Printer command.

NOTE #10 You can express Format Commands in tenths of units. For example, LM = 10.5 means set a Left Margin at 10.5 Margin Units.

Sort File

INTRO

Every time you create an index, the entries you identify in your file are arranged in a predefined way. XyWrite gives you the ability to modify the order in which your index entries are arranged.

CONTENTS

PURPOSE

XyWrite has an internal set of sorting rules that automatically arranges index entries in alphabetical order and makes other prioritizing decisions. For example, in multi-word entries, XyWrite alphabetizes the first word only, unless more than one entry has the same first word. The result is a list of entries that looks like this:

```
wild alphanumeric, 6-56
wild letter, 6-57
wild number, 6-50
wildcard, 6-55
```

You can change any of the default sorting rules by creating a new Sort File listing the ASCII characters in the order you want them sorted. (ASCII characters include letters, numerals and other symbols. The complete list is in Appendix A.)

ACTION

Creating the Sort File

The Sort File contains a list of characters in the order in which you want them sorted. The first line of the file contains the character or characters that you want to have sorted first in your list (e.g., aA). The second line contains those characters that you want have sorted second, and so on.

Any characters that you omit from the Sort File are ignored when you create an index.

Let's create a sort file that builds alphabetical indexes, but ignores spaces between words in multi-word entries. It would sort the example above like this:

```
wild alphanumeric, 6-56
wildcard, 6-55
wild letter, 6-57
wild number, 6-50
```

1. Create a new file in which to store the sorting information.

   Type: F5 nospace.srt ↵

2. Enter the first line, which contains the characters to be sorted first. In this example, we leave the space out of the file altogether (normally, it would be the *first* character in the file).

   Type: aA

3. Continue entering all ASCII characters that you want to have included in your index. Be sure to enter them in the correct order; those characters at the top of the Sort File will be sorted first in the index.

   Type: bB
   cC
   dD
   .
   .
   .
   zZ

4. Omit any ASCII characters that you do not want the sort program to use. In this example, omit the blank space from the list.

5. Store the sort file.

   Type: F5 store ↵

6. When you are ready to have an index sorted in the order of your sort file, load the sort file with LDSORT:

   Type: F5 ldsort nospace.srt ↵

The LDSORT command is described in more detail next.

Load Sort File — LDSORT

FORMAT **CM LDSORT** *filename*

filename is the name of a Sort File.
LDSORT is an immediate command.

PURPOSE The **LDSORT** (Load Sort File) command loads a Sort File into memory. This Sort File is a text file that defines the order in which you want index entries arranged.

ACTION **Loading a Sort File.**
To load a Sort File into memory, enter the LDSORT command along with the name of the Sort File you want to load. For example, if you have created a file called NOSPACE.SRT:

Type: F5 ldsort nospace.srt↵

Result: The Sort File NOSPACE.SRT is now in effect. When you create an index, the blank spaces between words will be ignored so the entries look like this:

```
wild alphanumeric, 6-56
wildcard, 6-55
wild letter, 6-57
wild number, 6-50
```

NOTE **Startup.** In order to use a Sort File you have created, you must load it each time you run XyWrite. You can have your Sort File automatically loaded at startup by adding the LDSORT command to your STARTUP.INT file.

Startup File

INTRO

The STARTUP.INT is one of the files we provide to enable you to customize XyWrite to your needs. This file tailors the initial conditions, such as loading a Printer File for your printer and loading your own Keyboard File.

It doesn't take much time to learn about the STARTUP.INT file, since there are no new commands to learn. Incidentally, if you are familiar with DOS, you'll notice the parallel between STARTUP.INT, which initializes XyWrite, and AUTOEXEC.BAT, which initializes DOS.

PURPOSE

STARTUP.INT is the *initial startup* file for XyWrite. It contains commands that are automatically executed every time XyWrite is started up. These commands set up XyWrite to be custom-tailored for you. They can load your Printer File, set the default page length, and set the default drive, for example. A sample STARTUP.INT file is shown on the opposite page.

STARTUP.INT is actually a program file (see "User Programming"). XyWrite comes prepared with a simple STARTUP.INT file; you are welcome to modify it to include any commands you like.

ACTION

Running the STARTUP.INT File

The STARTUP.INT file is run two different ways: automatically and manually.

- **Automatically.** Each time you *start* XyWrite, STARTUP.INT runs automatically:

  1. Start at the DOS prompt (A> or B>):
  2. Type: editor↵

  Result: XyWrite is loaded, the familiar XyWrite screen appears and STARTUP.INT runs automatically. Each command is executed in sequence, appearing briefly on the Command Line. Upon completion, the XyWrite serial number screen is displayed. Press any key to make it vanish (we suggest you press F5 since the keystroke is accepted).

- **Manually.** You can run STARTUP.INT any time you want while in XyWrite. You would do this to re-initialize XyWrite, as follows:

  1. STORE any documents you are working on before you run STARTUP.INT (This is only a precaution — if your startup file contains the ABORT command, it will clear any document from the display.)
  2. Type: F5 run startup.int F9

  Result: Each command in the file is executed, one after another.

A Sample STARTUP.INT File.

```
BC default bk=1←                    (1)
BC default pl=54,60,50←             (2)
BC ldhelp long.hlp←                 (3)
BC ldprn epson.prn←                 (4)
BC abort←                           (5)
BC ldkbd ibm1.kbd←                  (6)
BC ldsgt altkeys.sgt←               (7)
BC ldsub char.sub←                  (8)
BC b:←                              (9)
BC                                  (10)
```

The ten lines do the following things:

Line 1. Sets XyWrite to keep a previous version of each file as a backup copy.

Line 2. Sets the PL command (Page Length) to nominal 54 lines, maximum 60 lines, and minimum 50 lines.

Line 3. Loads all Help files into memory, for quick access.

Line 4. Loads the Printer File into memory — setting XyWrite up, in this case, for an Epson printer.

Line 5. ABORT, which clears the display, putting into effect any new display settings that were loaded as part of the Printer File in step 4.

Line 6. Loads the keyboard file named IBM1.KBD into memory, putting that keyboard into effect.

Line 7. Loads the Save/Gets from the file ALTKEYS.SGT, (a set that you use regularly).

Line 8. Loads the Character Substitution File CHAR.SUB

Line 9. Sets drive B as the default drive.

Line 10. Clears the command line. Notice there is no carriage return following the **BC**, as indicated by the absence of the left arrow (←).

ACTION

Modifying the STARTUP.INT File

The following procedure shows you the simplest way to modify the STARTUP.INT file. For a more formal approach, refer to "User Programming".

Let's modify the STARTUP.INT file shown below so that XyWrite will start up with the directory displayed. To do this, you add one command (DIR) to the end of the STARTUP.INT file.

The letters **BC** which begin each line are described in the note below titled "Analysis." To create a new line beginning with **BC**, as a shortcut we will simply *copy* another line in Step 2 (rather than go into User Programming).

1. Call the existing STARTUP.INT file to the display:

   Type: F5 `call startup.int` ↵

2. Each line must begin with the special letters **BC**. To do this:

   Move the cursor to the *left end* of the ABORT line.

   Press: F4 (to define the ABORT line)
   Press: F7 (to copy the ABORT line)
   Press: F3 (to release the defined line)

   Result: The ABORT line now appears twice.

3. Move the cursor up one line and to the right one character. It should now be on the letter A of ABORT. We want to keep **BC**, delete ABORT, and type in DIR where ABORT was:

   Press: Ctrl Del (To delete the word ABORT)
   Type: `dir`

   Result: The DIR line is now finished.

4. Now store the file:

   Type: F5 store ↵

5. And finally, test the file:

   Type: F5 run startup.int ↵

Result: The commands in the STARTUP.INT file should appear one at a time on the Command Line, each executing in turn. The last command, DIR, should display the directory.

NOTE #1 **Analysis.** Each line in a STARTUP.INT file represents a series of keystrokes.

- **BC** (Blank the Command Line) represents the key: F5
- The return arrow (←) represents the key: ↵

For example, here is one line and the keystrokes it represents:

```
BC ldprn 3epsonfx.prn←
F5 ldprn 3epsonfx.prn↵
```

NOTE #2 **Order of Commands.** The order of commands in STARTUP.INT can be very important when setting defaults. The rule is: The last setting takes precedence. For example, in Case 1 below, if 3EPSONFX.PRN sets the offset to OF 8 (with the statement DF OF 8), then the offset would be overridden by the next statement, DEFAULT OF=12.

Case 1:

```
BC ldprn 3epsonfx.prn←
BC default of=12←
```

Similarly, in Case 2, DEFAULT OF=12 is overridden by the default setting in 3EPSONFX.PRN.

Case 2:

```
BC default of=12←
BC ldprn 3epsonfx.prn←
```

EXAMPLES

Additional Commands. Spark your imagination. In addition to the commands shown on the previous page, your STARTUP.INT file might include commands to:

BC dir←	Displays the directory.
BC call**CC**←	Displays the directory *and* types CALL on the Command Line. You can then simply move the cursor onto the filename you want and press F9 to call that file.
BC b:,a:←	Assigns a default drive (driveB) plus a save drive (drive A).
BC ldpm sample.pm,s←	Loads a program file (SAMPLE.PM) to a Save/Get key (S).
BC default tp=6←	Sets a default top margin of 6 lines for all documents.
BC window 1,1,1,80,22← **BC** window 2,40,1,80,22←	These two lines open windows 1 and 2 and set their sizes.
BC run choice←	Runs a user program named CHOICE. This might be a program that lets you choose which printer you want to use — and loads the corresponding printer file.

Two caveats about including RUN in a STARTUP file:

1. Be sure to end the program (CHOICE) with «EX». This ensures that program execution will return to thenext statement in STARTUP.INT.
2. If the program CHOICE contains a Read Character statement «RC», you must position RUN CHOICE so that it runs with *no window open*. That is, RUN CHOICE must be located *ahead* of any WINDOW statement (or DIR or CALL).

Appendixes

INTRO

This section contains useful reference information.

CONTENTS

PURPOSE

The chart on the following two pages shows all 256 ASCII characters. This is the complete set of characters that can be displayed on an IBM PC (in text mode). You can enter foreign characters, graphic symbols and math symbols by the following method.

ACTION

Typing An ASCII Character

To type an ASCII character into text:

Press and hold: Alt Shift

Type the ASCII number at the numeric keypad. For example, to enter ASCII 174:

Press: 174

Result: When you release Alt Shift, the ASCII code appears on the screen.

NOTE #1

Three-Byte Characters. The codes 0, 8, 9, 10, 13, 26, 27, 32, 174, 175, and 255 are entered into the text file as a special three-byte sequence (but are *displayed* as single characters). All others are entered as a single byte. However, all codes entered using Alt Shift are output as a single byte by the TYPE command.

NOTE #2

Escape Character. You can enter the Escape character into text two ways: by pressing Esc, or by pressing Alt Shift 27. The Escape character is sent to the printer as ASCII 27, the left arrow ←. However, the Escape character is *displayed* as the left-pointing triangle, ASCII 17 (◄), to differentiate it from the arrow displayed at the end of each line. The Escape character is often used in sending control codes to the printer.

ASCII Characters

0	BLANK (NULL)	16	▶	32	BLANK (SPACE)	48	0	64	@	80	P	96	`	112	p
1	☺	17	◀	33	!	49	1	65	A	81	Q	97	a	113	q
2	☻	18	↕	34	"	50	2	66	B	82	R	98	b	114	r
3	♥	19	!!	35	#	51	3	67	C	83	S	99	c	115	s
4	♦	20	¶	36	$	52	4	68	D	84	T	100	d	116	t
5	♣	21	§	37	%	53	5	69	E	85	U	101	e	117	u
6	♠	22	▬	38	&	54	6	70	F	86	V	102	f	118	v
7	•	23	↨	39	'	55	7	71	G	87	W	103	g	119	w
8	◘	24	↑	40	(	56	8	72	H	88	X	104	h	120	x
9	○	25	↓	41	)	57	9	73	I	89	Y	105	i	121	y
10	◙	26	→	42	*	58	:	74	J	90	Z	106	j	122	z
11	♂	27	←	43	+	59	;	75	K	91	[	107	k	123	{
12	♀	28	∟	44	,	60	<	76	L	92	\	108	l	124	¦
13	♪	29	↔	45	—	61	=	77	M	93	]	109	m	125	}
14	♫	30	▲	46	.	62	>	78	N	94	^	110	n	126	~
15	☼	31	▼	47	/	63	?	79	O	95	_	111	o	127	⌂

The following eight characters are interpreted by printers to mean:

0 = NULL, 07 = BEEP, 08 = BACKSPACE, 09 = TAB, 10 = LINE FEED,
12 = FORM FEED, 13 = CARRIAGE RETURN, 27 = ESCAPE

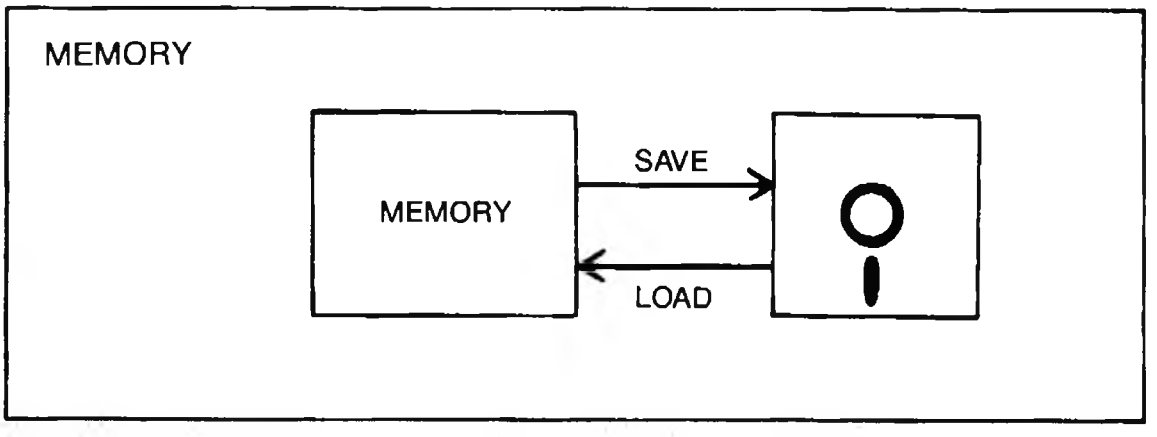
128	Ç	144	É	160	á	176	▒	192	└	208	╨	224	∝	240	≡
129	ü	145	æ	161	í	177	▓	193	┴	209	╤	225	β	241	±
130	é	146	Æ	162	ó	178	▓	194	┬	210	╥	226	Γ	242	≥
131	â	147	ô	163	ú	179	│	195	├	211	╙	227	π	243	≤
132	ä	148	ö	164	ñ	180	┤	196	─	212	╘	228	Σ	244	⌠
133	à	149	ò	165	Ñ	181	╡	197	┼	213	╒	229	σ	245	⌡
134	å	150	û	166	ª	182	╢	198	╞	214	╓	230	μ	246	÷
135	ç	151	ù	167	º	183	╖	199	╟	215	╫	231	τ	247	≈
136	ê	152	ÿ	168	¿	184	╕	200	╚	216	╪	232	Φ	248	°
137	ë	153	Ö	169	⌐	185	╣	201	╔	217	┘	233	Θ	249	∙
138	è	154	Ü	170	¬	186	║	202	╩	218	┌	234	Ω	250	·
139	ï	155	¢	171	½	187	╗	203	╦	219	▉	235	δ	251	√
140	î	156	£	172	¼	188	╝	204	╠	220	▄	236	∞	252	n
141	ì	157	¥	173	¡	189	╜	205	═	221	▌	237	φ	253	²
142	Ä	158	₧	174	«	190	╛	206	╬	222	▐	238	∈	254	■
143	Å	159	ƒ	175	»	191	┐	207	╧	223	▀	239	∩	255	BLANK

XyWrite has a special capacity to handle large files comfortably, up to the limits imposed by the size of your memory. XyWrite will make the most of your memory — it will use up to 640K — the limit imposed by DOS. More memory results in faster performance, especially when editing large files.

Your computer holds information in two ways: in **memory** and on **disk**.

- **Memory** is also called RAM, which stands for Random Access Memory. This is the part of the computer where information (such as the XyWrite program and the text file you're currently working on) is kept while it's being used. The display terminal is your window into memory; you cannot view text on the screen unless it resides in memory. (To view text that is stored on disk, you must first *load* (copy) it into memory.) Any information kept in memory is erased when the computer is turned off.
- **Disks** include both floppy disks and hard (fixed) disks. This is where information is stored on a permanent basis. You load information from disks into memory for use. When done working with the information, you store or save it back to the disk from memory.

Loading to Memory. When you first turn on your computer, the memory is blank and all of the programs are on disk. You need to load XyWrite from the disk into the memory in order to use it.



The XyWrite III program can occupy anywhere from approximately 140K to almost 300K of memory, depending on which files are loaded: printer file, help file, Save/Get file, hyphenation dictionary file, and keyboard file. Data files can take up the remaining memory. (1K = one kilobyte = 1024 characters).

When you call a text file, you are copying the file from the disk into memory. The changes you make while working in the file exist only in memory, and are not recorded on the disk until you save the file.

The XyWrite commands which load from disk into memory are:

CALL *filename*	Loads the specified file into memory and display
DIR	Loads the disk directory into memory and display
CAP *filename*	Loads the specified program file into memory and display
CAF *filename*	Loads the specified form document into memory and display
LDSGT *filename*	Loads the specified Save/Get Key file into memory only
LDPRN *filename*	Loads the specified printer file into memory only
LDSUB *filename*	Loads the specified character substitution file into memory only
LDKBD *filename*	Loads the specified keyboard file into memory only
LDPM *filename,#*	Loads the specified program file into memory, attached to the specified Save/Get key
LDHELP *filename*	Loads the index of the specified Help File into memory.
MERGE *filename*	Loads the specified file into memory and display (at the cursor location)

Clearing from Memory. The XyWrite commands which clear data from memory are:

ABORT	Erases the current file from the display and memory, and does not disturb the disk files.
CLRSGT	Erases the current Save/Get keys from memory.
REMOVE	Clears a single Save/Get or User Program from memory.

Saving to Disk. When you turn off the computer, any information left in memory is wiped out, so it's important to store your documents onto disk before quitting. Examples of saving to disk are the commands:

SAVE *filename*	Saves the current file to the specified file on disk.
STORE *filename*	Saves the current file to the specified file on disk and clears the display.
STSGT *filename*	Saves the current Save/Get Keys to disk.
SAVEDEF *filename*	Saves the defined block of text to disk.

Deleting from Disk. The XyWrite command you use to delete files from the disk is ERASE.

ERASE *filename*	Erases the specified file from disk. (DEL is identical to ERASE)
ERNV *filename*	Erases the specified file without stopping to verify.

Related Commands. Keeping in mind the distinction between memory and disk allows you to understand commands better and use them more confidently. You can know where information is at any point in time. For instance:

- NEW creates a new file in memory, but does *not* create a new file on disk. The file is not created on disk until you SAVE or STORE it.
- STORE performs exactly the same function as SAVE followed by ABORT. STORE saves the current document and clears the display.
- Windows are all allocated in memory. When a new window is opened, part of memory is allocated to that window.
- If a file is too large to fit into memory when using CALL (or when typing in new text), XyWrite automatically creates two files to contain the overflow — one for the overflow at the top of the file, the other for the bottom overflow. (See "Overflow" on the next page.)

Recommendation. To make the best use of your computer's memory:

Keep files to a size where they fit entirely into available memory. If your file gets so large that it starts scrolling the excess to disk (into the temporary overflow files), try breaking the file into smaller files.

Primary File Buffers. Each file that you open has its own buffer called a Primary File Buffer. The buffer for each file starts small and grows in size as the file grows — the buffer can grow as large as 64K. The file can continue to grow in size *past the 64K*, to the limit of available memory (at which point it begins overflowing to disk). Thus:

> **XyWrite will use *all* available memory before creating temporary overflow files on disk.**

You can open as many as nine files — each will be in its own buffer, and each buffer can grow to 64K (assuming you have that much memory). Save/Gets are allocated to their own buffer, which can be as large as 50K.

As we just stated, the Primary File Buffer does *not* impose a limit on the size of files you can load into memory.

For example, if your system has 256K of RAM, and you load 40K for DOS and 120K for XyWrite, you can load one file as large as 96K (256K — 40K — 120K) before overflow occurs. The absolute limit to the size of a file, if you include overflow files, is determined by the amount of storage on your disk.

Overflow. If a file is too large to fit in available memory, the overflow at either end is stored onto disk in temporary overflow files. While the files appear to be given random names (such as CKDBRBHD.TMP and DMFAABDA.TMP) their names are in fact derived from the date and time so as to be unique. The names have the extension TMP (temporary). These files are created automatically whenever they're needed. When you store the file, the parts are recombined and the file is saved as a whole.

Constraint Imposed by Primary File Buffer.
The Primary File Buffer imposes a limitation on defined blocks of text. (This limitation is seldom-encountered in practice.) This constraint is best demonstrated by an example.

If you define a paragraph at the beginning of a 70K document and then attempt to move it to the bottom of the file, you will get the message CAN'T SCROLL DEFINE OR DISPLAY. This message means XyWrite cannot scroll the defined block or the display. It indicates that the Primary File Buffer cannot contain all the text from the defined text to the bottom of the document. The solution is for you to store the paragraph in a Save/Get, *release the defined text*, and then move to the bottom of the file and insert the Save/Get into the text.

The two constraints imposed by the Primary File Buffer are:

1. **Moving or Copying a Defined Block of Text.** If you want to define a block of text and move or copy it a long distance within the file, the size of the Primary File Buffer limits how far you can move it — you cannot move it farther than 64K.

2. **Defining a Large Block of Text.** You cannot define a block of text larger than the Primary File Buffer itself (64K).

This appendix describes the error messages and system prompts for XyWrite III. These messages appear on the prompt line (the second line) of the display.

ABORT, STORE OR OPEN A WINDOW There is already a file open on the screen.
Action: STore or ABort the document on the screen, or press Ctrl F10 to display the Window Menu and open a new window.

ALPHANUMERIC KEY REQUIRED When Saving a defined block, you must strike a letter or number key immediately after striking F2.
Action: Start over by striking F2 followed by the appropriate letter or number.

ALPHANUMERIC REQUIRED When you use the Insert Save/Get command you must supply a letter or number for the desired Save/Get.
Action: Supply the letter or number for the desired Save/Get.

ALREADY TYPING The TYPE command is already outputting to the printer.
Action: Wait until the printer is finished and retry the command.

ARG REQUIRED When using the Insert Save/Get command, you must provide the letter or number of the Save/Get you wish to insert.
Action: Press F10 to return to the command line, and enter the IS command again--this time with the corresponding number or letter.

BAD CONFIG LINE XyWrite is unable to interpret one or more of the printer configuration lines when executing an LDPRN command.
Action: If you watch the loading of the table, there is a beep for each line in error. Call the file to the screen and check that all comment lines are preceded by a semi-colon and that all of the named modes (e.g., MD NM, MD BO) are valid.

BAD LINE There is something wrong with one of the lines in the Character Substitution file.
Action: Check the file to see that all comment lines start with a semi-colon.

BAD DEST FILE No Destination file has been named for the command.
Action: Refer to the reference manual page that describes the proper entry of the desired command.

BAD PATH XyWrite is unable to follow the given DOS 2.0 path.
Action: Check to see that all of the directory names are correct and linked in the indicated order.

BAD STRING When entering a Change or Search command you have not entered a string, or the string does not have the correct separators.
Action: Refer to the descriptions of these commands in this manual.

BAD TABS CMD XyWrite expects the tab column values to be given in ascending order.
Action: Delete the TS command from the text and re-enter.

BREAK The Ctrl-Break key combination has been struck while executing a CH, CV, or SE command. The current command is aborted.
Action: If you wish to continue with the CH, CV or SE, press F9. If you are finished, no action is required.

BUFFER FULL You have exceeded the text buffer capacity of your system.
Action: Save the file. If you have a second screen active, reset the first screen (RS), and save the second file as well. If you have exceptionally large Save/Gets, save them to the disk and execute a CLRSGT. Now you can try reopening the files that you have just stored. It may be advisable to work in one screen only if your files are long.

CAN'T FORMAT WHILE PRINTING Same as ALREADY TYPING.
Action: Wait until the printing is complete and retry the command.

CAN'T GET MEMORY There is not enough memory to complete the function you have requested.
Action: See "Out of Memory" message.

CAN'T NEST X, RF, RH, FN CMDS The Index Marker (X*), Running Header (RH), Running Footer (RF), and Footnote (FN) commands cannot be given one inside of the other.
Action: Rearrange the order of the commands so they are not nested.

CAN'T OPEN INDEX FILE XyWrite is unable to open a file to create an index.
Action: Check the directory to be sure there is enough space on the diskette. Make sure the diskette is not write protected.

CAN'T REASSIGN MEMORY XyWrite cannot make enough room to go to DOS or load the indicated program.
Action: Try storing any open files and retry the command.

CAN'T REVIEW WHILE PRINTING Just as in TYPEF and TYPE, you cannot TYPES while printing.
Action: Wait for the printer to finish and proceed with TYPES.

CAN'T RUN COMMAND The DO command cannot run the indicated program.
Action: Check to see that the named program is on the diskette and in the current directory. (It may also be that there is not enough room in memory to run the program.)

CAN'T SCROLL DEFINE OR DISPLAY XyWrite does not allow a block that is actively defined to be buffered to the diskette in either of the two temporary files that XyWrite creates.
Action: Press F3 to release the defined block. Or press F2 X to save it to a Save/Get key for later use, and then release the define.

CAN'T SCROLL OUT OF REGION When entering text after the F3 TERMINATES TEXT INPUT prompt, you cannot move your cursor outside of the input window until you are done.
Action: If you want to scroll out of the cursor area, first press F3 to close the window.

CAN'T SCROLL A disk error is reported from DOS.
Action: Either save the file to another drive or abort the file.

CAN'T CREATE TEMP FILE When you're doing background printing and a file exceeds the capacity of the buffer memory, XyWrite automatically creates .TMP files on the default drive. This message means the default disk is write-protected or full.
Action: Check to see if the disk is write-protected. If the diskette is full, you can recover by placing a blank formatted diskette in the default drive.

CHANGE ABORTED XyWrite's reply when you enter A in response to the TYPE A, N, S, OR Y prompt displayed in the CV command.

COMMAND INPUT ERROR The command that has been entered cannot be acted upon by XyWrite.
Action: Press F3 (reset defined block); this may fix it. Refer to the page describing the command to verify that it contains all of the required information. Go to Expanded Mode Ctrl F9 to see if any embedded characters are present, causing the error message.

COMMAND.COM NOT FOUND In trying to load DOS, COMMAND.COM was not found on the specified drive.
Action: Insert a diskette with DOS on it and retry, or specify other drive.

COMMAND TOO LONG FOR INSERT Embedded insert commands are limited to a maximum length of 78 characters.
Action: Shorten the command so that it doesn't exceed the 78-character limit.

COULDN'T RELOAD MEM There isn't enough available memory to reload the indicated subroutine.
Action: See "Out of Memory."

COULDN'T SAVE OVERLAY When using RLSMEM, this message indicates that the overlay can't be saved to the designated disk.
Action: Make sure that you're addressing the right drive. Check to see that the proper diskette is in the drive and that it isn't write-protected or full. See that the drive door is closed. Retry the command.

DEFAULT FOR WHAT? The DEFAULT command does not recognize the desired default or default code.
Action: Make sure there is a comma but no spaces between commands. Refer to the pages on the Default command.

DISK ERROR A disk drive fault has been detected. This message will appear when the door to the drive is open, when there is no diskette in the drive, or when the diskette in the drive is write-protected or full.
Action: Correct the condition and retry.

DO YOU WISH TO QUIT (Y/N) This Prompt is given in response to Ctrl Alt Del.
Action: Press Y if you wish to quit, N otherwise.

DONE XyWrite response indicating that the command has been completed.

ERROR CLOSING INDEX FILE XyWrite was unable to close the index file.
Action: Insert another diskette and retry the TYPE or TYPEF or IX commands.

ERROR HANDLING INSERTS XyWrite could not 1) find a Save/Get in response to an Insert Save/Get format command, or 2) find a Data File in response to a Put Field command.
Action: 1) Make sure you have loaded the correct set of Save/Get keys, and are requesting the right key. 2) Make sure there is data in the Data File and that you have properly entered the Field Identification command in your Main File.

ERROR LOADING HELP FILE The Help files may not be in the current directory of the default disk, or there may not be enough memory to load them.
Action: Change the default drive, or move the Help files to the default drive. If there is insufficient memory, make more room by clearing Save/Gets (CLRSGT) or storing a file (and closing a window if more than one is open).

ERROR LOADING OVERLAY An error has occurred while reading an overlay into memory.
Action: Try reading the overlay from another drive. If this fails, it may mean that either your drive or your diskette is failing. Try to recover what you can from the diskette and reformat.

ERROR LOADING SGTS There isn't enough room for the Save/Get file you've requested, or a reading error has occurred while loading them.
Action: Check the directory to make sure that you have the correct diskette in the named drive. Try loading the Save/Gets from another drive. You may need to create more memory by storing one of the screens before attempting to load the file again.

ERROR ON OUTPUT XyWrite has encountered an error in trying to open a new file on the output diskette.
Action: Insert another diskette in the non-default drive and attempt to store before trying again.

ERROR READING COMMAND.COM An error was encountered in reading COMMAND.COM.
Action: Insert a different diskette with COMMAND.COM on it and retry. If the problem persists, save any open files and quit.

ERROR READING INDEX FILE An error occurred while reading the temporary index file.
Action: See Error Reading COMMAND.COM above.

ERROR SORTING INDEX FILE A disk error occurred while the index was being sorted.
Action: Use another diskette and retry the command.

ERROR WRITING FORMAT FILE An error occurred while writing a format file to the diskette.
Action: Direct the output to another drive or place another diskette in the named drive.

ERROR WRITING INDEX FILE XyWrite is unable to output the index file to the diskette.
Action: See Error Writing.

ERROR WRITING An error has occurred while writing the file to the diskette.
Action: Make sure the disk in the specified drive has enough room for the entire file and that it isn't write protected before trying SAVE again. Or insert a fresh disk into the non-default drive and try again (don't change the disk in the default drive, in case it has a BOTTOM.TMP or TOP.TMP file on it). Other possible problems include a bad spot on the disk, or a bad drive.

EXTRA START COMMAND An embedded command has an extra start character. They should always come in pairs—start character («) and end character (»).
Action: Find the unpaired start command. Either delete it or insert an end character at the appropriate place.

FILE ALREADY EXISTS You are attempting to give a NEW file the same name as an existing file.
Action: Rename the new file.

FILE EXISTS, OVERWRITE IT? You are attempting to store a document under a filename that already exists on the directory of the indicated drive.
Action: If you wish to replace the contents of the existing file with your current document, type Y. (Note: the data that was previously in the file will be lost.) If you don't want to overwrite the existing file, type N, then give the file another name.

FILE NOT FOUND The named file is not found on the diskette.
Action: Check the directory (DIR) for the correct filename to verify that you have the correct diskette in the named drive.

ILLEGAL COMMAND XyWrite does not recognize the command entered.
Action: Refer to the page in this manual for the correct form of the command.

ILLEGAL IN FORMS MODE You are not allowed to use Define functions (Save/Get, etc) while in forms mode. This protects you from accidentally altering the template.
Action: Leave Forms mode before attempting the define function.

ILLEGAL MARGIN The value given with the Margin command exceeds the limits allowed.
Action: Retry the command, observing the 255-character limit.

INDEX ITEM TOO LARGE One or more of the items marked for indexing is too large.
Action: The buffer for holding the index items can handle about 400 characters. Trim some of the text off the offending references and retry.

INSERT DISKETTE FOR DRIVE B: AND STRIKE ANY KEY WHEN READY This is a DOS prompt that may appear if you attempt to address drive B: on a one-drive system. XyWrite has no control over where this message may appear.
Action: This message affects only the display, not the data in your file. A Store or Abort command returns the screen to normal.

INSUF. MEMORY FOR SORTING XyWrite does not have enough memory to sort the index.
Action: Either shorten the index, or increase the memory by clearing the Save/Gets and closing the window you're not using.

INVALID FORMATTING COMMAND The formatting command is not recognized by XyWrite. The command in question is shown expanded with the double angle brackets in the text.
Action: You can delete the command or strike Ctrl F10 to go to expanded mode and correct it. If it occurs on output using TYPE or TYPEF, call the file to the screen and proceed as above.

LABEL NOT FOUND The Go To Label (GL) command does not have a matching label (LB) command.
Action: Make sure the label name exactly matches the name used in the GL statement (including spaces).

INSERT DISKETTE FOR DRIVE B: AND STRIKE ANY KEY WHEN READY This is a DOS prompt that may appear if you attempt to address drive B: on a one-drive system. XyWrite has no control over where this message may appear.
Action: This message affects only the display, not the data in your file. A Store or Abort command returns the screen to normal.

LEGAL ONLY WITH DOS 2.0 You cannot give DOS 2.0 commands to DOS 1.1

MISMATCHED OPERANDS This usually appears when you attempt to equate literals, booleans and numeric values in an XPL embedded command.
Action: Correct the keystroke program.

MISSING SEPARATOR The SEARCH (or CHANGE) command requires a separator character at each end of the search string.
Action: Insert a slash (/) at each end of the search string (unless the string contains a slash — then use any character not contained in the string).

MORE THAN 1 UNARY OP You cannot use the uppercase function and XOR functions in the same operation.
Action: Rewrite the program so that unary operations are run in succession.

NEED 1 OR 2 NUMBERS The Indent Paragraph (IP) command requires only two number values.
Action: Retry the command with two numbers.

NEED FILE NAME & SGT ID When loading a program (LDPM) on a Save/Get key, you must provide both the name of the Save/Get file and the key ID.
Action: Check the order in which you have entered the needed information. Refer to the page on LDPM in this manual.

NEED NAME TO SAVE DEFINED BLOCK You need to give a filename with the SAVEDEF command when saving a defined block from the screen.
Action: Provide the name of the file to which you want the block saved.

NEED NUMBER OR MODE ABR. A mode (MD) command has been given with a two-letter name that is not recognized by XyWrite.
Action: Look in this manual under Character Modes to find the correct name for the command you want.

NO COMMAND Either (1) no command has been entered on the Command Line, or (2) the first position is blank.
Action: Provide the command. If this message appears after you have entered a command, remove any word spaces before the command and continue.

NO DEFINE BLOCK You cannot copy, move or delete if no block has been defined.
Action: Define the block and try again.

NO END IF When programming, you must provide an "end if" statement for every "if" statement.
Action: Look for stray "if" statements, and complete them with "end ifs."

NO EQUAL SIGN A numerical expression must have an equal sign for proper calculation.
Action: Type an equal sign (=) after the expression and try again.

NO FILE OPEN You are trying to execute a command that requires a file be open on the screen.
Action: Enter the desired filename with the command, or open the desired file to the screen.

NO FILE TO RUN There is no file specified on the command line.
Action: Press F10 to move the cursor to the command line, and enter the name of the file you want to run.

NO INDEX FILE XyWrite was unable to open an index file.
Action: Retry with another diskette.

NO LEADER CHAR. — USING SPACE The Leader command was given with filler character specified. The command assumes you want a blank space for the filler character.
Action: If you want to use a filler character instead of blank spaces, delete the embedded triangle in the text and begin again.

NO MEMORY FOR NEW SCREEN There is not enough memory to create another screen.
Action: This generally means the file you have opened is too large. First Store the file. Then you can split the screen and re-open the file.

NO MEMORY FOR NUMBERS There is not enough memory to calculate the expression given on the Command Line.
Action: Store or Abort the open file(s) and retry.

NO PATH SPEC There is no path specified with the CHDIR, MKDIR or the RMDIR commands.
Action: Provide a proper path and retry the command.

NO SAVE GET There is nothing assigned to the indicated Save/Get.
Action: Use Alt F2 to review of the contents of the Save/Gets and verify where the desired information is located.

NO SAVE GETS TO SAVE This appears in response to the STSGT (Store Save/Get) command. There is currently nothing assigned to any of the Save/Get keys.
Action: Load or create the Save/Gets you want stored, and try again.

NO SPACE FOR INSERT XyWrite cannot add a specified Insert Save/Get to the text during printing because the insert is too large for the available buffer.
Action: Try storing all files and then try printing from disk (TYPE *filename*) from the insert point to the end of the file.

NO SPLIT SCREEN The current screen is the only one open.

NO SUCH FIELD When using Mail Merge, you have requested a field number that does not exist in the list file.
Action: Check the list file to make sure that you have the correct field number.

NO SUCH FILE The named file is not found by XyWrite.
Action: Check to see that the name is correct and that the file is in the directory of the diskette you have referenced.

NOT ENOUGH MEMORY FOR DOS There is not enough memory to load DOS.
Action: To get to DOS you must QUIT. You may want to get more memory if this is a frequent problem.

NOT FOUND The Search command has reached the end of the file and has not found the indicated string.
Action: Press Ctrl Home and re-execute the command, as you may have started past the point of the string in the file.

NUMBERS ONLY You must give number values for this command. You can't use letters here.
Action: Retry the command using numbers.

ONE NUMBER REQUIRED A single digit numeric value is required for this command.
Action: Retry the command using only one number.

ONLY 10 SHIFT KEYS ALLOWED Only 10 shift keys can be used in the Keyboard File.
Action: Limit the number of shift keys you use. See the Keyboard File section in this manual.

ONLY 20 STATES ALLOWED Only 20 keyboard states can be used in the Keyboard File.
Action: Limit the number of keyboard states you use. See the Keyboard File section in this manual.

POINT TO NUMBER The cursor must be on a number when using the cursor arithmetic capability Alt + and Alt -.

PRINTER ACTIVE, QUIT ANYWAY? When you quit XyWrite, any file that is currently being printed will be aborted as well.
Action: Press Y if you want to quit anyway, N if you'd rather wait and try again after the file has finished printing.

PRINTER ERROR XyWrite cannot output to the printer for any of several reasons.
Action: Check printer connections to the computer and electrical outlets. Make sure the printer is turned on, on-line (not on pause), has sufficient ribbon, and the paper is feeding correctly. Correct any problems and try again.

READ ERROR A disk error was encounterd in reading the named diskette.
Action: This generally means a problem with the diskette or drive. When you begin to experience this error with any frequency your drive may be failing. The short term remedy is to read from another drive if available. If this fails as well, recover what you can from the diskette and re-format it if possible.

RESTORE DEFINE You cannot change strings or define further blocks until you have released the currently defined block.
Action: Press F3 and try again.

RESULT TOO LARGE The calculated result of the arithmetic expression exceeds the 32 digit limit.
Action: Look for ways to calculate a smaller result.

SPECIFY MODE The Mode (MD) command has been given with no mode specification.
Action: Supply a two-letter abbreviation or a decimal number with the mode command. Refer to the Character Display Modes description in the reference section.

SPECIFY PROGRAM NAME You need to specify a program name with the DO command.
Action: Check to see that the name you give is on the diskette.

SPLIT TOO BIG The value given for splitting the screen horizontally or vertically exceeds the limits of the screen.
Action: Values for a horizontal split must be 21 or less; for a vertical split, 78 or less. Refer to the Windows section for specific information.

TERMINATE BATCH? (Y/N) This response is given if you press Ctrl Break while executing a RUN file. This allows you to terminate a RUN file before completion.
Action: Press Y if you want to end the RUN file, and N if you want it to continue.

TERMINATE TEXT INPUT You can't use the GO command while entering footnotes (FN), Running Headers, or an index.
Action: Press F3 to leave the text input window, then try GO again.

TEXT SAVED After closing a header, footer, index marker, or footnote, XyWrite responds with this message. You also get this response after saving a defined block with the F2 key.

TEXT WINDOW WITHIN DEFINE You can't open a window and input text for an index, footnote, running header, etc. while inside of a defined block.
Action: Press Alt F3 to release the define and try to open the window again. Or move the cursor to beyond the end of the block define area and input the text there.

TOO MANY CHARACTERS You are typing too fast for XyWrite to keep up. Normally the input of text presents no problem, but if you type ahead while executing longer commands, you can get ahead of XyWrite.
Action: Stop entering characters and wait until XyWrite has caught up. Otherwise, your keystrokes are being wasted.

TOO MANY LEVELS OF INSERTS XyWrite cannot handle more than ten levels of nested Insert defined blocks. In practice, this should not happen often.

TOO MANY PROGRAM CALLS The Program calls are nested too deeply.

TOO MANY VALUES You have exceeded the 21-tab maximum while using the Tab Set command.
Action: Retry the command with fewer tab settings.

TYPE "+" TO CONTINUE PRINT The printer pauses after each page—press "+" to continue printing.

TYPE A, N, S, OR Y The Change Verify command offers you the following options: A (Abort command), N (No change but continue the search), S (change this last one and Stop), and Y (Yes, make the change and continue searching).
Action: Type the letter for the option you want.

TYPE ANY CHAR TO CONT PRINT This prompt appears when you are printing a document on single sheets of paper, using a Pause command or the P option with the TYPE command.
Action: Put a fresh sheet of paper in the printer and press any letter key to continue.

TYPE Y OR N XyWrite is expecting Y for yes or N for no in response to a prompt.
Action: Give one or the other and continue.

UNEXPECTED ERROR If you get here, we both may be in trouble.
Action: Immediately save your file to minimize the risk of losing your work. QUIT XyWrite and reload. If you can reproduce the circumstances that caused this error, please report them to us.

UNIMPLEMENTED COMMAND XyWrite recognizes the command, but it has not been implemented in XyWrite. This means that the feature is not available in this version, but may be scheduled for future release in another version.

WORD TOO BIG You have entered a Search or Change command string that does not have the proper separators. Quite often this message is followed by BAD STRING as a further indication.
Action: Refer to the Editing chapter to find the correct format for the command you want.

6 STATE DEFINITIONS ALLOWED Only six state definitions are allowed in the Keyboard File.
Action: Limit your state definitions to six (see the section in this manual on Keyboard File.)

4 TOGGLE DEFINITIONS ALLOWED Only four state definitions are allowed in the Keyboard File.
Action: Limit your toggle definitions to four.

INTRO

The purpose of this section is to help you get familiar with User Programming. There are two different kinds of commands used here:

- Function Calls — These are two-letter mnemonics (for example, BC, XC, CC, CS, GT, SS) which represent keystrokes. Refer to Function Calls at the end of the Keyboard File section for a complete list of these Function Calls.
- Embedded Program Commands — These are commands which appear in the program listings surrounded by double-angle brackets. For example: «LB-A», «GL-A», «PVU». Refer to Advanced User Programming for a description of these commands.

Program #1

PURPOSE

This program allows you to insert the current date from a Save/Get into your document as a hard date. It's useful for time-stamping a file. (This program performs the same function as the TODAY command.)

ACTION

Creating the Hard Date Program.
To create the hard date program:

1. Create a program file called HARDDATE.PM:

   Type: [F5] nep harddate.pm [↵]

2. Press: [Scroll Lock] (to turn *on* Scroll Lock)

3. Type: [F5] new temp [F9]
   [F5] da Mmmm d, yyyy [F9]
   [F5] types [F9]
   [F1]
   [Ctrl] [→]
   [F1]
   [F2] D
   [F5] abort [F9]
   [F5] rmvscr [F9]
   [F5] abort [F9]

4. Press: [Scroll Lock] (to turn *off* Scroll Lock)

5. Type: [F5] store [↵]

6. Type: [F5] run harddate.pm [↵]

Result: Steps 1-5 create the program. Step 6 runs the program, which stores the current date in Save/Get key D. Press [Alt] D to insert the date into your text. To cause the program to run every time you start XyWrite, add the following command to your STARTUP.INT file:

```
BC run harddate.pm
```

Analysis. Each time you start XyWrite this program opens a new file called TEMP (temporary) for the sole purpose of loading the current date to Save/Get D. Once this is done, the program aborts TEMP. Thereafter, you can insert the current date into your document by pressing [Alt] D.

EZ-Macro Program #2

PURPOSE

A macro is a program you write that contains a sequence of keystrokes to do a specific task. EZ-Macro is an aid for creating and using macros. It allows you to create macros on-the-fly -- it opens a second window, enters the NEP command for you, and turns on Scroll Lock. The setup procedure is simple, and the operation even simpler: All you use is Alt M and Scroll Lock.

This section includes the following:

Setting up EZ-Macro
Using EZ-Macro
Program Listing

Requirements. As written here, EZ-Macro has the following requirements:

- It makes use of two of your Save/Gets: Alt M to hold the program and Alt U to hold an indicator (0 or 1). (You can move these to different Save/Get keys if you want.)
- Running EZ-Macro requires that at least one window be open. (If all nine windows are open, it does not work properly.)

ACTION

Setting Up EZ-Macro

Before you begin using EZ-Macro, you set up your disk as follows. Step 2 assigns EZ-Macro to a key so that you can use it quickly, any time you want.

1. Create the files MACRO-M.PM and MACRO-U.PM, recording the keystrokes shown in the Program Listing. Store these files on your XyWrite disk (if you use subdirectories, store them in the same subdirectory that contains STARTUP.INT).
2. Add to your STARTUP.INT file the two lines:

```
BC ldpm  macro-m.pm,m
BC ldpm  macro-u.pm,u
```

Result: Now every time STARTUP.INT is run, these files are loaded into Save/Get keys M and U.

Then go to the command line and type RUN STARTUP.INT to load these files into memory.

ACTION

Using EZ-Macro

To write a macro:

1. When you are typing along in a document and you decide you'd like to create a macro:

   Type: Alt M

2. Then type the keystrokes you want to record.

   Result: The program will accept *all* keystrokes — text, cursor movements, function keys, Save/Gets, and the running of other macros or programs.

3. Turn Scroll Lock OFF to make corrections, and then back on to continue recording.

4. To save the macro you just created:

   Press: Scroll Lock to turn Scroll Lock off.

   Type: Alt M

   If you forget to turn Scroll Lock OFF, you will get an @M in the text (the symbol for Alt M). To fix this, turn Scroll Lock OFF, backspace to erase the @M, and try Alt M again.

At this point your macro is stored in a file called EZ-Macro. The command RUN EZ-MACRO appears on the command line. To run the macro:

1. Move the cursor to where you want to start your macro.
2. Press: F9

To save the macro, see Notes #1 and #2.

NOTE #1

Saving Your Macro to a Key. Each time you start a new macro, the previous macro is erased. If you wish to save your macro, you can do so anytime prior to starting the next macro. This allows you to build a library of macros.

To save the macro to Save/Get key X:

Type: F5 `ldpm ez-macro,x` ↵

NOTE #2

Saving Your Macro to a File. Another way to keep your macro is to save it to a file. You then run the macro by typing RUN followed by the filename on the command line (the same way we used EZ-Macro).

To rename the file, call it to the display and save it under the new name (for example, AMUCK):

Type: F5 call ez-macro ↵
Type: F5 store amuck ↵

To run the macro:

Type: F5 run amuck ↵

Sometimes this method is more convenient than saving the macro to a Save/Get key for either of two reasons: 1) If you've filled up your set of Save/Get keys, or 2) You'd rather assign it to a filename that you're more likely to remember. After all, what good is a macro if you can't find it when you need it?

Program Listing. This is the program file for EZ-MACRO. The file is shown as you would actually view it in Expanded Display (with RM 78).

MACRO-M.PM This is the main program file for EZ- MACRO.

```
<LB-EZ-MACRO Program                    ><IF(<PVU>==1)><GL-FINISH><EI><GL-START>

<LB-START>SW nSW tLD LD LD LD LD LD LD LD LD LD LD LD LD LD LD LD LD LD LD XC
CS BC ES 1XC BC DEL EZ-MACROXC BC ES 0XC BC NEP EZ-MACROXC BC TYPE YOUR MACRO.
(Scroll Lock OFF for corrections) END: Scroll Lock OFF, ALT-MCC SS <SXU,1><EX>

<LB-FINISH>CS BC STOREXC BC RMVSCRXC BC RUN EZ-MACRO      Move the cursor where
you want.  Then PRESS F9 to RUN macro.CC <SXU,0>
```

MACRO-U.PM This file contains only the single character zero — the program changes it to a one for the duration that the macro is in process, and then back to zero.

Analysis. EZ-Macro works as follows:

« LB-EZ-MACRO Program » This label is an identifier — it appears in the Save/Get directory (when you press Alt F2). There is a reason for all of the spaces following the word "Program" — so that you will see the title "EZ-MACRO Program" in the Save/Get directory without any text following it to clutter the Save/Get directory. The subsequent IF statement test the parameter U. The Save/Get U changes state each time you press Alt M — U is initially 0 (zero). When you press Alt M to start a macro, the program changes U to 1. When you press Alt M to close a macro, the program returns U to 0. The value of U determines which branch the program takes — LB-START or LB-FINISH.

« LB-START » This routine opens the next available window, moves its top border down 19 lines, disables the error beep (ES 1), deletes any previously-created file EZ-MACRO, turns back on the error beep (ES 0), creates a new file called EZ-MACRO, puts up a message telling the user to begin typing the macro, and sets Save/Get U equal to 1. The reason we disable the error beep is because this prevents the DEL EZ-MACRO statement from beeping if the EZ-MACRO file we're deleting doesn't exist. (If you try to delete a file which doesn't exist, you get a beep.)

« LB-FINISH » This stores the EZ-MACRO file you just created, closes the window, puts up the command RUN EZ-MACRO, moves the cursor into the text area, and sets the value of Save/Get U to 0. Then to run the program, you just press F9 .

PURPOSE

This appendix describes a keyboard file called SUPER.KBD, which is included on your original XyWrite disk. This particular keyboard puts many convenient functions at your finger tips — functions that are not provided in the keyboard file IBM.KBD. These functions were derived from a study of the more frequent keystroke actions of a cross-section of word processing users.

We encourage you to load SUPER.KBD (in place of IBM.KBD) and try it out. Modify this file or the standard IBM.KBD file to suit your needs — it is presented here as only one of many possibilities for assigning keys. We are sure that you can improve upon SUPER.KBD for your particular use.

For instructions on changing Keyboard Files, see Keyboard File in Chapter 6.

ACTION

Loading the Super Keyboard

To load this keyboard:

Type: F5 ldkbd super.kbd ↵

Result: The keyboard is loaded in place of the previous keyboard.

NOTE #1

Linear Cursor. The cursor assignments have been changed to Linear Cursor motion. This means that when using the four cursor keys, the cursor is free to move beyond the right margin, and to move up and down without ever switching columns. If you enter text to the right of a carriage return symbol (←), XyWrite fills in the area to the left of the cursor with spaces.

NOTE #2

Change Default Drives. Ctrl A changes the default drive to A: and Ctrl B changes the default drive to B:.

NOTE #3 **Display the Current Directory.** Ctrl D displays the current directory of the default drive.

NOTE #4 **Call a File.** With a directory displayed, when you place the cursor on the name of the file you want to call, press Ctrl C. This calls the file.

NOTE #5 **Intermediate Save.** Ctrl S Saves the active file to the disk and returns you to your editing. (As a precaution, it is a good practice to save your work frequently, at least every ten minutes. Then if a power failure were to occur, you would lose only the changes you made since you last saved the file.)

NOTE #6 **Edit Next File.** You can edit a number of files in order. You do this as follows:

1. Call a group of files by using the wildcards (* or ?) as part of the filename (e.g., CA *.PRN). The first file that matches is displayed on the screen.
2. After viewing or making changes to this file, press Ctrl N. This displays a prompt that asks whether you want to save any changes or go to the next file that matches the file specification.

You can continue in this manner until all of the files have been called up.

NOTE #7 **Format Commands in the Text.** The following keys insert format commands directly into the text.

Ctrl H	Flush Left
Ctrl J	Flush Center
Ctrl K	Flush Right
Ctrl P	Page Break

NOTE #8 **Start a Footnote or Index Entry.** Ctrl F starts a Footnote and Ctrl X starts an Index entry. After entering the text, press F3 to close the definition.

NOTE #9 **Caps Lock is a Shifting Key.** The use of Caps Lock has been redefined to be a shifting key. You use it by holding down the Caps Lock key and pressing another key. We have assigned two convenient functions to this key: outline numbering and moving directly to windows.

NOTE #10 **Automatic Numbering and Outlines.** There are two keys that insert the formats for the automatic numbering counters.

Ctrl L (Legal Numbering) inserts at the beginning of a document the definition for legal paragraph numbering: 1.1.1.1.1.1.1

Ctrl O (Outline) inserts at the beginning of a document the definition for writing an outline: I A 1 a (1) (a)

Caps Lock # where # is 1-9 at the top of the keyboard selects the level of the outline or the paragraph numbering you want. For example, pressing Caps Lock 2 gives either outline level 2 or paragraph level 2 (depending on which you have defined).

If you are using legal numbering most of the time, you may want remove the automatic insertion of Tabs with Caps Lock #. To do this, call SUPER.KBD and look through the file until you see TABLE = CAPS. On the lines that start 3 =, 4 = through 10 = remove the Tab characters and the associated commas. Store the file and reload the keyboard file.

NOTE #11 **Express Windows.** Caps Lock in combination with the numeric keypad numbers 1-9, goes directly to the indicated window number. For instance Caps Lock 3 moves directly to window 3.

NOTE #12 **Closing Windows.** Ctrl R performs a Remove Screen (RS) command. In the window menu, the window is indicated NOT IN USE. It is good practice to close unused windows, as they take up about 8K of memory even though they have no files open in them. Making them NOT IN USE releases that memory to the other files you may have open.

Memory Resident Programs **Appendix F**

PURPOSE

Memory-resident programs are programs that are available for immediate use — you load them into memory and then access them directly from XyWrite (without going through DOS). Examples are Sidekick, Ready, ProKey and Lightning.

Memory-resident programs must be loaded *before* the program you intend to run them from, such as XyWrite III.

A few (very few) of these programs can run with XyWrite III without special accommodations. (Sidekick Version 1.5 or greater is such a program.)

However, other programs need special consideration by XyWrite. There are two options available to allow these programs to work with XyWrite; if your program is not listed here, try Option 1 first:

- **Option 1**- Use with Lightning, Ready, and Homebase, and other programs.
- **Option 2** - *Any* memory resident program can be run using this option. However, in doing so, you give up some of the XyWrite functions (such as Alt → to move by word).

NOTE

Memory Requirements. Keep in mind that XyWrite together with DOS requires 256 KB of memory. Whatever program you intend to use with XyWrite will increase this minimum memory requirement. Refer to the user manual of the program you intend to run to identify how much additional memory it needs.

OPTION 1

Setting Up XyWrite with XYKBD.COM.

Many programs can run with XyWrite using Option 1. Examples are: LIGHTNING by Borland International, READY by Living Videotext, Inc., and HOMEBASE by Amber System Inc.

Other programs may work with this option, even though not called out above. If a program does not work with Option 1, it should work with Option 2.

ACTION

Running with Option 1.

To run LIGHTNING with XyWrite, for example:

1. While in DOS before running LIGHTNING, insert the XyWrite Program Disk and type the following:

   A>xykbd↵

   This runs a small keyboard handler called XYKBD.COM.

2. Then run LIGHTNING:

   A>light↵

3. Now load XyWrite:

   A>editor↵

You are ready to go.

NOTE #1

Startup. Add the line XYKBD to your AUTOEXEC.BAT file to automatically load XYKBD.COM whenever you startup your machine.

OPTION 2 Setting Up XyWrite with KM=2.

Any program that poses special problems if run with Option 1 can be handled by the following procedure. Examples are SUPERKEY by Borland International, PROKEY by RoseSoft Inc., and MULTILINK by The Software Link, Inc.

In selecting Option 2 you can run these programs with XyWrite at a cost of some functionality in keyboard use. This means that not all of the functions that you find on the normal XyWrite III keyboard can be made available if you choose this option. A list of these changes is included in the file DOS.KBD.

The good news is that any memory resident programs can be run in this manner. All keyboard input takes place through the DOS BIOS keyboard software.

ACTION Running with Option 2.

To run SuperKey with XyWrite, do the following:

1. Before starting XyWrite, run SuperKey:

   A>KEY↵

   then start XyWrite:

   A>editor↵

2. Call the configuration file for your printer: (Epson printer in this example)

   CM ca 3EPSONFX.PRN

3. Add this line to the beginning of the file:

   KM=2

4. Store the file then load it:

   CM ldprn 3EPSONFX.PRN↵

5. Load the DOS keyboard table from the Program Disk (that compensates for some of the diminished functionality imposed by DOS):

   CM ldkbd DOS.KBD↵

   You are ready to go.

NOTE #2 **Differences with Option 2.** Read the comments at the beginning of the DOS.KBD file by calling it to the screen. You will find notes there outlining the differences with this keyboard table and the standard XyWrite keyboard table IBM.KBD.

A tip to learning this new keyboard: Load the LONG.HLP file and use the BY_KEY selection. Then sample the various keys that are different in DOS.KBD.

NOTE #2B **Returning to Normal XyWrite Keyboard Mode.** If you want to return XyWrite to its normal keyboard functionality, change the KM=2 to KM=1 in the printer file. Then load IBM.KBD to restore all of the key combinations.

NOTE #2C **Startup.** To have Option 2 load automatically, add the LDKBD DOS.KBD command to your STARTUP.INT file.

Processing Files for Electronic Mail Appendix G

PURPOSE

Electronic mail is a service that allows you to send and receive messages to and from other computers over normal telephone lines. Because XyWrite's files are stored as ASCII text, XyWrite makes an excellent editor for your electronic mail.

To prepare XyWrite files for use as electronic mail, you don't need to learn any new commands. All you have to do is load a special printer file called MAIL.PRN, create your message, and then "print" the message to a file (using TYPEF).

ACTION

Processing a File for Transmission.
To prepare a file to send to another computer via electronic mail:

1. Make sure you have a copy of the special Printer File MAIL.PRN on your working XyWrite diskette (or in your XyWrite subdirectory if you have a hard drive). This Printer File is provided on your original XyWrite disk.

2. Load MAIL.PRN into memory:

   Type: F5 ldprn mail.prn ↵

   Result: MAIL.PRN is now in effect. This Printer File contains the instructions to automatically convert all XyWrite line endings to hard carriage returns, as required by electronic mail services.

3. Open a new document in which to write your message. Let's call this document MEMO.

   Type: F5 new memo ↵

4. Enter the following format commands into the top of your document.

   Type: F5 pt 1 ↵
   Type: F5 lm 0 ↵
   Type: F5 rm 65 ↵

   Result: These commands override the defaults you have in effect and establish the correct margins and printer table for electronic files.

5. Write your message. You may use most of XyWrite's normal editing procedures but don't use any of the mode commands except Mode Normal («MDNM»).

6. Store your document.

7. Use the TYPEF command to create an output file that is correctly formatted for electronic mail:

   Type: F5 typef memo,mail ↵

   Result: You original message remains intact in the file MEMO and can be used as backup for your records. The file MAIL is ready to be transmitted.

NOTE **Reload Printer File.** Be sure to reload your normal Printer File when you have finished processing your messages for electronic mail. You can refer to STARTUP.INT if you aren't sure of the name of the Printer File you regularly use.

Index Update

Enclosed is an updated index for the *XyWrite III Plus Reference Guide.* This index integrates the change pages that were part of the XyWrite III Plus upgrade. We apologize for any inconvenience we caused you by delaying shipment of this index, and hope these correction pages will increase your enjoyment of XyWrite III Plus.

Index

Primary page references are in boldface.

A

Index

B

C

Index

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Z

Special Characters