SCRIPT/PC

IBM Software
for IBM Personal
Computers

IBM

Personal Computer
Productivity Series

SCRIPT/PC

Book 1 How to Use

Program by: *Robert N. Seidel*
Charles W. Gainey Jr.

First Edition (January, 1984)

About This Book

How to Use This Book

This book is the guide to the installation and use of SCRIPT/PC for the IBM Personal Computer.

The information in the three SCRIPT/PC books is organized as follows:

- The book you are now reading (Book 1), is designed to guide you in a step-by-step manner through examples of designing letters, invoices, or other types of documents. You can then use the examples as guides in preparing your first document. There are Practice Sessions to get you started if you are inexperienced in SCRIPT. There is a tutorial section that gives you the exact language to create labels, letters, and memos. This section also shows you how to set up a page for information placement. Of course, this section also shows you how to start using SCRIPT/PC.

- The second book (Book 2), is a quick reference that allows you to look up any SCRIPT/PC control word or Starter Set Generalized Markup Language (GML) tag. The control words and GML tags list all the options available. This book also contains information on heading levels and printer options.

- The last book (Book 3), is a complete reference section that describes in detail all the functions of

each SCRIPT/PC control word and GML tag. This book also contains numerous examples of how to use control words and GML tags.

Book 3 also contains the glossary for SCRIPT/PC.

Related Books

We assume that you are familiar with the operation of the IBM Personal Computer and the Disk Operating System (DOS). DOS is described in the following publications:

- IBM Personal Computer *Guide to Operations*
- IBM Personal Computer *Disk Operating System*

This product is a subset of the IBM Document Composition Facility (DCF); that is, SCRIPT/PC control words and GML tags operate in a similar way on the IBM Personal Computer or on a large computer system that uses the Document Composition Facility. For detailed knowledge at this level refer to the following publications:

- *IBM Document Composition Facility: User's Guide;* order number, SH20-9161

- *IBM Document Composition Facility Introduction to the Generalized Markup Language: Using the Starter Set;*order number, SH20-9186

For additional information on transferring files between personal computers or between a personal computer and a large host system, refer to the: IBM Personal Computer *Asynchronous Communications Support* publication.

To create a backup copy of this product, go to the IBM
Personal Computer *Guide to Operations* book and the
chapter describing DOS.

Contents

Chapter 1. What is SCRIPT/PC

Contents

About SCRIPT/PC

Welcome to the world of text processing on your IBM Personal Computer.

The product you have chosen is a powerful and versatile text formatter that will allow you to create professional-looking documents on your IBM Personal Computer. You will learn how to mark up your text as you write it, using the special SCRIPT/PC markup language and your choice of editors. Then you will use SCRIPT/PC to format your document and, if you wish, print it. It formats your original text input into an attractive, professional format with a minimum of effort on your part.

SCRIPT/PC is a subset of an IBM product called SCRIPT/VS, which is part of the IBM Document Composition Facility (DCF). DCF is a document composition system that runs on larger computer systems that have IBM VM or MVS operating systems installed. SCRIPT/PC was designed so that you can use the same text formatting commands at your IBM Personal Computer (IBM PC) as you do on the larger IBM systems. Also, the files you create can be transferred from your IBM PC to another IBM PC or larger computer system using telephone lines and a product such as the IBM Personal Computer Asynchronous Communications Support product.

If you are going to be sending your files to a larger computer system for processing by SCRIPT/VS instead of SCRIPT/PC, then you might find it useful to obtain the IBM Systems Reference Library document for DCF. Its title and order number are:

Document Composition Facility: User's Guide; order number; SH20-9161

What You Need to Use SCRIPT/PC

To use SCRIPT/PC, you need:

- An IBM Personal Computer or an IBM Personal
  Computer XT with at least 128KB memory. A
  larger amount of memory allows you to process
  larger documents, create extended libraries of tags,
  create an additional amount of symbols and so on.

- One single- or double-sided diskette drive.
  However, two diskette drives are recommended for
  better ease-of-use with the IBM PC.

    Note: If you have an IBM PC XT or IBM PC
    with a fixed disk, you may load SCRIPT/PC and
    your input files on the fixed disk for additional
    processing enhancement.

- One of the following:

    - The IBM Monochrome Display and the IBM
      Monochrome Display and Printer Adapter.

    - An 80-column color monitor with the IBM
      Color/Graphics Monitor Adapter.

- The IBM Disk Operating System (DOS) Version
  1.10 or higher.

- A blank diskette.

If you want to print SCRIPT/PC formatted output, you
will also need an IBM PC Graphics Printer, IBM PC
Matrix Printer or compatible printer. SCRIPT/PC is
designed to take full advantage of the IBM PC
Graphics Printer and its additional functions.

Where SCRIPT/PC Fits In

There are many ways to use SCRIPT/PC to get a final
professional-looking document. SCRIPT/PC will
process your input text and commands and arrange
(format) your text in a concise fashion. You have the
flexibility to:

- Format the input file and print the output file at
  your IBM PC.

- Format the input file and send the output file for
  printing to another IBM PC either by:

  - Sending the formatted file over telephone
    (communications) facilities or,

  - Sending the diskette that contains the formatted
    file.

- Send an original unformatted (input) file, using
  communications facilities or diskettes, to another
  IBM PC, where it can be formatted and printed.

- Send the original unformatted (input) file, using
  communication facilities, to a larger computer
  system and use SCRIPT/VS for formatting and
  printing.

In addition, you can create a file at a larger computer
system and then send the original unformatted file,
using communication facilities, to an IBM PC for
formatting and printing.

The following diagram is an example of where the
SCRIPT/PC product is used in the total context of a
text processing system. Note that SCRIPT/PC does
not contain an editor, but will format input files that
have been edited and proof read.

Editing, Formatting and Printing a Text File

Edit	Proof-Read	SCRIPT/PC	Local
Create Input File	Verify Spelling (PC)	Format Output (PC)	Print Output (PC)

Unformatted   Formatted

Remote

Another IBM Personal Computer → Print Output (PC)

Another IBM Personal Computer

SCRIPT/PC — Format Output (PC) — Remote Print Output (PC)

System 370 VM/SP or OS/MVS Host System

SCRIPT/VS — Format Output (DCF) — Remote Print Output (3800)
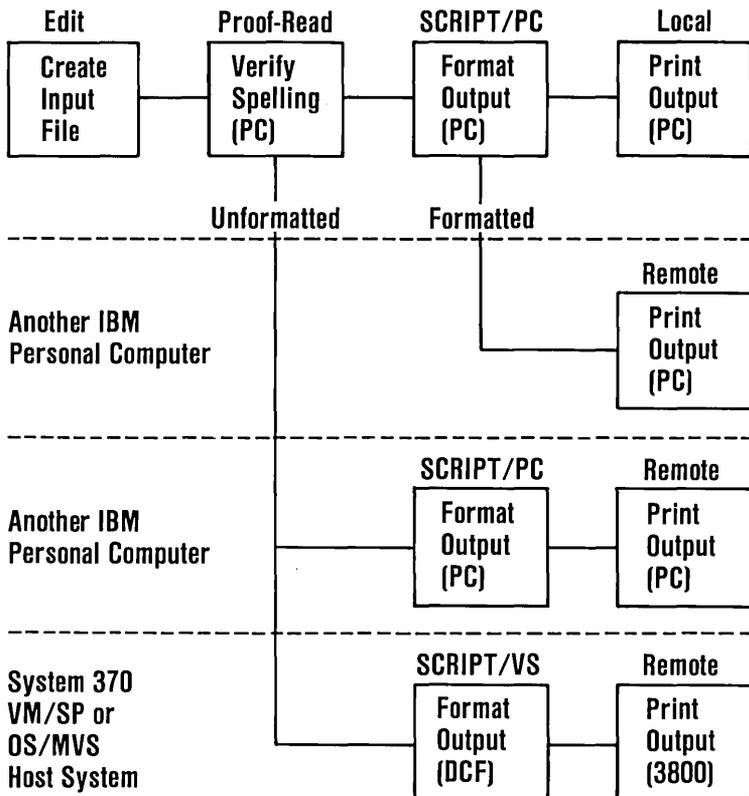
Where Input Text is Created

The input text (source) can be created using any editor that does not imbed special code into the file. Any editor that will create a text file will suffice. Examples of available IBM editors are the:

*   IBM Personal Computer DOS - EDLIN

    A full display screen editor will be more productive.

*   IBM Personal Computer Personal Editor

    If you create a file using this editor, the automatic setting of tabs causes formatting problems. You must use the NOTABS option when saving files.

*   IBM Personal Computer Professional Editor

You can proof read your own input file or, best of all, you can check your spelling with a product such as The IBM Personal Computer Word Proof.

> **Note:** The SCRIPT/PC books were developed with an IBM PC XT using the IBM PC Personal Editor and IBM PC Word Proof. Preliminary copies were processed on SCRIPT/PC for editing prior to sending the final files to SCRIPT/VS. The files sent to SCRIPT/VS were then processed to create the final copy for printing.

Input files can be created from a variety of sources: your own personal computer, another personal computer, or a larger system such as System/370. The input files can be brought to your own personal computer in one of two ways, by diskette or over the telephone lines (communications) by a product such as

the IBM Asynchronous Communications Support
product. Once the input file is in your personal
computer, no matter where it was created or how it
arrived, you are now ready to format it with
SCRIPT/PC.

Input for SCRIPT/PC

Your Personal Computer

```
┌─────────────────┐
│   Create        │
│   Input         │
│   File          │
└─────────────────┘
```

Another IBM
Personal
Computer

```
┌─────────────────┐
│   Create        │
│   Input         │
│   File          │
└─────────────────┘
```

SYSTEM/370

```
┌─────────────────┐
│   Create        │
│   Input         │
│   File          │
└─────────────────┘
```

SCRIPT/PC

- Read
  Input
  File
- Format
  Output
  File
- Print/
  Display
  Output
  File
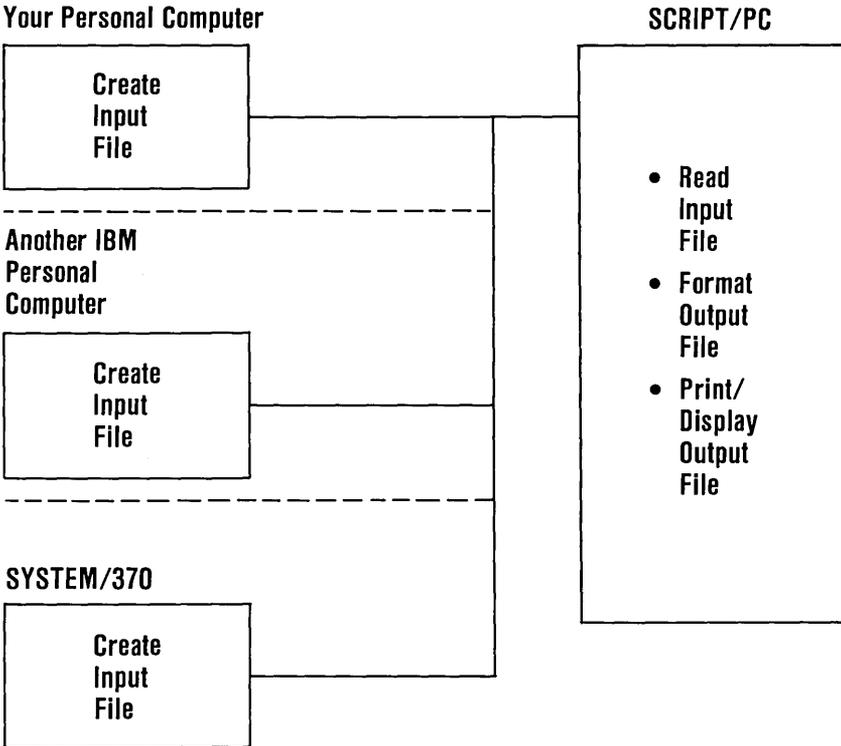
What is a Text Formatter

If you have done document preparation on personal computers before, you will probably have noticed that there are two different types of word or text processors available. One type is what is called the *single-step* word processor. In this type of system, both the editing (input and correction of commands and data) and the processing (actually formatting and creating the document) steps are done by one product, at the same time. Very often, you see the results of your commands immediately at the display. For example, if you tell this type of processor to center a line, you see the line centered immediately on the screen in front of you. This type of word processor is best suited for small documents and correspondence.

The second type of word or text processor is the *two-step* type. SCRIPT/PC fits into this latter category. This type separates the editor and formatting products, optimizing each product. You can use any available editor to create the input text. The following diagram shows your text processing system at your IBM PC.

Your IBM PC Text Processing System

```
    Edit            Proof-Read       SCRIPT/PC
┌──────────┐      ┌──────────┐     ┌──────────┐      ┌──────────┐
│ Create   │      │ Verify   │     │ Format   │      │ Print    │
│ Input    │──────│ Spelling │─────│ Output   │──────│ Output   │
│ File     │      │ (PC)     │     │ (PC)     │      │ (PC)     │
└──────────┘      └──────────┘     └──────────┘      └──────────┘

                                                     ┌──────────┐
                                                     │ Disk     │
                                                     │ Output   │
                                                     │ (PC)     │
                                                     └──────────┘
```
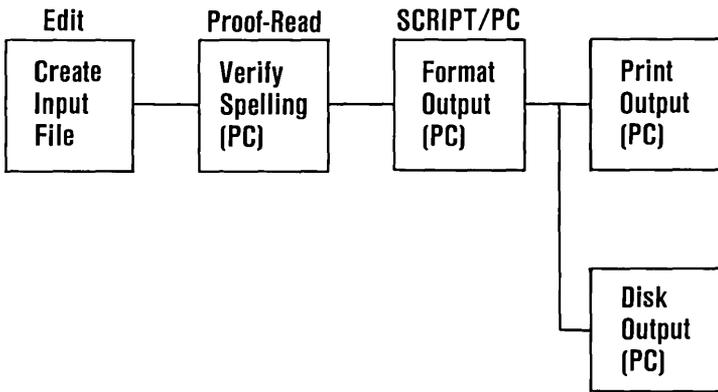
When you want to write a document or letter, one of
the jobs you have to do is to create the body of the
document itself, the words that express your thoughts
and ideas. No computer product, not even
SCRIPT/PC, can do this for you! But, once you know
what you want to say, you have to do the mechanical
task of actually writing or typing it, and formatting it
(laying it out) the way you want.

- If you want an item centered on a line, you have to
  calculate how many spaces should be before and
  after it, and then count them accurately.

- If you want an even right margin, instead of a
  ragged one, you will have to insert some extra
  blanks within each line.

- If you want to use page numbers, you must keep
  track of them yourself. Suppose you want to add a

new paragraph or move the paragraphs already typed into a new sequence. You might have to retype the document entirely just because the page numbers changed.

This is where SCRIPT/PC can help you. SCRIPT/PC will keep track of pages for you, and number them. It will put standard top and bottom titles on each page, fully control formatting and centering, and keep a Table of Contents if you wish. Paragraphs, lists, indentations, and so on are all automatically handled. You don't have to worry about formatting when you make changes to a document.

One advantage of letting SCRIPT/PC do the formatting for you is that it makes it so easy to change the format of your text. Because the input for your document is stored on diskette or fixed disk, you can change this input at any time, and each time SCRIPT/PC will print out a completely reformatted, professional-looking document for you! The following diagram shows some of the output formatting paths available to you.

Formatted Output Paths

```
SCRIPT/PC                                            Local
┌──────────┐                                      ┌──────────┐
│ Format   │         SCRIPT/PC Output             │ Print    │
│ Output   ├──────────────────────────────────── │ Output   │
│ (PC)     │                                      │ (PC)     │
└──────────┘                                      └──────────┘

              ┌──────────┐
              │ Disk     │
              │ Output   │
              │ (PC)     ├────┼──── (Communications)
              └──────────┘

─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ┼ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─
Another IBM                      │         ┼
Personal Computer                │         │

        SCRIPT/PC Output
        ┌──────────┐                  ┼
        │ -Diskette-│                  │  ┌──────────┐
        │    ○      │                     │ Print    │
        │    ◊ °    │ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ │ Output   │
        └──────────┘                     │ (PC)     │
                                         └──────────┘
```
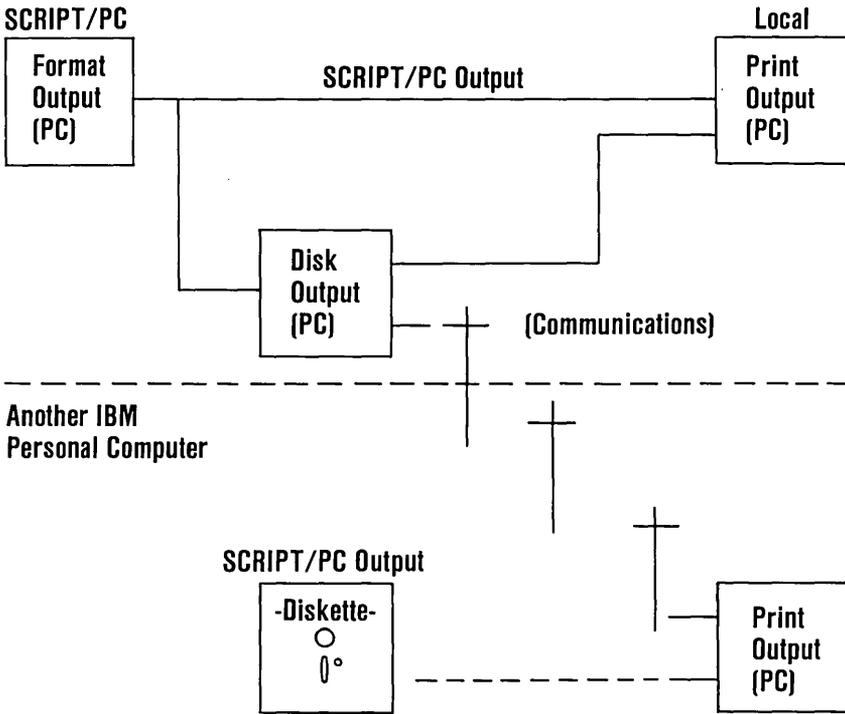
What Does SCRIPT/PC Do

Briefly, let's examine the main functions of
SCRIPT/PC that make document creation easy for
you.

- SCRIPT/PC will read your input, assemble it, and
  place the right number of words on each line. You

don't have to worry about it yourself. If you want, SCRIPT/PC will add extra blanks between words to create an even right margin.

- Similarly, SCRIPT/PC will keep track of how many lines are on the page, and skip to the next page (with appropriate page titles) automatically. You can specify that a page number appear in the top or bottom page title area.

- You can control indentation, as for lists or paragraphs.

- You can request that a Table of Contents or Index be produced.

- You can use SCRIPT/PC to insert the date or time anywhere in your document, or even the day of the month!

For those of you who are experienced text processing users, a number of additional functions are available; those of you who are not, soon will be in the category of experienced users. You will soon be proficient in marking up your raw text with commands to SCRIPT/PC (called control words and tags). SCRIPT/PC will interpret your markup and format your text.

You can choose to:

- Format your document with two columns per page, instead of one. This format looks especially attractive for reference manuals or glossaries.

- Create form letters. For example, you can prepare a list of names and addresses, along with a document. Then SCRIPT/PC can read through the list and create a unique letter or document for each entry.

- Customize your document by using the starter set of markup language tags that are provided or even create your own markup tags to assist you in custom tailoring your document. A tag library, also called a macro library, is supplied with SCRIPT/PC, which contains a number of useful starter set tag definitions. The tags in this library are similar to the Starter Set tags of the Generalized Markup Language (GML), that are part of the IBM Document Composition Facility (DCF) used in very large computer systems.

These features will be described in detail in the following chapters.

Compatibility with SCRIPT/VS

Letters and correspondence can be created and moved between your IBM Personal Computer and a IBM System/370 system using the same SCRIPT/PC markup language.

SCRIPT/PC is a compatible subset of SCRIPT/VS, which is part of the IBM Document Composition Facility (DCF) that runs on large IBM computer systems. You can create input files on your IBM Personal Computer and then format them, using either SCRIPT/PC or SCRIPT/VS. Input files can also be created at the large computer system and sent to your system for formatting and printing on SCRIPT/PC. The special formatting words you will be entering into your input file are recognizable on both SCRIPT systems.

Your IBM Personal Computer

```
EDITOR                              SCRIPT/PC        Local
┌──────────┐      Unformatted     ┌──────────┐   ┌──────────┐
│ Create   ├─────────────────────│ Format   │   │ Print    │
│ Input    │                      │ Output   ├───│ Output   │
│ File     ├─┐  ┼                 │ (PC)     │   │ (PC)     │
└──────────┘ │  │                 └──────────┘   └──────────┘
             │  │
             │  ┼
─ ─ ─ ─ ─ ─ ─ ┼ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─
IBM System/370  │
   VM/SP or
   OS/MVS
                          ┼    SCRIPT/VS       Remote
                          │  ┌──────────┐   ┌──────────┐
                          ├──│ Format   │   │ Print    │
              Input ─ ─ ─ ┤  │ Output   ├───│ Output   │
            ┌──────────┐  └──│ (DCF)    │   │ (3800)   │
            │-Diskette-│     └──────────┘   └──────────┘
            │    O     │
            │    0°    │
            └──────────┘
```
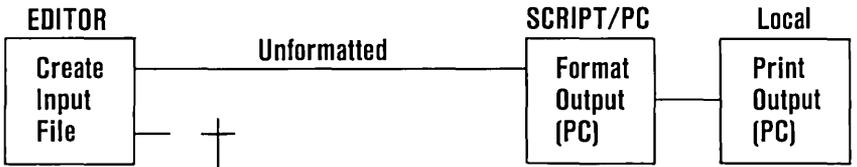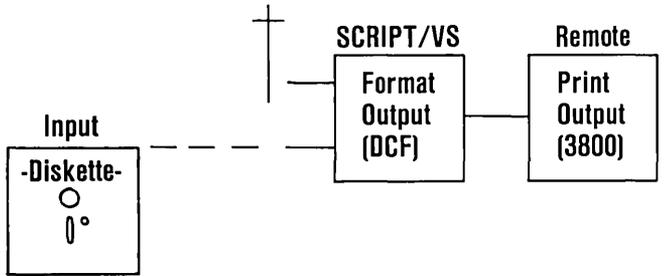
Before You Begin

The steps below ask you key questions about your
computer and the things you need to do before you can
use SCRIPT/PC. Start at step 1. If you don't know

how to do a step, or can't remember how, go to the indicated section and complete the step. Then return to this list.

When you reach the bottom of the list, you will be ready to start SCRIPT/PC.

```
1. Do you have everything you need?

   Yes   No   or   Don't Know
    |     |
    |     Go back to "What You Need to Use
    |     |          SCRIPT/PC"
    |     |
2. Do you know how to take care
   of your diskette?

   Yes   No   or   Don't Know
    |     |
    |     Go to Guide to Operations
    |     |
    |     |
3. Do you know how to make a
   backup copy of your diskette?

   Yes   No
    |     |
    |     Go to Guide to Operations
    |     |
    |─────┘    Make a backup copy
    |
    |
4. You are now ready to begin -
   Go to "Using SCRIPT/PC"
```

Chapter 2. Using SCRIPT/PC

Contents

Understanding SCRIPT/PC Files

I know you are anxious to get started, however, there are a few things you should learn before you begin.

SCRIPT/PC runs under the IBM Personal Computer Disk Operating System (DOS). Before you ask SCRIPT/PC to format a document for you, DOS must be up and running on your personal computer.

There are also a number of files in the IBM Personal Computer SCRIPT/PC that you should become familiar with, to the level of "what they do" not "how they do it."

The SCRIPT/PC diskette you received contains a number of files that are used to allow you to format your input source file(s):

File-id	Description
SCRIPT.EXE	*Main Program
SCRIPT.SBR	*Subroutines
PROFILE.SCP	*Program Options
SCRIPT.SCR	*Display Screens
SCRIPT.MSG	*Error Messages
GMLLIB.SCT	Library (GML Tags)
HEADDEF.SCT	Library (Headings)
NO-OP.SCT	No-Op commands
filename.ext	Your Input (Source)

Where * = Files that must be on the same disk or subdirectory.

All the files on the SCRIPT/PC diskette are described in the following list:

SCRIPT.EXE   Every time you start SCRIPT/PC, you will be starting the file called **SCRIPT.EXE** (the main program). This file processes your input text and commands and is the main program for this product. It will format your input files under control of a set of options stored in the file **PROFILE.SCP** or stored in a filename of your choosing.

SCRIPT.SBR   This file contains important external subroutines.

PROFILE.SCP This file is a complete set of options that have been created to allow you to use SCRIPT/PC for the first time, use

Practice Sessions, or to have an input file formatted and sent to the display screen for viewing and checking.

SCRIPT.SCR The display screen menus file contains all the files that create the display screen menus, options, and help facilities.

SCRIPT.MSG The error messages that SCRIPT/PC uses to signal to you that an error has been found are kept in this file.

GMLLIB.SCT This file is a library that contains the Starter Set of GML tags. You may create additional tags or change existing tags. They can be placed in the same library or in a new library that you can name. SCRIPT/PC assumes a file extension of **SCT** unless you specify your own file extension.

HEADDEF.SCT This file is a library that contains the coding necessary for the formatting of the heading control words (Heading 0 through Heading 6).

NO–OP.SCT This file contains the list of control words and tags that are recognized by SCRIPT/VS but not available with SCRIPT/PC. This file allows you to have the unavailable commands recognized, however, no operation will take place. These commands listed in this file will be ignored by SCRIPT/PC.

filename.ext This entry represents the input files created by you with an editor. These

input (source) files may be on the same
diskette that contains SCRIPT/PC, on
a separate diskette, or if you are using a
fixed disk, you may have your input
files in the same directory or
subdirectory on your fixed disk. If you
have an IBM PC with a fixed disk, the
performance of SCRIPT/PC is
enhanced if you read all the files from
your diskette into a directory or
subdirectory of your fixed disk.

You may select any extension name.
However, if you select the extension
name of **SCT**, you will not have to
enter the extension name when you are
selecting an input file.

The following files must always be located in the same
disk or subdirectory if it is a fixed disk:

1. SCRIPT.EXE
2. SCRIPT.SBR
3. SCRIPT.SCR
4. SCRIPT.MSG
5. PROFILE.SCP

You may enter SCRIPT/PC control words and tags in
either upper- or lowercase (unless specified otherwise).
For clarity, all control words and tags in this book will
be shown in UPPERCASE.

Before you ask SCRIPT/PC to format a file, you
should create the input file using your own editor, for
example, the IBM Personal Editor or the IBM
Professional Editor. SCRIPT/PC *does not* contain an
editor. Upon completion of editing the file, you may
proof read the file by yourself or use the IBM Personal

Computer Word Proof to assist you. When your spelling and editorial marks appear to be correct, you can format your file with SCRIPT/PC.

Understanding SCRIPT/PC Display Screens

All display screens were written to contain common information in the same screen location from screen to screen. You can always expect the same response to a similar action.

1. The display screens contain a screen name (subject), directions, option list and escape. All functions on the screen have been separated for visual clarity.

2. Keeping escape functions on every screen allows you to change your mind, even in the middle of a series of steps. You may return to the Main Menu at any time, even while your input file is being formatted. A help function has also been provided to assist you.

3. When a wrong answer is entered, a sound (beep) is generated to get your attention. You are then allowed to retype your answer and enter it again, without starting all over.

How to Begin

1 If you have an:

- IBM PC with one or two diskette drives:

  Place your DOS diskette in the left drive
  (drive **A**), if DOS and SCRIPT/PC are on
  separate disks. Go on to step 2.

- IBM PC with a fixed disk: Run procedures
  will change because of the large storage
  capacity of your fixed disk. DOS should
  already be resident on your fixed disk, if it is
  not:

  1. Place the DOS Diskette in drive **A**,

  2. Get the IBM PC *Guide to Operations* and

  3. Copy the diskette that contains DOS to
      the fixed disk.

  If DOS has been loaded prior to this time,
  continue with the next step.

2 Switch on your IBM Personal Computer (if off)
or do a *system reset* by holding down the **Ctrl** and
Alt keys and pressing the **Del** key. This will make
sure that DOS is loaded into your IBM PC.

3 SCRIPT/PC can provide automatic time and
date insertion, depending upon your markup of
the input file. When the computer prompts you
for a date, type the current date. For example:

    6-25-83

and then press the Enter (◄┘) key.

If the system does not prompt you for a date and you need the date to be entered, type:

   DATE

and then press the ◄┘ key. Then follow the previous example to insert the current date.

4   When the computer prompts you for a time, type the current time. For example:

   8:30

and then press the Enter (◄┘) key.

If the computer does not prompt you for a time and you need the time to be entered, type:

   TIME

and press the ◄┘ key. Then follow the previous example to insert the correct time.

> **Note:** If you haven't made a backup copy of the SCRIPT/PC diskette, you should do so now. Because your IBM Personal Computer may have one, two or more diskette drives, one or more fixed disks, and operate under DOS 1.10 or higher, you should go to the correct IBM Personal Computer *Guide to Operations* and the correct DOS book to create a backup copy of SCRIPT/PC. When you have completed backing up your SCRIPT/PC diskette, you may continue with the next step.

5   When the DOS prompt is visible on your screen, the IBM Personal Computer is ready for you to proceed:

> A>

> **Note:** The DOS prompt character will vary, depending upon which default drive you selected. For purposes of clarity, the DOS default drive in this book will always be identified as *drive A*.

6   **For a one-drive IBM PC,** remove your DOS diskette (if inserted) and place the SCRIPT/PC diskette in drive A. Go on to "Batch Mode."

For a two-drive IBM PC, you should:

1. Remove the DOS diskette from drive A.

2. Insert your SCRIPT/PC diskette into drive A.

3. Insert your input (source) diskette into drive B, the drive on the right.

    Note: If you have an IBM PC with a fixed disk, you should copy the files on the diskette into a directory or sub-directory. This will aid in ease-of-use as SCRIPT/PC will always be ready to be used.

SCRIPT/PC allows you to process a file two ways: in batch mode or interactive mode.

Batch Mode

The first processing method is called *batch mode* and is shown in the following format:

SCRIPT d:filename.ext [options–filename]

Where:

SCRIPT Specifies the name of the program you want to start (SCRIPT.EXE).

d: Is the disk drive specifier. This selects the disk where your input (source) file is located. SCRIPT/PC will assume the input file is on the DOS default disk if the disk drive specifier is omitted.

filename Specifies the name of the input (source) file you want SCRIPT/PC to format. The filename may be from one to eight characters in length.

.ext Specifies the name of the file extension. SCRIPT/PC will assume an extension of **.SCT** if the extension is omitted.

options– Specifies a permanent options list.
filename If you do not specify an options-filename the options-filename PROFILE.SCP is used.

If you already have an input file, (for example: ABC.SCT) and you know that all the options you want while SCRIPT/PC is formatting the file are correct, you will only need to type the filename after the word SCRIPT, as shown below:

```
A>SCRIPT ABC
```

and then press the ◄┘ key. The file **ABC.SCT** will be:

1. Located,

2. Processed (formatted), and

3. Placed on an output device.

SCRIPT/PC allows you to control a number of key items about processing your input file. The directions are stored in the file named **PROFILE.SCP.**

There is a standard set of options saved in a file called **PROFILE.SCP** (supplied on the SCRIPT/PC diskette). The profile can be changed to reflect your individual computer and your preferred processing procedures.

You can create your own alternative set of options under a separate filename. This allows you to set up a number of options-profiles to assist you in processing a variety of files. For example:

```
A>SCRIPT ABC ANOTHER
```

Where:

SCRIPT Specifies the name of the main program you want to run (SCRIPT.EXE).

ABC Specifies the name of the input (source) file you want to have SCRIPT/PC format (ABC.SCT).

ANOTHER Specifies the name of a profile options file (ANOTHER.SCP), that is different than the SCRIPT/PC supplied default (PROFILE.SCP).

2-12

Interactive Mode

The second method to run SCRIPT/PC is called
interactive mode. If you:

• Are running SCRIPT/PC for the first time,

• Want to review or change the options for running
SCRIPT/PC, or

• Want to practice using SCRIPT/PC and GML
markup words,

then, after the DOS prompt, type:

```
A>SCRIPT
```

and then press the ◄┘ key.

> **Note:** In this mode you will have the ability to
> change your profile options or to create one or more
> alternative profile options files.

SCRIPT/PC displays the IBM logo screen and then
presents the Main Menu where you are presented with
a choice of:

1. Formatting a document with selected options,

2. Practicing with SCRIPT control words or GML tags
or

3. Returning to DOS.

If you are unsure about how an individual control word
or tag works and whether it will give you the output you
want, use the Practice Session to see how SCRIPT/PC

will format your test input. If you keep Book 3 handy
during your session, you will be able to increase your
knowledge about a given control word or tag.

Using Your Keyboard

The keyboard has been set up to assist you in your
menu selection and practice sessions. The following
keyboard keys are active in both menu selection and
practice sessions:

←┘   In practice sessions Enter is used to signal
          the end of an input line (carriage return). In
          selecting options in the options profile, this
          key is used to advance the cursor to the
          next option.

←   Moves the cursor to the left on an input
          line.

→   Moves the cursor to the right on an input
          line.

↓   Moves the cursor forward or down one
          input field (top-to-bottom). It is not
          operative during practice sessions.

↑   Moves the cursor backward or up one input
          field (bottom-to-top). It is not operative
          during practice sessions.

End   Moves the cursor to the first blank
          character position at the end of the line.

Ctrl-End Erases from the current cursor position to
          the end of the line. These keys should be
          pressed at the same time.

Ctrl–Home This combination of keys is only available in Practice Sessions.

These keys will recall the last line entered, leaving the cursor at the end of the recalled line. You may now continue to edit the input line. Both keys should be pressed at the same time.

> **Note:** This key combination is *not* valid if any characters have been entered after a SCRIPT/PC prompt (=>).

Ins Allows insertion of another character at the current cursor position, moving all characters to the right of the cursor to the right. Pressing this key changes the cursor from a blinking underscore to a blinking box (Insert Mode). Pressing the key again removes the IBM PC from Insert Mode and returns the cursor to normal operation.

Del Removes (deletes) a character at the current cursor position, moving all characters on the right of the cursor to the left. The blank created at the end of the line by the delete action will be filled by a null character (as opposed to a space character). Delete is always available to be used and is not dependent upon the Insert Mode function.

How to Select Menu Options

The display screens are written to assist you in
completing the SCRIPT/PC task you are selecting, as
quickly and effortlessly as possible.

Main Menu

When you select SCRIPT/PC in Interactive Mode, the
IBM logo is displayed for a short time and you are given
control at the Main Menu. When you decide to either
format a document, practice, or return to DOS, type
your choice. SCRIPT/PC will start the operation you
selected or return to DOS.

```
                              Main Menu


                   Options


                   1.  Format a document

                   2.  Practice SCRIPT or GML

                   3.  Return to DOS


                        Choice  [ ]




   F1 Help
```

Main Menu Error

If you make a typing error, entering a choice that is not allowed, you will be reminded by a sound (beep) to enter a correct option number.

Format a Document

```
                              Format a Document

         Format input file-id . . . . . . . [              ]

         Options                              Choice

         Output device . . . . . . . . . . [P]
         Disk output file-id . . . . . . .  [              ]
         Error messages file-id  . . . . .  [              ]

         Passes  . . . . . . . . . . . . .  [1]
         Page number - Start . . . . . . .  [1  ]
         Page number - End . . . . . . . .  [999]

         Display while formatting  . . . .  [Y]
         Display screen pause  . . . . . .  [N]
         Printer pause . . . . . . . . . .  [N]

         Macro library file-id . . . . . .  [GMLLIB        ]
         Options profile file-id . . . . .  [PROFILE       ]

 *********************** Error Message Line ***************************

   F1 Help    F2 Save Options     F9 Main Menu     F10 Start     Esc Quit
```

Selecting a File-id for Formatting

The first entry area at the top of the screen is reserved
for the file-id of the input file you want to format. The
"Format input file-id" entry area will accept a file-id in
the following format:

d:filename.ext

Where

d: Specifies the disk drive if it is not the same as the disk drive where SCRIPT/PC is located. This is an optional item.

filename   Specifies the filename of the input file you want to format (1 to 8 characters).

.ext   Specifies the name of the file extension. SCRIPT/PC will assume an extension of **.SCT** if the extension is omitted.

Selecting Profile Options

If this is the first time you have used a text formatter, some of the options may be unfamiliar to you. The following explanations tell you what each response field will accept as a choice.

You can move the cursor to the response field you want by use of the Enter (←┘)key or the Cursor Down (↓) and Cursor Up (↑) keys. When the options list is filled out to your satisfaction, either use it as a temporary file (for this run only) by pressing the **F9** or **F10** key, or create a new permanent options file by pressing the **F2** key. If you want to ignore all your changes to the profile and return to the Main Menu, press **Esc** to quit.

> **Note:** You may select a new options filename. You would normally create a set of profiles that could tailor the processing of your formatted output using different options lists.

An explanation of the options list and possible responses follows:

1. **Output Device   [D], [P] or [S]**

Selects where you want your formatted output to be sent:

a. Disk - [D]

b. Printer - [P]

c. Display Screen only - [S]

2. **Disk output file-id [d:filename.ext]**

   If you are going to put your formatted output on disk, SCRIPT/PC will normally place that file on the default output disk under the filename inserted in this field and the SCRIPT/PC assumed extension of .SCO. However:

   - If you want the formatted output placed on the default output disk with a filename that is a duplicate of your input filename and an extension of .SCO, enter an ampersand (&) in this options field.

   - If you want a different output disk or a different extension, enter the corrected file-id in this options field.

   - If this option field is left blank, an error message is issued when SCRIPT/PC is started.

3. **Error message file-id [d:filename.ext]**

   SCRIPT/PC also allows you to create a permanent file for all your *error messages*. You may want to examine this file after completing a preliminary formatting run, as it is a permanent record of where errors occurred in your document. You can analyze and correct the input file at a later time.

   Your errors are normally stored on the default output disk under the filename inserted in this field and the SCRIPT/PC assumed extension of .SCE. However:

- If you want the error messages placed on the default output disk with a filename that is a duplicate of your input filename and an extension of .SCE, enter an ampersand (&) in this options field.

- If you want a different output disk or a different extension, enter the corrected file-id in this options field.

- If this option field is left blank, no error message file is created.

4. **Passes   [1] or [2]**

The only valid replies to the **pass** option are **1** and **2**. Use the **2**-pass option for all documents that have a table of contents at the front of the document and documents that contain figures using figure references. For all other formatting applications, use the **1**-pass option.

If you select **2**-pass, SCRIPT/PC will take a longer period of time to format a file than if you had selected the **1**-pass option. On its first pass through your file, SCRIPT/PC will display the page number it is processing. The formatted output will be available for display only during the second pass through your input file. If you should respond to a SCRIPT/PC control word during processing of your file, the prompts for the response will be presented only on the first pass through the file.

5. **Printing Pages**

The *starting page number* and *ending page number* allow you to print or display a partial file. For example, if you have a 10 page document and you make changes to pages 8 through 10, you can ask for only pages 8 through 10 when you reformat the file. You do not need to reprint or redisplay the total output file.

Note: If your changes are extensive, there is a chance your page numbers will ripple to the end of the chapter or book and partial printing of the document can cause you to lose information.

- **Page number – start   [nnn]**

  The smallest page number that can be entered is 1. This number must be smaller than the ending page number or an error message is issued.

- **Page number – end   [nnn]**

  The largest page number that can be entered is 999. This number must be larger than the starting page number or an error message is issued.

6. **Display while formatting   [Y] or [N]**

   You also have the ability to display your text as it is being formatted. This allows you to monitor what is being printed, while it is printed or what is being sent to disk, as it is written on the disk.

7. **Display screen pause   [Y] or [N]**

   This option allows you to examine your formatted output when the *display screen* is full.

8. **Printer pause   [Y] or [N]**

   This option allows you to examine your formatted output when the *printer page* is full.

9. **Macro library file–id [d:filename.ext]**

   The macro library file-id is the name of the library the contains the tags provided by SCRIPT/PC, tags you have created or control word headings. The Starter Set of tags library name provided by SCRIPT/PC is **GMLLIB.SCT**.

The filename you select *should not* be the same as a normal input filename, but should be either the name of a valid tag (macro) library or left blank. SCRIPT/PC assumes a file extension of **SCT**, however, you may change the extension if you wish. You will only need a library if you are using Starter Set tags, your own tags or control word headings (HEADDEF.SCT) in a document.

If your library is on a different disk than SCRIPT/PC, a drive specifier (d:) must be entered.

10. **Options profile file-id [d:filename.ext]**

The final option is the file-id of the profile options file. You may replace your normal options list by keeping the filename **PROFILE.SCP**. If you want an alternate profile to use, type a different filename with SCRIPT/PC assuming an extension of **SCP** or type your own extension, and then press the **F2** key. There will be a permanent file under the new file-id for your use.

Selecting an Options Profile

The options displayed are read from the options profile file selected when SCRIPT/PC is started; the default file-id is PROFILE.SCP.

To update a particular profile, select the profile name to be updated when SCRIPT/PC is started. The correct profile will be selected by using an input file-id of **NUL** followed by the options file-id that you want to change. For example:

```
SCRIPT NUL ANOTHER
```

To make changes to the selected options profile, select "Format a Document" by pressing the **1** key, and then make your updates. The keys described at the bottom of the display screen give you a variety of options:

F1   Help

> This key selects the Help facility that explains the options.

F2   Save Options

> This key allows you to make permanent changes to the file identified in the option field of the **Options profile file-id**. This is useful in creating a number of profiles that you can use in Batch Mode.
>
> If you saved a profile that had the option **Output device ... [P]** selected and it was saved in a file-id of **PRINT.SCP**, it would then be easy to use Batch Mode and print the output by typing:

```
SCRIPT ABC PRINT
```

> In addition, if you created an options profile named **DISK.SCP** that selected the disk for formatted output, it would be easy to format the file and save the output on disk by typing:

```
SCRIPT ABC DISK
```

> The file on disk can be transported to another IBM PC or stored for future printing. For example, DOS 2.00 contains a PRINT option which will print the formatted output for you.

F9   Main Menu

> This key allows you to make a temporary change to the profile and use it for only this session. It will not be stored on disk.

F10   Start

> This key starts SCRIPT/PC, using the options displayed on the screen when the key is pressed.

The file that is formatted is in the **Format input file–id** field. This key starts the formatting operation.

Esc Quit - This key returns you to the Main Menu *without* changing the profile. It will remain as it was when you selected the **Format a Document** display screen. However, if the **F2** key has previously been pressed, the options profile was permanently stored with any changes that were made since selection of **Format a Document.**

Format a Document Error

When an error is detected, an error message is displayed and the cursor is positioned at the start of the option that is incorrect. Type a correct answer.

If SCRIPT/PC cannot find a file it gives you a choice of:

1. Inserting a diskette that contains the missing file and trying to find the file again before continuing by pressing the **F10** key.

2. Canceling the processing by pressing the **Esc** key.

Practice SCRIPT or GML

If you want to do a practice session, then select choice 2 on the Main Menu (Practice SCRIPT or GML).

```
┌─────────────────────────────────────────────────────────────┐
│                    Practice SCRIPT or GML                     │
│                                                               │
│          Enter commands when you see the prompt    (=>).      │
│                                                               │
│  =>                                                           │
│                                                               │
│                                                               │
│                                                               │
│                                                               │
│                                                               │
│                                                               │
│                                                               │
│                                                               │
│                                                               │
│  Esc  Quit                                      F10  Pause    │
└─────────────────────────────────────────────────────────────┘
```

Practice SCRIPT or GML Error

When an error is detected during your practice session,
it is displayed on the screen at the moment it is
detected. The error messages are also being recorded in
an error message file named **PRACTICE.SCE** if the
Error message file-id field in the options profile has
been set to the ampersand character (&). If this field in
the options profile is blank, no error messages will be
permanently recorded.

> **Note:** When the first SCRIPT/PC command is
> entered in practice mode, there will be one jump of
> a few lines for the prompt symbol (=>). This is
> normal and will appear only on the first page of the
> practice session. These are the blank lines that
> normally appear at the top of your "page."

Now that you know how to get SCRIPT/PC started,
you may begin to use the Practice Sessions in the next
chapter to gain an understanding of the control words
and tags, or if you are adventurous, you may wish to
create a permanent file of real work using an editor
(not supplied with SCRIPT/PC). The chapters after
the practice session give you tips and hints about how
you might wish to markup your document, set up your
pages, and then test and print your document.

Chapter 3. How to Use the Practice Sessions

Contents

Introduction

This chapter describes the SCRIPT/PC Practice
Sessions feature. This feature lets you enter text and
SCRIPT/PC commands from your keyboard, which
simulates SCRIPT/PC reading an input file. The
commands for SCRIPT/PC are arranged in two
categories, descriptive and declarative.

Types of Commands

Descriptive - Control Words

These commands tell SCRIPT/PC exactly how to build
your formatted document, somewhat like a detailed list
of all the materials needed to build a house and the
exact instructions "how to do it." All commands used
by SCRIPT/PC that are *descriptive* are called control
words and start with a period (.).

Declarative - Tags

These commands tell SCRIPT/PC to begin or end
certain pieces of your document, like a paragraph, a list,
or a heading. This is similar to telling a building
contractor to: build a wall here, put a door there, and
cut out a window over here. The details are left for
someone else to do. You do not have as much control
over this method. However, it is usually easier to use.
All commands used by SCRIPT/PC that are declarative
are called tags and start with a colon (:).

You will be learning to use both *descriptive* and
declarative types of commands. There sometimes may
be a need for both in the same document, depending
upon how you design your document. However, you
should not use both types of commands on the same
input line.

This practice session has been written for control words (the descriptive commands) to let you establish a good idea of "how to exactly build a house." Once you know how to build it, you can then "design a house" (using the declarative tags) without the roof caving in. It is a good way to begin to get acquainted with the various control words that make up the SCRIPT/PC language. You do not need a separate editor for Practice Sessions as you do for creating permanent input files; however, if you interrupt your Practice Session and then start again at a point other than the beginning, items such as line length or other default values may cause unpredictable results.

You are going to input text and the controlling markup control words from your keyboard. Practice Sessions do *not* create permanent input files; you must use an editor to create permanent files.

Getting Started

Practice sessions are selected from the Main Menu:

1 To start to use SCRIPT/PC, type:

```
SCRIPT
```

and press the ◄─┘ key.

2 When the Main Menu appears type:

2

3 After a short wait, SCRIPT/PC places you in Practice Sessions and you see the following display screen:

```
                        PRACTICE SCRIPT OR GML

                Enter commands when you see the prompt  (=>).

  =>

  Esc   Quit                                          F10   Pause
```

You can now start to become familiar with
SCRIPT/PC. Practice Sessions will accept all
descriptive and *declarative (tag)* commands.

1. You will be prompted to enter input text or control
   words by the arrow symbol (=>), followed by the
   cursor. All of your expected entries (input) will be
   shown with this prompt symbol. SCRIPT/PC
   output responses to your input will not have the
   prompt symbol. Practice Sessions allows you to
   simulate SCRIPT/PC reading text and control
   words from an input file. Instead of SCRIPT/PC
   reading your text and commands from a input file
   on a disk, it will be reading your text and commands
   from your personal computer keyboard.

2. After you respond to a prompt, press the ◄─┘ key.

3. Either SCRIPT/PC will output a line of text or the
   prompt will reappear. This is entirely dependent

upon the text or commands you entered. Every time you press the ◄─┘ key, SCRIPT/PC will decide how to process the line you just typed.

Now let's begin using the practice session feature!

Learning How to Format a Letter

This practice session consists of entering descriptive control words to write the following letter. Once you have entered the control words and see how SCRIPT/PC processes them, you will be better able to compose your own letters, enter them into an input file (using an editor), process them with SCRIPT/PC, and then print the output file.

Do not enter any information until you are told to do so.

Sample Letter 1 Output

```
Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345


Dear Mary,

This is to inform you that you have been
selected  as  a qualified  candidate for
admission to this school.

A complete copy of  our  catalog,  along
with financial assistance forms, will be
sent to you in a few days.

Respectfully yours,


Jonathan Wise
Dean of Admissions
```

The special SCRIPT/PC control words and text that created the letter are shown on the next page. Do not enter any information until told to do so. For explanation purposes in this book, assume that all line lengths (left to right margin) have been set to 40 characters.

Sample Letter 1 Input

```
.SP 2
.FO OFF
Mary Jones
1234 Pickford Avenue
Great Barrington, MA  12345
.SP 2
Dear Mary,
.SP
.FO ON
This is to inform you that you have been
selected as a qualified candidate for
admission to this school.
.SP
A complete copy of our catalog, along with
financial assistance forms, will be sent
to you in a few days.
.SP
.FO OFF
Respectfully yours,
.SP 4
Jonathan Wise
Dean of Admissions
```

By examining the original input (source) file, you will notice a few items that may seem different at first glance, the markup words. However, once you begin to understand the special words, called control words, the differences will disappear.

The Inside Address

Let's look at the first few lines, the inside address:

```
→ .SP 2
   .FO OFF
   Mary Jones
   1234 Pickford Avenue
   Great Barrington, MA  12345
```

The first line (pointed to by →) is the SCRIPT/PC control word that created the space at the top of the example letter, in this case, two lines of space. The Space control word is written as **SP**.

All control words must be written to follow certain rules, also known as syntax. Control words in SCRIPT/PC:

1. Are two characters long.

2. *Must* start with a period (For example, .SP 2).

3. Must also be at the beginning of an input line.

    An input line can start in column 1 or it can start after a semicolon (;), which is called a *control word separator.*

The value after the control word directs SCRIPT/PC to move the paper vertically (up) a specific number of lines. The display screen will space the same amount of lines as the printer. Let's try it!

1   Type **.BR**

and then press the ◄┘ key.

This control word forces SCRIPT/PC to start on a new line and completes the setup of your Practice Session.

> **Note:** When the first SCRIPT/PC command is entered in practice session mode, there will be one jump of a few lines for your prompt symbol (=>). This is normal and will appear

only on the first "page" of the practice
session. These are the blank lines that
normally appear at the top of your "page."

You are now ready to start practicing with the
letter.

2   Type **.SP 2**

and then press the ◄─┘ key.

Notice that SCRIPT/PC responds to your entry
and creates 2 blank lines. It will do the same
thing when it finds the Space control word in
your input files. Notice that when the control
word was processed, the prompt symbol was
moved down 2 lines to prompt for your next
entry:

```
=> .SP 2


=>
```

Looking back at our source file, you will see
that the next control word is **.FO OFF**, the
format control word. **FO** tells the SCRIPT/PC
formatter to either format your file under
automatic control (stringing your input lines
together) or to start copying input lines *exactly*
as you entered them. Since the name and
address should show on the letter exactly in the
format shown, formatting should be turned off
by use of the format control word, **.FO.** In this
case, the item we enter is not a value but a
direction, **OFF.**

3   Type **.FO OFF**

and then press the ◄─┘ key.

When you enter this control word you will notice
that there is no visible action taken by
SCRIPT/PC.

```
=> .FO OFF
=>
```

You are now ready to enter the name and
address fields.

4   Type **Mary Jones**

and then press the ◄─┘ key.

Type **1234 Pickford Avenue**

and then press the ◄─┘ key.

Type **Great Barrington, MA 12345**

and then press the ◄─┘ key.

As you finish each line, SCRIPT/PC will
generate an output line on the screen as follows:

```
=> Mary Jones
Mary Jones
=> 1234 Pickford Avenue
1234 Pickford Avenue
=> Great Barrington, MA  12345
Great Barrington, MA  12345
=>
```

Every input line you entered was copied by
SCRIPT/PC and an output line was created to
match your typed input text.

Now that you have used SCRIPT/PC with
formatting off, let's look at the name and
address fields as they will be created if the input
lines are strung together, also called
concatenation.

5   First, turn formatting on:

Type **.FO ON**

and then press the ◄┘ key.

Now set the horizontal line length to 40
characters, SCRIPT/PC had selected 72
characters as its default:

Type **.LL 40**

and then press the ◄┘ key.

This will cause the SCRIPT/PC to gather up
input text lines and create an output line 40
characters long. SCRIPT/PC will not break
(hyphenate) within a word. Your display screen
appears as follows:

```
=> .FO ON
=> .LL 40
=>
```

You are now going to enter the same name and
address fields you did in the previous step, but
you will notice that each input line will not
create an output line. SCRIPT/PC will gather up
your input text until 40 characters have been
accumulated or until it is told to quit gathering
text.

6   Type **Mary Jones**

and then press the ◄┘ key.

Type **1234 Pickford Avenue**

and then press the ◄┘ key.

Type **Great Barrington, MA 12345**

and then press the ◄─┘ key.

Your display screen is as follows:

```
=> Mary Jones
=> 1234 Pickford Avenue
=> Great Barrington, MA   12345
Mary Jones   1234 Pickford  Avenue   Great
=>
```

SCRIPT/PC started stringing characters together until the value of the line length was reached. Therefore, **Barrington, MA 12345** did not appear on the screen. The line length automatically set by SCRIPT/PC is 72. The line length for these examples is set to 40 to better illustrate the results of stringing together of the input lines.

7   To force SCRIPT/PC to output the partial line of text:

Type **.SP 2**

and then press the ◄─┘ key.

The display screen should be as follows:

```
Mary Jones   1234 Pickford  Avenue   Great
=> .SP 2
Barrington, MA   12345


=>
```

> **Note:** You will notice that extra spaces have been inserted into the output line to fill out the line length of 40. This will be done on all full output lines and is called *justification*. Justification can be turned off, if necessary, by another form of the format control word (**.FO LEFT**).

Since this format is not appropriate for all input, be sure to turn formatting off, **.FO OFF,** when you want your input text to be left exactly as you type it. Stringing input lines together (called concatenation), as formatting does, is needed to create *paragraphs*. When you want to stop automatic concatenation, you must use a control word that causes a formatting break, such as Space (**.SP**), or Break (**.BR**).

The concept of stopping and starting automatic formatting is very important to understanding SCRIPT/PC. If you want, continue to practice with the example letter until you feel comfortable with turning formatting on and off.

Salutation

1   To continue, be sure formatting is off (.FO OFF) before the salutation is entered.

Type Dear Mary,

and press the ← key.

After the salutation is complete and just prior to the *body* of your letter, formatting should be turned on.

2   Type **.FO ON**

and press the ← key.

Body

1   Type the following, pressing the ◄─┘ key at the
end of every input line:

```
This is to inform you that
you have been selected
as a qualified candidate
for admission to
this school.
.SP
```

Your screen should appear as follows:

```
=> .FO ON
=> This is to inform you that
=> you have been selected
This is to inform you that you have been
=> as a qualified candidate
=> for admission to
selected as  a qualified  candidate  for
=> this school.
=> .SP
admission to this school.

=>
```

Once again, every time SCRIPT/PC
accumulates 40 characters in a string, it creates
an output line.

The Space control word caused the partial
output line to be created. Remember that the
Space control word was one of the commands
that caused a formatting change (break). If a
value is omitted for the Space control word, the
value 1 is assumed by SCRIPT/PC.

2   Continue to enter the second paragraph by
typing:

```
A complete copy of our catalog,
along with financial assistance forms,
will be sent to you in a few days.
.SP
```

The display should appear as follows:

```
=> A complete copy of our catalog,
=> along with financial assistance forms,
A complete  copy of  our catalog,  along
=> will be sent to you in a few days.
with financial assistance forms, will be
=> .SP
sent to you in a few days.

=>
```

Closing

After the second paragraph is entered, formatting should be turned off to enter the closing.

Type **.FO OFF**

and finish the closing by typing the following:

```
Respectfully yours,
.SP
Jonathan Wise
Dean of Admissions
```

The display should appear as follows:

```
=> Respectfully yours,
Respectfully yours,
=> .SP 4



=> Jonathan Wise
Jonathan Wise
=> Dean of Admissions
Dean of Admissions
```

Once you have practiced using the control words
(those starting with a period), and understand them,
you have the ability to start to write your own
letters. If you feel comfortable enough to write
your own letter now, continue with the next step. If
you want to learn more right now, go to
"Formatting Additional Letters" instead of Quitting
Practice Sessions."

Quitting Practice Sessions

Exit the Practice Sessions by pressing the **Esc** key and
answering the next prompt by typing **Y.**

This is *only* a practice session. You will not created a
permanent file when you are in this mode. To write an
input file, you must use an editor such as the IBM
Personal Editor or the IBM Professional Editor to enter
the information necessary to create either the example
letter given in this chapter or one of your own.

Formatting Additional Letters

Now that you are familiar with writing a basic letter,
let's look at some additional functions you might want
to control. The explanations are followed by additional
practice sessions.

Indentation

When writing (typing) a letter, one of the most
commonly used functions is controlling the amount of
indentation. Indentation is normally used for starting
paragraphs, placing information in a column and,
generally, creating a letter that is visually pleasing as
well as informative. Instead of counting spaces and
then moving a typing element to the correct starting
position, SCRIPT/PC provides you with a number of

control words that easily do the work for you. There are 5 control words that are important for controlling information:

- Center one or more lines (.CE).
- Indent only one output line (.IL).
- Offset (moving to the left) the first line and indenting all the following lines to the right (.OF).
- Indent all output lines (.IN).
- Stop indenting for only one output line (.UN).

The following letter will be used as our example for this topic. You will notice that it is a variation of Sample Letter 1.

Sample Letter 2 Output

```
        College of Business Excellence
             Collegetown, New York

Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345


Dear Mary,

     This is to inform you that you have
been selected as  a qualified  candidate
for admission to this school.

     A  complete  copy  of  our catalog,
along with  financial assistance  forms,
will be sent to you in a few days.

                    Respectfully yours,




                    Jonathan Wise
                    Dean of Admissions
```

Centering a Line

One of the most time-consuming things to do when
typing a letter on a typewriter is centering a line.
SCRIPT/PC makes this function very convenient.
SCRIPT/PC allows you to center one or more lines by
use of the Center control word, **.CE.**

1   Let's create the new letter. Type the heading of
the letter as shown in the following example:

```
.CE ON
College of Business Excellence
Collegetown, New York
.CE OFF
.FO OFF
```

The output on the display should be:

```
=> .CE ON
=> College of Business Excellence
       College of Business Excellence
=> Collegetown, New York
           Collegetown, New York
=> .CE OFF
=> .FO OFF
```

If these control words and text were in a
permanent input file they would be printed as:

```
       College of Business Excellence
           Collegetown, New York
```

2 If you have only one line to center, the centering
control word can be entered on the same line as
your text. For example:

```
=> .CE College of Business Excellence
```

will center the one line of text.

Indenting the Start of a Paragraph

Depending upon the style of letter you are writing, you
may wish to indent the first line of a paragraph.

1 The Indent Line (.IL) control word is used to
indent only the first (next) line of a paragraph. It
is only active for the accumulation of one output
line.

Type the following paragraph from Sample
Letter 2 to see how the Indent Line control word
functions:

```
.FO ON
.IL 5
This is to inform you that you
have been
selected as a qualified candidate for
admission to this school.
.SP
```

Your display should appear as follows:

```
=> .FO ON
=> .IL 5
=> This is to inform you that you
=> have been
      This is to inform you that you have
=> selected as a qualified candidate for
been selected as a  qualified  candidate
=> admission to this school.
=> .SP
for admission to this school.

=>
```

You will notice that Indent Line was functional for one line.

If these control words and text were in a permanent input file they would print the paragraph as:

```
      This is to inform you that you have
been selected as a  qualified  candidate
for admission to this school.
```

Offsetting the Start of a Paragraph

A control word that works the opposite from indent line is the Offset (.OF) control word. It formats the first (next) line as it has been formatting previous lines and then indents *all* following output lines of text.

1   To illustrate offsetting, let's use the same sample paragraph again.

Type the paragraph as follows. Items to be
typed are to the right of the prompt. The display
screen output is also included:

```
=> .OF 5
=> This is to inform you that you
=> have been selected
This is to inform you that you have been
=> as a qualified candidate for
     selected as a  qualified  candidate
=> admission to this school.
=> .SP
     for admission to this school.

=>
```

If these control words and text were in a
permanent input file they would print the
paragraph as:

```
This is to inform you that you have been
     selected as a  qualified  candidate
     for admission to this school.
```

2   To turn off offsetting:

Type **.OF**

Indenting a Paragraph

If you want to indent the full paragraph, use the Indent
(.IN) control word. The indent control word stays
active until you:

- Change it by typing it again with a different value.

- Reset it to zero by typing it without a value.

Let's see how the same paragraph is changed when we
use the indent (.IN) control word.

1   Type the same sample paragraph using the following sequence. The display screen output is also included.

```
=> .IN 5
=> This is to inform you that you
=> have been
      This is to inform  you  that  you have
=> selected as a qualified candidate
=> for admission to
      been selected as a qualified candidate
=> this school.
=> .SP
      for admission to this school.

=>
```

Notice that indent works on all output lines following the control word.

If these control words and text are written into a permanent input file, they would print as:

```
      This is to inform you that you have
      been   selected   as   a   qualified
      candidate  for  admission  to   this
      school.
```

2   To turn off indentation:

Type **.IN**

          or

Type **.IN 0**

3   To create the closing of the letter, the Indent control word is used in a different manner. Type the following closing. The display screen output is also included:

```
=> .FO OFF
=> .IN 20
=> Respectfully yours,
                    Respectfully yours,
=> .SP 3
=> Jonathan Wise
                    Jonathan Wise
=> Dean of Admissions
                    Dean of Admissions
```

If these control words and text were written
into a permanent input file they would print the
closing as:

```
                    Respectfully yours,




                    Jonathan Wise
                    Dean of Admissions
```

4   To turn off indentation:

Type **.IN**

Undenting a Line

If you want to stop indenting for only one line, use the
Undent control word (.UN) to allow one output line to
be moved to the left.

1   To indent and turn format on:

Type **.IN 5**

Type **.FO ON**

2   Now type the sample paragraph. Display screen
output is also included:

```
=> This is to inform you that you
=> have been
     This is to inform you that you have
=> selected as a qualified candidate
     been   selected   as   a   qualified
=> .UN 5
=>   candidate
=> for admission to this school.
=> .SP
for admission to this school.

=>
```

If these control words and text are written into
a permanent input file they would print the
paragraph as:

```
     This is to inform you that you have
     been  selected   as   a   qualified
     candidate
for admission to this school.
```

Emphasizing Printed Characters

Emphasizing a Phrase

SCRIPT/PC also allows you to emphasize input lines
to force bold print on the printer. To do this you must
first identify another set of characters (font) that, when
printed, will be emphasized (bold):

* You set up the new font to print by using the Define
  Font (.DF) control word, which allows you to
  define a new font and select a name for the font.

* You then use the new font by writing the Begin
  Font (.BF) control word followed by the new name.

* You can return to the old font by asking for the
  previous font (.PF).

The following example illustrates the three steps:

```
.DF BRIGHT FONT DS
.
.
.
.BF BRIGHT
.
.
.
.PF
```

Let's set up a similar example:

1   Type the following heading. The display screen
output is also included:

```
=> .DF BRIGHT FONT DS
=> .BF BRIGHT
=> .CE ON
=> College of Business Excellence
      College of Business Excellence
=> .PF
=> Collegetown, New York
         Collegetown, New York
=>.CE OFF
=>
```

If these control words and text were in a file
they would be formatted and print the phrase as:

College of Business Excellence
Collegetown, New York

2   This completes using the Practice Session. If
you want to end the session:

Press: **Esc**

Press: **Y** (in response to the prompt)

You should now be out of Practice mode.

Now that you have used some of the SCRIPT/PC
control words, you may wish to write some letters using
an editor such as the IBM PC Personal Editor or the

IBM PC Professional Editor. You should be able to
easily use the control words you have been practicing.
Feel free to change the values, to experiment, rather
than just copy the examples. Once you have finished
editing the file, proofread it, and then turn to the next
chapter to learn how to have SCRIPT/PC provide you
with formatted output.

Notes:

1. If the following are entered during a practice session
    they produce the results listed.

```
Key(s): Decimal: Results:

Ctrl-L              No action - form feed
Alt     012         No action - form feed
Alt     010         Same as F10  Continue
Alt     027         Same as Esc  Quit
Alt     013         Same as "Carriage Return"
```

2. When you have finished reading Chapter 5, you may
    wish to enter practice mode again to see how the
    tags work.

Have fun! Additional information on any control word
can be found in Book 3 Reference.

Chapter 4. How to Use Descriptive Commands (Control Words)

Contents

Introduction

This chapter will show you how to use SCRIPT/PC control words to create the documents you want. The chapter is arranged to show the basic function of a control word and where you might use it in your input document.

To input your document, you should have an editor such as the IBM PC Personal Editor or the IBM PC Professional Editor. When you have completed creating your input file and it has been proof read, either visually or with a product such as IBM PC Word Proof, you are ready to use SCRIPT/PC. For now, you are going to concentrate on how to markup your input text for SCRIPT/PC.

> **Note:** If you select the IBM PC Personal Editor, you must save your files using the NOTABS option for correct SCRIPT/PC processing.

SCRIPT/PC has its own language called control words (descriptive) and tags (declarative). You will use control words in this chapter to direct SCRIPT/PC to format your input text into an output document. If you have used the Practice Sessions part of SCRIPT/PC, you already have an idea about how to markup your input file to create an output file. This chapter is not intended to be used with Practice Sessions, although you may do so if you wish. To create the input files in this chapter you must use an editor that does not alter the ASCII characters your keyboard is capable of entering. SCRIPT/PC *does not* contain an editor that will create permanent input files.

Marking up the text of your input file as you create it is similar to all the operations you do when you type a document on a typewriter: tabbing, carriage return, indexing, centering text, indenting, and all the other operations you may be familiar with. If you don't have

typing experience, don't feel lost as SCRIPT/PC has been developed for both new and not-so-new users of text processors.

In this chapter, the control words are grouped together by function. You will be presented with examples of output text and the marked up input text that created the output. Examples of labels, letters, and customized form letters will be used to show you how SCRIPT/PC affects the output.

How to Create Labels

Since a label is one of the smallest forms you may be working with, it also becomes the easiest to create. Following are some example labels.

Printed Labels

```
Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345
```

```
John Smith
5678 Block Road
Tampa, FL   54321
```

```
Margaret Victoria
c/o Jennifer Reilley
5432 Main Street
Seattle, WA   67890
```

Label Input

The example labels are printed on gummed labels containing room for 5 lines. There are 6 lines from the top of one label to the top of the next. The format of each label is a maximum of 5 printed lines and a skip (space) to the top of the next label.

The sequence of input lines that created the example labels is:

```
.FO OFF
.IN 5
Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345
.SP 3
John Smith
5678 Block Road
Tampa, FL   54321
.SP 3
Margaret Victoria
c/o Jennifer Reilley
5432 Main Street
Seattle, WA   67890
.SP 2
```

Let's look at the control words and what they mean. All control words:

- Start with a period (.) at the beginning of each input line.

- Are two characters long.

- May contain values or directions following the control word.

You can write more than one control word on a line to conserve line space. Separate each control word or any text that follows it, with a semicolon (;). You will notice that the examples in this section are written using both methods.

Formatting

The Format control word allows you to either format the document exactly as you enter it, line for line, or it allows SCRIPT/PC to format the document for you. When you release control to SCRIPT/PC, all input lines will be strung together in a continuous stream until an output line is filled. In the labels example, you do not want the names and addresses strung together. That is why formatting is turned off.

```
.FO OFF
```

Indentation

To place the output lines correctly on the label, the indent control word (.IN) is used to create 5 character spaces:

```
.IN 5
```

Once set, indentation will remain the same until another indent control word or another control word that controls indentation is found. Control words that affect indentation are:

- Indent = .IN
- Indent Line = .IL
- Offset = .OF
- Undent = .UN
- Center = .CE

For the example output, the first label's input text is:

```
.IN 5
Mary Jones
1234 Pickford Avenue
Great Barrington, MA  12345
```

> **Note:** Formatting should be turned off (.FO OFF) before these labels are to be processed by SCRIPT/PC.

This results in SCRIPT/PC output text of:

```
Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345
```

The Indent Line control word is only active for one line, the next output line. The input text:

```
.IN 5
.IL 5
Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345
```

results in SCRIPT/PC output text of:

```
     Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345
```

Control of the starting character position is then returned to whatever indent control was in effect before the Indent Line control word was found, in this case .IN 5.

The Offset control word *adds* its value to the indent value. This increases indentation for all lines except the first line of output text that follows the control word.

The input text:

```
.IN 5
.OF 4
Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345
```

results in SCRIPT/PC output text of:

```
    Mary Jones
        1234 Pickford Avenue
        Great Barrington, MA   12345
```

The Undent control word changes the indentation
toward the left margin for one line. The value specified
is *subtracted* from the current indentation value.

The input text:

```
.IN 5
Mary Jones
.UN 3
1234 Pickford Avenue
Great Barrington, MA   12345
```

results in SCRIPT/PC output text of:

```
     Mary Jones
   1234 Pickford Avenue
     Great Barrington, MA   12345
```

Centering the output lines can be done by using the
Center control word. You can center one line or many
lines:

```
.CE ON
Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345
.CE OFF
```

results in SCRIPT/PC output text of:

```
           Mary Jones
        1234 Pickford Avenue
      Great Barrington, MA   12345
```

Additional details about these control words can be
found in the reference manual.

Spacing and Skipping Lines

Since there are six lines on each label (five print lines plus one line for label edges), use the space or skip control words to get to the start of the next label. Both of these control words work the same, except that skip will not be processed if it is at the top of a label or larger page.

- Space = .SP
- Skip = .SK

To skip to the top of the second label after the text on the first label, use either the skip (**.SK 3**) or space control word (**.SP 3**):

```
.IN 5
Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345
.SP 3
John Smith
5678 Block Road
Tampa, FL   54321
.SP 3
```

results in SCRIPT/PC output text of:

```
Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345
```

```
John Smith
5678 Block Road
Tampa, FL   54321
```

More About Spacing and Skipping

There are more ways to get to the top of a new page beside spacing or skipping the exact amount. You do

not need to know how much of the page you have already used to create a skip to top of page:

- Page Eject - written as .PA
- Conditional Page Eject - allows you to test for a minimum amount of lines remaining on a page. This control word is written as .CP.

How to Create a Letter

When you want to write a letter, there are more items to worry about than when you created labels. Page layout becomes all important when writing letters.

SCRIPT/PC will assume the following page layout:

```
Page Size            8 1/2  11 Inches
Page Length          66 Lines
Lines-per-inch       6 Lines-per-inch

Top Margin           6 Lines
Bottom Margin        6 Lines

Page Margin          0 Characters
Line Length          72 Characters
Column Width         72 Characters

Running Heading      None
Running Title        None

Single-column Mode
Single Space Mode
```

If you want to use a page layout other than the one SCRIPT/PC assumes, see the Chapter 6, How to Design a Page.

When you consider the input text for a letter, you should group the parts of a letter together:

- Your Organization Heading
- Inside Address
- Salutation

- Body
- Closing

An example letter is on the next page. Note that it is similar to the one in the Practice Sessions chapter.

Printed Letter

```
          College of Business Excellence
               Collegetown, New York


Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345


Dear Graduate,

     This is to inform you that you have
been selected  as  a qualified candidate
for admission to this school.

     A  complete  copy  of our  catalog,
with financial assistance forms, will be
sent to you in a few days.

     If you decide to  register for  our
fall semester, you will need to:

   1.  Send a completed application.
   2.  Attach a non-refundable check for
       $25.00.
   3.  Send a copy of your birth
       certificate.
   4.  Send a copy of a health physical
       held within the last 90 days.

                  Respectfully yours,


                  Jonathan Wise
                  Dean of Admissions
```

For explanation purposes in this book, the line length of our example letter has been set to 40 characters.

Letter Input

The input text and the corresponding control words that created the output letter are as follows:

```
.SP 2;.CE ON
College of Business Excellence
Collegetown, New York
.CE OFF
.SP 3
Mary Jones
1234 Pickford Avenue
Great Barrington, MA  12345
.SP 2
Dear Graduate,
.SP;.FO ON;.IL 5
This is to inform you that you have been
selected as a qualified candidate for
admission to this school.
.SP;.IL 5
A complete copy of our catalog, along with
financial assistance forms, will be sent
to you in a few days.
.SP;.IL 5
If you decide to register for our
fall semester, you will need to:
.FO OFF;.SP;IN 3
1.  Send a completed application.
2.  Attach a non-refundable check for
    $25.00.
3.  Send a copy of your birth
    certificate.
4.  Send a copy of a health physical
    held within the last 90 days.
.SP
.IN 17
Respectfully yours,
.SP 4
Jonathan Wise
Dean of Admissions
```

One of the first things you probably will notice is that
all control words and input text do not match the final
text. You do not have to create the final format in your
input text. Add the correct control words to do it for
you.

Once finished entering your input file, you may look at
the output file after SCRIPT/PC processes it, either on
the screen (to check for correct format) or in printed
form. A third option is also available — to put the
output file on a disk for future use.

How to Control Formatting

Note that formatting was turned off to start the letter, and turned back on after the salutation and prior to the body. it was turned off again just prior to the list. If you removed the **.FO OFF** and **.IN 3** control words prior to the list, then your output would have changed to:

```
         College of Business Excellence
              Collegetown, New York


Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345


Dear Graduate,

        This is to inform you that you have
been selected  as a  qualified candidate
for admission to this school.

        A  complete copy  of  our  catalog,
along  with  financial assistance forms,
will be sent to you in a few days.

        If  you decide to register for  our
fall  semester, you  will  need  to:  1.
Send   a   completed   application.   2.
Attach  a   non-refundable   check   for
$25.00.   3.  Send a  copy of your birth
certificate. 4.  Send  a   copy   of  a
health  physical  held  within the  last
90 days.

                 Respectfully yours,



                 Jonathan Wise
                 Dean of Admissions
```

In this case, leaving formatting on for the list changed how the letter looks. You may wish to use this to your

advantage when you are having a problem with fitting a letter on a page. Formatting on uses less lines than formatting off.

When formatting is on, it starts gathering up input lines until the output line width (specified by the Line Length control word in single-column mode or Column Width control word in 2-column mode) is reached, or until another control word tells SCRIPT/PC to do something else. These other control words cause SCRIPT/PC to produce an output line and begin to gather up a new line. This is called a formatting change, or in more common terms, a *break*.

Not all control words cause a break in formatting. However, you are already familiar with some that do, such as the space, skip, and page eject control words. Each of these control words, when found, cause SCRIPT/PC to stop gathering input lines and space down one or more lines. This "break" in formatting does not occur with all control words but will be identified under each control word in the Reference Section.

One of the simplest ways to force a formatting break is to use the Break control word (.BR).

In the preceding letter, if you wanted to leave the closing of the letter at the left column margin, then you would change the indent control word (.IN 16) to indent 0 (.IN or .IN 0). All indentation is then cancelled.

How to Create a Form Letter You can Customize.

If you want to send the same letter to more than one address, it would take a long time to create the same letter for a list of different names and addresses. Not

with SCRIPT/PC! If you create one form letter and a separate list of names and addresses, SCRIPT/PC will create duplicate copies of the letter, placing a different name and address at the desired position on each letter.

The letter we are going to use is the same as the last example.

Creating a Name and Address File

Let's use the file that created our name and address labels. We will modify it a little bit to create a name and address data file that we can use for the inside address of a form letter. Each letter will be the same, except for a different inside address. We can change the original input file easily by changing *all* the space control words (.SP 2 and .SP 3) to End File (.EF) control words and removing any other SCRIPT/PC control words. The End File control word will identify to SCRIPT/PC the end of every name and address sequence.

The last input line is responsible for changing the value of the symbol QUIT to YES. This value will be set after the last name and address has been read by SCRIPT/PC. Let's call this file **NAMES.SCT** and set it up as follows:

```
Mary Jones
1234 Pickford Avenue
Great Barrington, MA   12345
.EF
John Smith
5678 Block Road
Tampa, FL   54321
.EF
Margaret Victoria
c/o Jennifer Reilley
5432 Main Street
Seattle, WA   67890
.SE QUIT = YES
```

Creating a Letter File

The original sample letter file must be changed by replacing the name and address (inside address) with a control word that will allow you to insert a name and address from the **NAMES.SCT** input file. The control word that allows us to do this operation is the Imbed control word (.IM).

Let's call the letter file we are creating **LETTER.SCT** and assume that it is stored on the diskette in drive B. We will assume that the input file, **NAMES.SCT,** will also be stored on the same diskette in drive B.

The end of the letter file, **LETTER.SCT**, must be changed to include the control word that allows the letter to be started again, if there are any remaining names in the **NAMES.SCT** file. This control word is the Append control word (.AP).

The decision to process more letters is made by the If control word (.IF). If the test is *true* (&QUIT = NO) then the input line continues to be processed (.AP B:LETTER). If the test is *false* (&QUIT = YES), the rest of the input line (the target) is ignored and processing continues on the *next* input line. Let's look at it another way. After the completion of every letter:

- If there are names and addresses still to be processed, SCRIPT/PC will return to the to the beginning of the letter and create another copy.

- If there are no more names and addresses to process, SCRIPT/PC will stop producing letters and continue with the next input line after the letter (if any). If there are no more input lines, SCRIPT/PC is ended.

```
.SP 2;.CE ON
College of Business Excellence
Collegetown, New York
.CE OFF
.SP 3
.IM B:NAMES
Mary Jones
1234 Pickford Avenue
Great Barrington, MA  12345
.SP 2
Dear Graduate,
.SP;.FO ON;.IL 5
This is to inform you that you have been
selected as a qualified candidate for
admission to this school.
.SP;.IL 5
A complete copy of our catalog, along with
financial assistance forms, will be sent
to you in a few days.
.SP;.IL 5
If you decide to register for our
fall semester, you will need to:
.FO OFF;.SP;IN 3
1.  Send a completed application.
2.  Attach a non-refundable check for
    $25.00.
3.  Send a copy of your birth
    certificate.
4.  Send a copy of a health physical
    held within the last 90 days.
.SP
.IN 17
Respectfully yours,
.SP 4
Jonathan Wise
Dean of Admissions
.PA
.IF &QUIT = NO .AP B:LETTER
```

Creating a Control File

There is only one additional file to be created. Let's
call this file **ADMIT.SCT** and also store it on the same
diskette as the previous files. This file identifies the
two input files that are to be used to create our
customized letters:

```
.SE QUIT = NO
.DD B:NAMES.SCT
.AP B:LETTER.SCT
```

- The Define Data File-id control word (.DD) identifies the file that contains the blocks of input text (inside addresses).

- The Append control word (.AP) identifies the file that contains the body of your text.

- The set symbol control word (.SE) gives the symbolname QUIT the value of NO. This QUIT symbol will be tested at the bottom of every letter to check for it being the last letter. The value of QUIT will be changed in the **NAMES.SCT** file, as the last inside address is completed.

How to Start

To create the output letters you want, the file-id that starts the processing of the form letters is **B:ADMIT.SCT.** See Chapter 2, "Using SCRIPT/PC," if you need to review the start procedures.

SCRIPT/PC will create output letters containing different names and addresses for each entry in the file **NAMES.SCT** (in this case, three letters).

The printing stops when the last name and address field is read. It is accomplished by using the Set Symbol (.SE) control word to change the value of QUIT from NO to YES. The test at the end of the **LETTER.SCT** file will no longer be met and creation of more form letters will be stopped.

How to Create Prompts and Replies

SCRIPT/PC has the ability to communicate with you as it is processing your files. When you need to send a message to the screen, use the Type on Terminal control word (.TY). An example is:

```
.TY Place BLANK diskette in Drive A!
```

To enter data from your keyboard into SCRIPT/PC (either text input or control words), use the Terminal Input control word (.TE). If it is used in conjunction with the Type on Terminal control word, you can set up a prompt/response (interactive) environment.

```
.TY Enter name of student!
.TE
```

When SCRIPT/PC finds the Type on Terminal control word, it issues the text to the screen, reads the next control word (.TE), stops processing and waits for you to type a reply. It then continues processing input lines after your reply is complete.

> **Note:** When you are using the Terminal Input [.TE] control word, you should press the ◄─┘ key *once* for every entry into SCRIPT/PC. You will be prompted for your entry with the same symbol you saw in practice sessions (=>). To end terminal input before SCRIPT/PC (for example, **.TE 10**) press the ◄─┘ key *twice* in succession.

How to Enter Date and Time-of-Day

When you want to enter today's date or the time-of-day into a document, a special set of commands have been created for you. To enter todays date in text, type:

```
&SYSDATE.
```

ALL words that start with an ampersand (&) are special SCRIPT/PC words (symbols). SCRIPT/PC is always looking for these words to help you keep track of some important items. &SYSDATE is a shorthand method of saying to SCRIPT/PC, "Insert today's date here." You inserted a system date when you turned on your IBM Personal Computer and you answered the **DATE** prompt. The date you entered will be printed at the location in your letter that &SYSDATE appears. There are other special names that will allow you to insert specific time and date information.

Some of the special words and their descriptions are:

1. &SYSDATE. - Insert Date in the form month-day-year.

2. &SYSTIME. - Time-of-Day

3. &SYSDAYOFM. - Day-of-Month

4. &SYSSECOND. - Seconds

5. &SYSMINUTE. - Minutes

6. &SYSHOUR. - Hour

7. &SYSYEAR. - Year

8. &. - Page Number

Each of these special words must be entered in *uppercase* characters and must be followed by a period.

The ampersand (&), by itself bears special mention. Whenever SCRIPT/PC finds the special symbol & by itself, it assumes you want the page number inserted at this point.

Use the special page numbering symbol (&) when you set up a multi-page form as shown in Chapter 6, "How to Design a Page." You can use the symbol (&) when you set up a running title, for example:

```
.RT BOTTOM 3 /&SYSDATE./Book Report/Page &./
```

results in printing at the bottom of every page:

```
05-22-1983      Book Report      Page 1
```

> **Note:** The page number will advance by one on
> every following page. See *Book 2 - Quick Reference*
> for a complete list of special symbols.

How to Draw Boxes

We have not yet discussed the Box control word. This
control word allows you to place text within boxes to
highlight important ideas or to create figures and
diagrams.

Getting Started

The box is column oriented, so all references start with
a beginning box statement. If you wanted a box to:

1. Start in column 1 and end in column 20,
2. Continue for 5 lines,
3. And then end, you would write the control words
    as:

```
.BX 1 20
.SP 5
.BX off
```

The box that is generated looks like this:

```
┌─────────────────────────┐
│                         │
│                         │
│                         │
│                         │
│                         │
└─────────────────────────┘
```
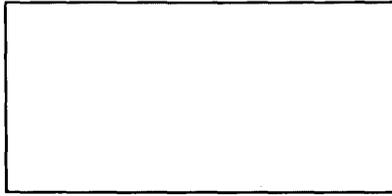
Adding Text

To insert text into the box, the text should also be column oriented. The following example shows how an area of text can be imbedded within the box:

```
.BX 1 20
.SP
  This Box
     is EASY
         to use.
.SP
.BX OFF
```

This results in the following output:

```
┌─────────────────────────┐
│                         │
│         This Box        │
│          is EASY        │
│           to use.       │
│                         │
└─────────────────────────┘
```
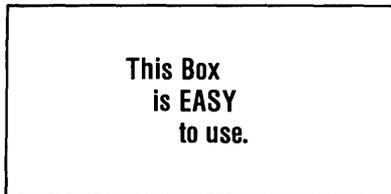
The following:

```
.BX 1 20
 This box is easier
.BX OFF
```

results in:

```
┌─────────────────────────────────┐
│          This box is easier!     │
└─────────────────────────────────┘
```

Many Boxes

You can also divide a large box into smaller boxes by
specifying vertical divisions between the beginning and
ending box number:

```
.BX 1 6 10 15 20
```

```
.BX OFF
```

results in:

```
┌────┬────┬────┬────┐
│    │    │    │    │
└────┴────┴────┴────┘
```
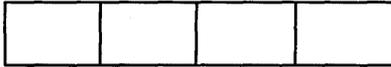
Tabbing to Enter Text

Addition of text to this sequence of boxes can be done
very easily if you use the Tab Setting (.TB) control
word and the Translate Input (.TI) control word.

Setting Tab Stops

Tabs are set by issuing the tab setting control word
followed by the column numbers in which you want the
tabs set. For example, to set tab stops for the previous
example, you would write:

```
.TB 2 7 11 16
```

Identifying Tabs in Text

The tabs must be identified within your input text. The actual internal tab control character is decimal 09. However, to place the code where you want it, select a character that will represent decimal 09 in your input line. If you selected the greater than (>) character, then the control word would be written as:

```
.TI > D009
```

Now, every time you place a > into your input line, SCRIPT/PC will skip to the next tab stop, filling the space with blanks. The following shows how to place text within the boxes we defined in the previous example.

```
.TI > D009
.TB 2 7 11 16
.BX 1 6 10 15 20
>This>one>is>not
```

results in:

This	one	is	null

Combination Boxes

Boxes are very easy to combine. The box can be redefined as it is being drawn. The following shows how you would combine the preceding examples:

```
.BX 1 20
 This box is easier
.TI > D009
.TB 2 7 11 16
.BX 1 6 10 15 20
>This>one>is>not
.BX OFF
```

This results in the combination box of:

This box is easier!			
This	one	is	not!

How To Print Two Columns

To use Multicolumn (2-column) Mode, you should take into consideration:

1. The starting column numbers, as specified by the *Column Definition* control word (.CD).

2. The width of each column, as set by the *Column Width* control word (.CL).

3. The *Multicolumn* control word (.MC) that starts two-column mode.

To print two columns of text, you would enter the following control word sequence:

```
.CD 2 1 35
.
.
.
.CL 30
.MC
```

Where:

.CD 2 Is the Column Definition control word and value for two columns.

1 35   Is the starting column number for each of the
          two columns.

.CL 30   Is the Column Width control word that
          specifies that each column contains 30
          characters.

.MC   Is the start of the multicolumn mode. It does
          not have to be coded directly after the column
          definition, but may be delayed until you want
          to use it.

   Note: To return to single-column mode, enter the
   Singlecolumn Mode control word (.SC).

This completes using control words. This chapter has
attempted to demonstrate to you some of their uses. It
has not attempted to show you all of the control words
and applications of each. The additional control words
located in *Book 3 Reference* will demonstrate even more
uses of each control word.

Upon completion of the Practice Sessions chapter and
this chapter, you should have a basic understanding of
some of the "how to build" techniques needed to create
a professional-looking document. The deSCRIPTive
control words allow you to customize your document at
every step.

The next chapter describes how to use the declarative
commands (tags). This allows you to specify "what to
build" and allows you more freedom for entering your
text with less attention needed for "how to build."
Since each tag is made up of many control words, the
tags create an easy way to create very complex
documents.

.

Chapter 5. How to Use Declarative Commands (Tags)

Contents

Introduction

SCRIPT/PC provides a starter set of tags for assisting in text markup. These tags are made up of many SCRIPT/PC control words. Instead of entering the same string of control words every time you want the same function to be performed, you can select one of the tags to do it for you, somewhat like a "shorthand" method of marking up your text. In other words, the Starter Set tags provide you with an additional and, in many cases, easier, shorter method to mark up a document for text formatting.

One of the major differences between control words and tags is the beginning character. A control word always starts with a period (.) and a tag always begins with a colon (:). A tag may have up to 10 characters in its name; however, the control word is limited to two. A tag must end with a period (for continuation) or a blank. Tags are also dependent upon the formatter being on (**FO ON**) for correct operation. Although tags and control words can be mixed within a document, if the formatter has been turned off by a control word, then it must be turned back on prior to the next tag. In no case can tags and control words be mixed on the same input line; results will be unpredictable.

Before we look at the tags and what they do, let's look at the normal way a document is organized:

Parts of a Document

There are normally 4 major parts (elements) to any document, whether it is a brochure, a small booklet, a novel, or a complex technical manual. They are the:

Front Matter   The front matter contains the title page, the abstract, the preface, the table of contents, and the figure list.

Body   The body of the document is the main part of your document; it contains your sections or chapters.

Appendixes   The appendix section contains all the material (usually reference) that does not normally fit into the body of the document.

Back Matter   The back matter is where the glossary and index are contained.

Each document you create will not have all the pieces of each major element. However, in general, a common format is followed.

Each of the major elements contains many parts (elements within themselves). It is the purpose of this chapter to identify the parts and to introduce you to the commands, called tags, that are available to markup a document.

Document Control

There are two tags which are used to define the beginning and end of your document. These tags must be used in any input file that contains tags. They are:

- General Document, written as:

  `:GDOC`

- End General Document, written as:

  `:EGDOC`

These tags also provide for generation of a bottom title which includes in the following order, from left to right:

1. Optional bottom running title (centered on the page).
2. Page number (right side of the page).

The page numbering is the only portion of the bottom title that will always be printed. When you add or remove text in your source file or even add or delete entire sections, the pages will be renumbered with no additional work on your part.

> **Note:** All documents that use the Generalized Markup Language (GML) Starter Set Tags *must* start with the general document statement (:GDOC) because of the procedures that are started by this tag.

Options to a tag are called attributes as opposed to options to a control word which are called parameters.

If you would like a security classification placed on the top of every page, you should use the option called SEC, by specifying as its value the text of the security classification. For example:

```
:GDOC  SEC='Top Security'
```

Front Matter of Document

The front matter consists of the material in the front of a book and itself contains many smaller parts (elements) such as the title page, abstract, preface, table of contents and list of illustrations. Not all of the elements should be used in every document. However, when you do use an element, it should be used in a very specific fashion. Each element listed below also identifies the tag name that is used to identify the element to SCRIPT/PC. You will note that most of

the tag names are similar to the element name, thereby aiding in your ease of learning them. The front matter is ended by SCRIPT/PC detecting the body tag.

- Front Matter - written as:

  ```
  :FRONTM
  ```

Information included after the front matter tag and before the body tag includes:

- Title page
- Abstract
- Preface
- Table of Contents
- List of Illustrations

The front matter tag is followed by the elements within the front matter as they normally occur. These elements are listed in the normal order of occurrence followed by the declarative command (tag). Indentation of the various elements is done for illustrative purposes throughout the chapter and does not mean that the tags have to be written in like manner. The tags have also been written in UPPER CASE for better visual clarity, just as the control words were:

```
Front Matter - :FRONTM
  Start Title Page - :TITLEP
    Title - :TITLE.document title
    Document Number - :DOCNUM.document number
    Date - :DATE.today's date or text
    Author - :AUTHOR.name of writer
    Start Address - :ADDRESS
      Address Line - :ALINE.first line of address
      Address Line - :ALINE.last line of address
    End of Address - :EADDRESS
  End of Title Page - :ETITLEP
  Abstract - :ABSTRACT
  Preface - :PREFACE
  Table of Contents - :TOC
  List of Illustrations - :FIGLIST
```

The ending tags such as End of Address and End of Title Page are shown in the front matter sequence to demonstrate to you where they should be located.

Tags within the front matter are described as follows:

:TITLEP   This tag follows immediately after the front matter tag and signals the beginning of the title page.

:TITLE.   This tag is immediately followed by the title of the document.

:DOCNUM.   This tag is immediately followed by the document number.

:DATE   If there is no text immediately following this tag, the current date (as it is known to your personal computer) is inserted into your output in the form of month, day, and year (for example, 07-22-1983). If there is text following the control word, the text is inserted into your output.

:AUTHOR.   This tag allows you to place the author's name on the title page. If there are multiple authors, you may repeat the tag.

:ADDRESS   This tag marks the start of the address within the title page.

:ALINE.   This tag allows you to place an address line on the title page. You may use more than one address line tag to complete an address; it should follow the address tag.

:EADDRESS   This tag ends the address area.

:ETITLEP   This tag ends the title page.

:ABSTRACT   This tag identifies the abstract page. The actual abstract is written just after this line in text:

```
:ABSTRACT
:p.This abstract is ......
```

or it may be imbedded using the Imbed
control word, for example, a file-id of
B:ABS.SCT:

```
:ABSTRACT
.IM B:ABS
```

The abstract section usually provides
some brief description of the document's
contents, purpose, and so on.

:PREFACE   This tag identifies the preface page. The
actual preface is written just after this
line in text:

```
:PREFACE
:p.This preface is ......
```

or it may be imbedded into the file by
use of the Imbed control word, for
example, a file-id of B:PRE.SCT:

```
:PREFACE
.IM B:PRE
```

Usually the preface of a document is
used to describe the document itself and
is not normally about the actual contents.

:TOC   The table of contents needs two passes
through the file to be built. On the first
pass, all the headings are only
accumulated. On the second pass
through the file, the table of contents
page(s) are filled. If you only take one
pass through the file, the table of
contents page will be there, but there will
be no entries on the page.

For documents whose table of contents is
greater than 1 page, a special symbol,

&TOC, must be changed by you to allow the larger table of contents. This symbol is defined in the library GMLLIB.SCT.

:FIGLIST   The list of illustrations also needs two passes through the file to be built. On the first pass, all the figures are only accumulated. On the second pass through the file, the list of illustrations page(s) are filled. If you only take one pass through the file, the list of illustrations page will be there, but there will be no entries on the page.

For documents whose list of illustrations is greater than 1 page, a special symbol, &FIGLST, must be changed by you to allow the larger list of illustrations. This symbol is defined in the library GMLLIB.SCT.

Example of a Title Page

The following title page has been designed for a book. To translate this design to SCRIPT/PC, you would use the GML tags that follow the example. Notice that the document number and date have been changed with the author and author's address in this example. However, they are still part of the title page:

```
Weather Patterns
in the United States

John S. Doe

5467 Stein Street
Hancock, NY 34567


AMERHIST123

08-10-1983
```

The file that created the title page contains the
following tags and text.

> **Note:** Indentation has been used to assist you in
> understanding groups of information.

```
:GDOC
  :FRONTM
    :TITLEP
      :TITLE.Weather Patterns
      :TITLE.in the United States
      :AUTHOR.John S. Doe
      :ADDRESS
        :ALINE.5467 Stein Street
        :ALINE.Hancock, NY 34567
      :EADDRESS
      :DOCNUM.AMERHIST123
      :DATE
    :ETITLEP
```

The title page is normally followed by the abstract,
preface, table of contents, and list of illustrations.
There may not be some of these elements in your
document.

```
:ABSTRACT
  :p.This abstract ..........
        . . .
:PREFACE
  :p.This preface ..........
        . . .
:TOC
:FIGLIST
```

Body of Document

When the front matter is finished, the body tag is used
to signal to SCRIPT/PC to reset the page count and to
start numbering chapter headings. This is the main
section of your document. All elements before or after
the body element are only supportive information.

The following elements are normally found within the body of a document and directly control the format of your document:

- Paragraph control

- Headings for topics - 7 levels

- Listings

  - Simple lists
  - Unordered lists
  - Ordered lists
  - Definition lists

The Body tag ends the front matter and starts the main part of your document. The following example shows you some of the tags (elements) within the body of a document.

```
:BODY
  :H1.Chapter heading text
    :P.Text of paragraph 1 ........
        . . .
        . . .
    :PC.continuation of paragraph 1
  :H2.Topic text
    :P.Text of paragraph 2 ........
      :UL  COMPACT
        :LI.Text of first list item
        :LI.Text of second list item
          . . .
          . . .
      :EUL
        . . .
```

When SCRIPT/PC processes this file, it results in:

<u>CHAPTER HEADING TEXT</u>

Text of paragraph 1 ....

. . .
continuation of paragraph 1

TOPIC TEXT

Text of paragraph 2 ....

o Text of first list item
o Text of second list item

Paragraph Control

There are two tags which are used to define paragraphs.

- Paragraph start - written as:

  ```
  :P.text of paragraph
  ```

- Paragraph continuation - written as:

  ```
  :PC.continuation of paragraph text
  ```

In SCRIPT/PC each tag operates the same way. The first tag (:P.text) marks a paragraph beginning while the other tag (:PC.text) is used to resume a paragraph. The latter is usually used to continue a paragraph which may have been broken up by a list or an example.

You do not have to skip a line before this tag, indent it, or use any special controls. Start each paragraph on a new line; the length of the input text line is not important. SCRIPT/PC will take care of line length as it formats your document.

Headings

Heading tags identify to the reader the various topics in your document. The heading tags are used to define various chapter, section and sub section titles. Additionally, they are used to create the table of contents file which forms the table of contents for the document.

The starter set allows you seven levels of headings, Heading 0 [:H0] through Heading 6 [:H6].

- Heading 0 - Section heading - used to divide a large document into several parts or sections, each part or section containing several chapters.

- Heading 1 - Chapter heading - Used to create chapter headings

- Heading 2 - Topic heading

- Heading 3 - Topic heading

- Heading 4 - Topic heading

- Heading 5 - Topic heading imbedded in text

- Heading 6 - Topic heading imbedded in text

The text used with a heading tag is capitalized, underscored, boldface printed or a combination, depending upon the heading type you select. A complete description of every heading is included in *Book 2 - Quick Reference*. However, there is no processing of the text for the table of contents so it appears "as typed" in the table of contents.

Lists

Four types of lists are supported via the Generalized Markup Language tags. Each type of list should be started and ended with a tag.

```
Type of list       Starting Tag  Ending Tag

Simple List        :SL           :ESL
Unordered List     :UL           :EUL
Ordered List       :OL           :EOL
Definition List    :DL           :EDL
```

Each entry within a simple, ordered or unordered list should be preceded by the list item tag:

```
:LI.text of item in list
```

The definition list is different. It creates a list of terms that will have their own definitions. It is a more complex list than the other three.

There is an additional list tag that allows you to temporarily suspend the list you are in.

```
:LP.text of additional material
```

How to Set Up Lists:

- Simple list

```
:SL
    :LI.text of first list item
        . . .
        . . .
    :LI.text of last list item
:ESL
```

- Ordered list

```
:OL
    :LI.text of first list item
        . . .
        . . .
    :LI.text of last list item
:EOL
```

- Unordered list

```
:UL
   :LI.text of first list item
       . . .
       . . .
   :LI.text of last list item
:EUL
```

- Definition List

```
:DL TSIZE='14' TERMHI='0'
   :DT.text of term
   :DD.text of term description
       . . .
       . . .
:EDL
```

Each list type has a start-list tag and an end-list tag.
The simple (:SL), ordered (:OL) and unordered (:UL)
lists have a list-item tag (:LI.text of item) to specify an
entry to a list. However, the definition list needs *two*
tags to define an entry, one for a list-entry term
(:DT.text of term) and one for the list-entry description
(:DD.text of definition).

The **ordered list** highlights each list-entry with a
numeric or alphabetic character which increases by one
for each list entry. Using this type of list makes it easy
to document "a list of *n* items."

The **unordered list** is similar to the ordered list except
that instead of using an increasing decimal digit, it just
uses a small "o" as its list-entry identifier. The second
level list is identified by a hyphen (–); the third by two
hyphens (—) and the fourth level by a repeat of a small
"o".

The **simple list** is similar to the unordered list. The
difference is that the simple list has no explicit bullet at
all.

The **definition list** is the most complex of the list types.
It also has a method to identify each list item.

However, instead of a number or a bullet (o), it uses a short term which you specify. It uses a different format than the other lists.

Normally each item in a list is separated by a blank line. However, SCRIPT/PC allows you to suppress the blank line, if you wish. This is done by selecting the COMPACT option (attribute) on the line that identifies the type of list.

A list-entry can contain a sub-list, which can also contain a sub-list, and so on. This *nesting* of lists gives you greater flexibility in structuring the presentation of your text.

Figures

A figure generally is a form of pictorial or graphic information which helps to clear up points you are explaining in your document. The figure oriented tags provide an organized way to show this data in a consistent format:

- Figure start - written as:

  ```
  :FIG
  ```

- Figure end - written as:

  ```
  :EFIG
  ```

- Figure Caption - written as:

  ```
  :FIGCAP.text for caption
  ```

- Figure Description - written as:

  ```
  :FIGDESC.text for description
  ```

- Figure Reference - written as:

  ```
  :FIGREF ID=xx
  ```

An example of a figure is as follows. The space is being
reserved for *drop-in* art:

```
:FIG
 1
   2
     3
       4
         5
           6
             7
               8
                 9
                  10
:FIGCAP.Hawaiian Islands
:FIGDESC.Rainfall variations
:EFIG
```

This results in:

```
-----------------------------------------
   1
     2
       3
         4
           5
             6
               7
                 8
                   9
                    10

   Figure 1.   Hawaiian
               Islands: Rainfall
               variations
-----------------------------------------
```
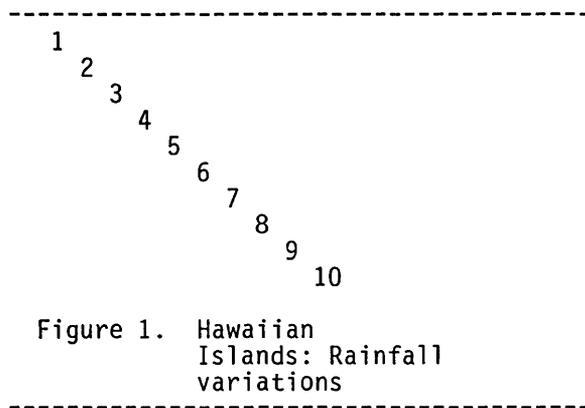
Highlighting

Certain highlighting techniques may be applied to text
to provide emphasis. The highlighting possibilities are:

- Underscoring   (red on color display)

  ```
  :HP1.text to be underscored :EHP1.
  ```

- Bold   (intensified on display)

```
:HP2.text to be made bold :EHP2.
```

- Combination

```
:HP3.text to be bold and underscored :EHP3.
```

An example of <u>Underscoring</u> or **Bold** or a **<u>Combination</u>**
can be done by using these tags.

Appendixes of Document

If you want appendixes in your document, you should
start them with the Appendix tag. Each new Heading 1
that is read is considered to be the start of a new
appendix. It resets the page count and bumps up the
alphabetic appendix designation (Appendix A,
Appendix B, Appendix C, and so on).

- Appendix start - written as:

```
:APPENDIX
  :H1.heading text for appendix A
     . . .
  :H1.heading text for appendix B
     . . .
  :H1.heading text for appendix C
     . . .
```

Back Matter of Document

The back matter of a document normally contains the
glossary, bibliography, and the index. The Back Matter
tag is used to tell SCRIPT/PC to stop the appendixes
or body if there are no appendixes. It is the last major
element of a document and is ended by the End
General Document tag.

- Back Matter start

- — Glossary heading

- — Bibliography heading

- — Index of terms

An example setup of the back matter of a document is as follows:

```
:BACKM
  :H1.Glossary
    . . .
  :H1.Bibliography
    . . .
  :INDEX
:EGDOC
```

Glossary

If you want your document to have a glossary, start the glossary as you would a normal type of chapter within the body of the document:

```
:BACKM
:H1.Glossary
  . . .
```

Index

The index tags for SCRIPT/PC will create an index to assist in referencing the important items in your document. For documents that are large, an index is a very valuable section to include. It must be the last part of a document and is part of the back matter:

- Index start - written as:

  ```
  :INDEX
  ```

- Index term - written as:

  ```
  :I1.text of index term
  ```

- Index term reference - written as:

```
:IREF
```

Miscellaneous

The following tags define footnotes, quotes (both short and long), examples, citations, and notes. They are listed in alphabetical order:

- Citation - written as:

```
:CIT.text for citation :ECIT
```

- Footnote - written as:

```
:FN.text for footnote :EFN
```

Space at the bottom of a page should be reserved before this tag can be used. It is reserved by writing:

```
.SE $FN = n
```

Where:

n   specifies the number of lines you want to reserve for your footnote(s)

- Footnote Reference - written as:

```
:FNREF REFID=xx
```

- Long Quotation - written as:

```
:LQ.text for long quotation
:ELQ
```

- Note - written as:

```
:NOTE.text for note
```

- Quotation -written as:

```
:Q.text for short quotation :EQ
```

- Example - written as:

```
:XMP.
    Text for example can be up to
    one page
            and is
                    copied
                            exactly
    as it is entered.
:EXMP
```

Chapter 6. How to Design a Page

Contents

Page Layout

This section describes the control words that have been made available to you for designing a page layout.

Each file you set up will be custom tailored to the size paper (length and width) you are going to use and how you want to divide the paper (top and bottom margins, page margins, number of columns, line length, and body of page). There will also be a number of reserved areas in which predetermined text, such as running headings and running titles, will be printed.

It is the intent of this section to guide you in the setup of your page.

Page Dimensions

When you want to create a document, you must tell SCRIPT/PC how you want the paper in your printer to be formatted, if it is different from the assumed (default) values. You must also tell SCRIPT/PC some key information about the paper size and where you want your text to print.

Page Format

The following figure shows the format of a page:

```
┌──────────────┬───────────────────────────────────┬──────────┐ ▲
│              │         Top Margin (.TM)           │          │ │
│              ├───────────────────────────────────┤          │ │
│              │      Running Head (.RH)            │          │ │
│ Page         │                                   │          │ │
│ Margin       │ ◄────── Line Length (.LL) ──────► │          │ Page
│ (.PM)        │                                   │          │ Length
│              │                                   │          │ (.PL)
│              │                                   │          │ │
│              │                                   │          │ │
│              ├───────────────────────────────────┤          │ │
│              │       Bottom Margin (.BM)          │          │ │
│              │                                   │          │ │
│              │       Running Title (.RT)          │          │ ▼
└──────────────┴───────────────────────────────────┴──────────┘
```
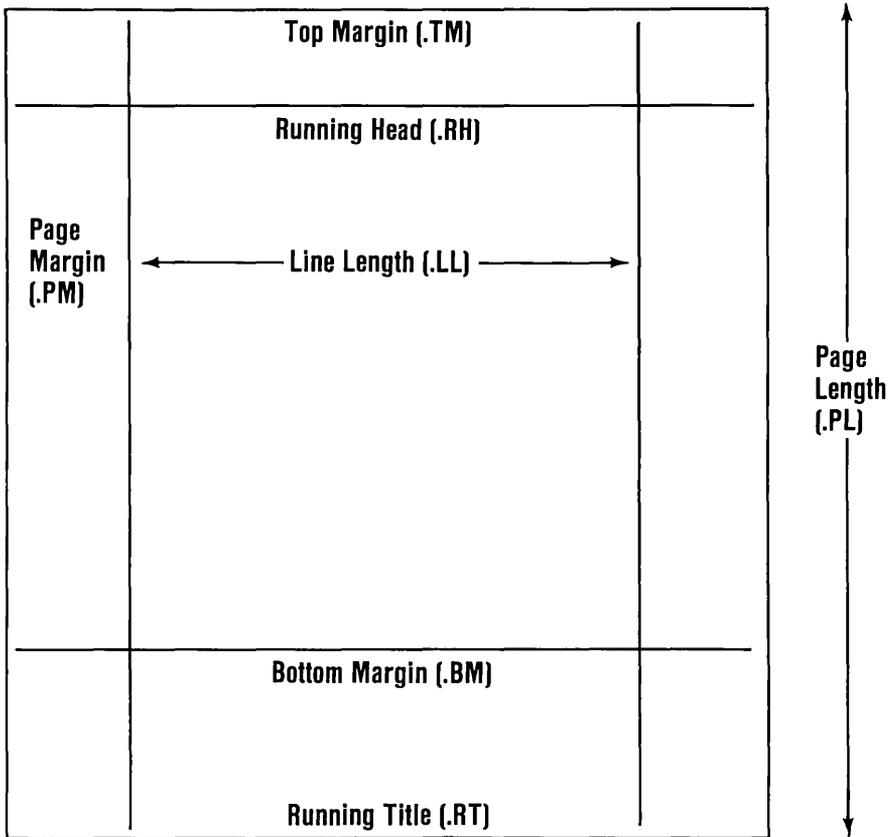
The items you can change, the corresponding control words and SCRIPT/PC assumed default values are shown in the following table:

Desired Item	Control Word	Default
Set page margin	.PM	0 chars.
Set page length	.PL	66 lines
Set line length	.LL	72 chars.
Set top margin	.TM	6 lines
Set bottom margin	.BM	6 lines
Set running heading	.RH	None
Set running title	.RT	None

SCRIPT/PC also assumes:

- Single-Column Mode.

- Single-Space Mode.

- Character Set 1 on the IBM Graphics Printer.

- Printer set at 6 lines-per-inch.

- Paper that is 11 inches long.

How to Change the Page Margin

The page margin control word causes SCRIPT/PC to shift the formatted output on each page to the right. For example:

```
.PM 5
```

sets the page margin to 5 character spaces. If each character space equals 1/10 inch, then you have

created a page margin of .5 inches (one half inch). If, however, you wish a larger margin of one inch, then you would set the page margin as:

```
.PM 10
```

How to Change the Page Length

Page length can be changed by using the page length control word; top and bottom margins would not change.

If your paper measures 11 inches, and you are printing at six lines per inch, then the page length for this paper is computed as: 11 X 6 = 66 lines. You would write the page length as:

```
.PL 66
```

If, however, you are printing at 8 lines per inch, then your page length for the same 11 inch paper is written as:

```
.PL 88
```

How to Change the Line Length

Once your page margin is set correctly and you have enough space for the binding you desire, your next step will be to calculate the maximum line length that you want.

If your paper is 8.5 inches wide and your page margin is set to .5 inches, then you have a maximum print area of 8 inches. However, if you want to leave a right margin the same size as your page margin, you have: 8.5 (paper width) - .5 (page margin) - .5 (right margin) = 7.5 inches for text. Since each inch on this example equals 10 characters (10 pitch), then your line length should be written as:

```
.LL 75
```

For this example, a line length of 7.5 inches (75 characters) is the maximum column width for single-column mode.

> **Note:** The control word that controls the column width of multicolumn (2-column) mode is Column Width (.CL).

How to Create an Example Page

In the following example, desired values are shown within the example page. Conversion of the desired values to SCRIPT/PC control words is shown in the second step on the next page. SCRIPT/PC control words needed to achieve the page layout are shown in step 3.

Step 1 - Design the page you want.

```
┌──────┬──────────────────────────────────────┬──────┐         ▲
│      │          Top Margin (6 lines)         │      │         │
│      │                                       │      │         │
│      ├──────────────────────────────────────┤      │         │
│      │           Tale of Two Cities          │      │         │
│      │                                       │      │         │
│ Page │                                       │      │         │
│Margin│ ◄──────  Line Length (7.5 inches) ──► │      │         │
│(1/2  │                                       │      │       Page
│inch) │                                       │      │      Length
│      │                                       │      │    (11 inches)
│      │                                       │      │         │
│      │                                       │      │         │
│      │                                       │      │         │
│      ├──────────────────────────────────────┤      │         │
│      │      Bottom Margin (1-1/3 inch)        │      │         │
│      │                                       │      │         │
│      │   03-22-1983 Book Report Page 1       │      │         ▼
└──────┴──────────────────────────────────────┴──────┘
```
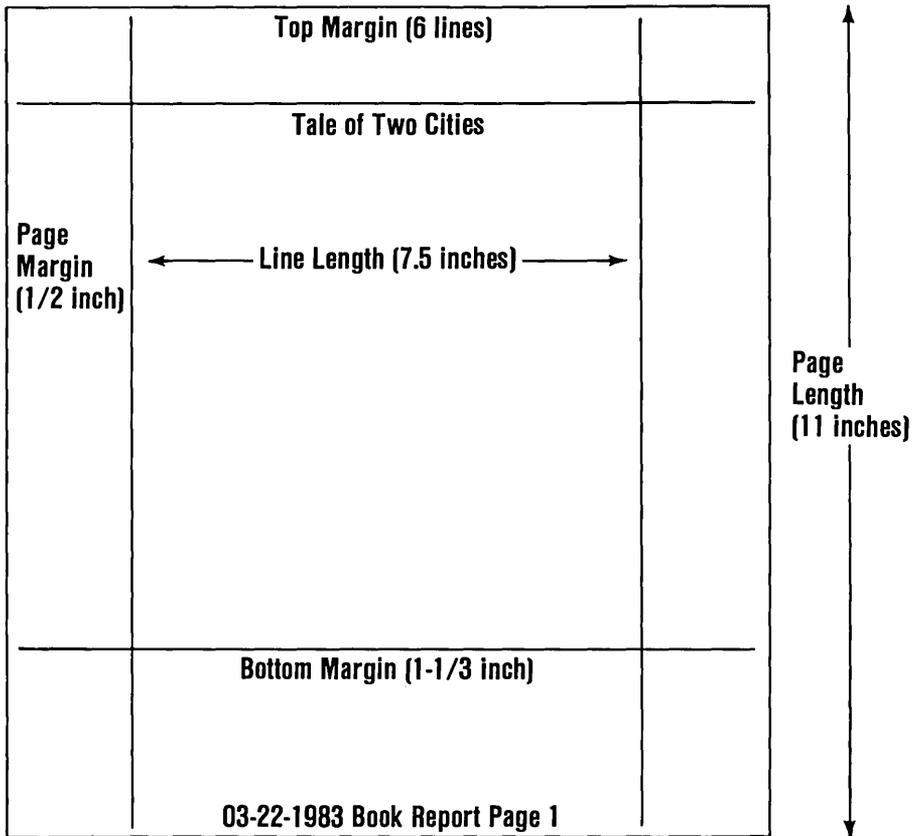
Step 2 – Convert your design to SCRIPT/PC values.

The document just defined has a:

1. Page Length of 66 lines:

   11 inch paper printed at 6 lines per inch

2. Line Length of 75 characters

   7.5 inches X 10 characters per inch (each character = .1 inches)

3. Page Margin of 5 characters - Left margin (binding)

   5 characters X .1 inch = .5 inches

4. Top Margin of 6 lines - Space at top of page

   6 lines = 1.0 inches

5. Bottom Margin of 8 lines - Space at bottom of page

   8 lines = 1.33 inches

6. Running Heading (centered) text of "Tale of Two Cities"

7. Running Title on the bottom of the page on line 3. The date is on the left side. The text "Book Report" is centered and the page number is on the right side of the paper.

Step 3 – Write the SCRIPT/PC control words.

The following example shows how the SCRIPT/PC
control words are to be entered to meet the
requirements of the page you just designed.

```
.PM 5
.PL 66
.LL 75
.TM 6
.BM 8
.RH ON
.CE Tale of Two Cities
.RH OFF
.RT BOTTOM 3 /&SYSDATE./Book Report/&./
```

Detailed descriptions of all of the control words that
affect page layout are contained in *Book 3 - Reference.*

Chapter 7. How to Use Your Printer

Contents

Introduction To Printing

SCRIPT/PC allows you to use a variety of printer fonts to take full advantage of the printing capabilities of the IBM PC Graphics and the IBM PC Matrix printers. This chapter shows you how to set up a selection of different printer characteristics and how to use them. Examples are included to assist you in determining which style of print you might wish to select for a given function. Any restrictions to font use are also pointed out.

SCRIPT/PC initially sets up the printer for printing by choosing:

- The Graphic Printer Character Set 1 — also used by the IBM Matrix Printer.
- A character font that is spaced at 10 characters per inch.
- Bidirectional printing.
- 6 lines of print per inch.

SCRIPT/PC allows you to easily set up or change the previous printer default values. One of the easiest tasks you can learn is how to highlight areas of your normal text. You can also change a number of other printer characteristics (options) that allow you to:

1. Select a print character set that gives you the group of characters you need.

2. Select the number of lines per inch to be printed.

3. Select direction of printing.

4. Select an available print font style.

5. Select a way to highlight a desired print font.

6. Select a method to emphasize (make bold) a desired print font.

Using Your Printer

Your printer options can be changed by defining new print characteristics. You may define up to 15 different sets of printer options for use in selected parts of your document. You can change your printer options for a whole document, a chapter, a page, a phrase, a word or a character within a word. This allows you greater flexibility in your document design. It is the purpose of this section to show you how to use and change these options.

SCRIPT/PC printer options allow you to make your documents more professional looking with items like bold headings, emphasized words or phrases, and large font charts or graphs that can be reproduced for overhead projection. It also gives you a wide selection of styles to be used when writing complex documents such as novels or technical documents.

How to Highlight Text

One of the most common tasks is the necessity to highlight words or phrases within your normal text, from letters to technical documents. Highlighting options available are underlining, capitalizing, and a combination of both. In SCRIPT/PC, there are two steps necessary to do the highlighting. You must:

1. Define the highlights you want to use.
2. Select the highlight you want at the appropriate place in your document.

Defining a Highlight

To define a highlight, you must change the characteristics of your existing font by defining a new font that will include the highlighting characteristics you need. To define a new font or change

characteristics of a font, use the Define Font [.DF] control word provided by SCRIPT/PC. Let's look at the format of this control word and its options:

```
.DF font-id ⎡ US ⎤ [ FONT definitions ]
            ⎢ UP ⎥
            ⎣ UC ⎦
```

Where:

.DF Specifies the define font control word for recognition by SCRIPT/PC.

font–id Specifies the name of a set of printer options. Up to 15 names are allowed.

The following is a list of the printer options as they must be written and an explanation of the option:

US Specifies underlining (underscoring).

UP Specifies that all alphabetic characters will be capitalized (uppercase).

UC Specifies that all characters are to be both underlined and capitalized.

Note: The large brackets mean you can select one or more of the three options listed within the brackets, either US, UP, UC. The FONT definitions options will be explained later in this chapter. First you are going to learn how to highlight your normal (default) font.

Let's define a highlight. We will be using the default font selected by SCRIPT/PC.

If you want to set up a printer option to underline (underscore) your text, you will have to:

1. Select a new name (font-id) for a define font selection, for example, UNDER.
2. Select the underscore option (US) from the options selection part of the Define Font control word format.
3. Type the new definition into your input file:

```
.DF UNDER US
```

Selecting a Highlight

To start highlighting at the text location you wish, you must use the Begin Font [.BF] control word. For example:

```
.DF UNDER US
.
.
.
.BF UNDER
Important:
```

To stop the font-id named UNDER, choose another font-id with a Begin Font [.BF] control word or go back to the previous font, (in this case, the default font) by using the Previous Font [.PF] control word:

```
.DF UNDER US
.
.
.
.BF UNDER
Important:
.PF
This form is for financial assistance.
```

In this example, only the word Important: will be underlined (underscored).

Note: There is no error checking of the define font control word. Error checking is done on the begin

font control word. If the font selected by the begin font control word contains an error, an error message is created.

Placing Your Defined Fonts Into a File

For ease of use, it is easier to define the three highlighted text font-ids in a separate file. This allows you to define them only once, yet have them available for use in any input file. For example:

1. To create the highlighted text combinations, use your editor to type the following into a file, for example, FONTS.SCT:

```
.DF UNDER US
.DF UPPER UP
.DF UPUN  UC
```

> **Note:** The FONTS.SCT file you create will be referred to during the rest of the chapter.

2. To be able to use the FONTS.SCT file in any input (source) file, use the Imbed [.IM] control word as follows. This should be done prior to using any of the new font-ids:

```
.IM FONTS
```

3. The font-ids contained in FONTS.SCT will be imbedded into your input file and will be available for selection:

   a. To start underscoring in your input file, type:

   ```
   .BF UNDER
   ```

   b. To start capitalizing in your input file, type:

   ```
   .BF UPPER
   ```

   c. To start both capitalizing and underscoring in your input file, type:

Note: Remember, you can also change to the previous font-id by using the Previous Font [.PF] control word.

Other Printer Options

Now that you can see how to highlight print for the normal (default) font, let's look at the "FONT definitions" part of the Define Font [.DF] control word. You will be using this section to change printer options and create new font styles.

We will also take a look at defining more than one option for a font-id. There are a few rules when defining a font-id. They are:

1. List the highlighting option (if needed) before the *FONT* option.

2. Select only one option from each group of options.

> **Note:** Some options have restrictions. Be sure to review the topic "Printer Font Restrictions" at the end of this chapter.

Lets look at the definitions available and their groups:

FONT definitions

Where:

definitions   Specifies the options available for your selection. You may select one option from each of the following *option groups* if you want to define a combination font-id. Each available definition is listed as it must be written, It is followed by a description of the definition:

Character Sets

C1   This selects the graphics printer character set 1.
SCRIPT/PC selects this option (default).

C2   This selects the graphics printer character set 2.

Printing Direction

SCRIPT/PC selects bidirectional printing (default).

BD   Bidirectional - prints as the print mechanism
moves in either direction, left to right or
right-to-left. This gains speed of printing as there
is no waiting for a "carriage return."

LR   Unidirectional - prints only as the print
mechanism moves from left-to-right. Does not
print when it returns from right-to-left. This is
similar to a standard typewriter style of printing.

Lines Per Inch

The assumed (default) printer setting is 6 lines per inch
(L6).

L6   Selects 6 lines of print per inch

L8   Selects 8 lines of print per inch.

Fonts

CM   Compressed Characters are selected. This font
creates very narrow characters.

DW   Double Width Characters are selected. This font
is twice as wide as the default print characters.

This font can be used with format on (.FO ON), although it may cause the right margin to lose justification. It is recommended that format off (.FO OFF) be used, because it is easier to use.

SP Superscript Character Set is selected. This font prints small characters at the top of the print line.

SB Subscript Character Set is selected. This font prints small characters at the bottom of the print line.

Emphasizing Text

DS Double Strike Print - bold

EM Emphasized Print - very bold

Selecting Printer Options

To create a font using the compressed characters option, type:

```
.DF COMP FONT CM
```

To highlight, underline, the compressed characters font, type:

```
.DF COMP1 US FONT CM
```

> **Note:** The font-id name you choose should be a name you can easily remember.

How to Create a Printer Fonts File

Lets take the file we started with (FONTS.SCT) and make it larger to contain more examples of commonly used font definitions. In the following example of FONTS.SCT, an example set of definitions has been

created. The double width and compressed fonts also include highlighted forms. This file should allow you an ample selection of fonts and use of multiple options. It also includes the SCRIPT/PC default font highlighting.

The lines that start with .* are comment lines that will be ignored by SCRIPT/PC. The comment lines are frequently used to provide information to someone editing the file, either to the person who created the file or someone else. It is a nice technique to add notes to a file.

```
.**************************
.*  SCRIPT/PC Font    1-3 *
.**************************
.DF UNDER US
.DF UPPER UP
.DF UPUN  UC
.**************************
.*   Compressed Font  4-7 *
.**************************
.DF COMP   FONT CM
.DF COMP1 US FONT CM
.DF COMP2 UP FONT CM
.DF COMP3 UC FONT CM
.**************************
.*   Double Width Font 8-11*
.**************************
.DF DBL   FONT DW
.DF DBL1 US FONT DW
.DF DBL2 UP FONT DW
.DF DBL3 UC FONT DW
.**************************
.*    Superscript Font  12 *
.**************************
.DF SUP FONT SP
.**************************
.*    Subscript Font    13 *
.**************************
.DF SUB FONT SB
```

If you imbed this version of FONTS.SCT into any of your input files, you will have 13 font-ids to select plus the two SCRIPT/PC default fonts. This selection will be available for use by using the Begin Font [.BF] control word.

If you want a different set of fonts, define them only if you delete a previous font. You are allowed to define a maximum of 15 font-ids. The best way to delete a previous font is to use the same font-id to define a new set of printer options. In this way, you will never have more than 15 names.

How to Use Defined Fonts

The fonts you have defined may be used for a variety of purposes. The following examples illustrate how you might use the fonts created in the file FONTS.SCT.

- The following example shows how phrases may be highlighted using capitalization:

```
.IM FONTS
.FO ON;.LL 40
Mary had a;.BF UPPER;little
.PF;lamb.
Its fleece was white as
.BF UPPER;snow.;.PF
Every;.BF UPPER;.CT where;.PF;
that Mary went, the lamb was sure to go.
```

results in:

```
Mary had a LITTLE lamb. Its fleece was
white as SNOW.    EveryWHERE that Mary
went, the lamb was sure to go.
```

Notice that the .CT (Continue Text) control word is used to close the text, so that you can highlight one or more characters within a word. You should be familiar with the other control words by this time.

- The following example shows you how to use the Double Width fonts. This is helpful when you are preparing reports needed for overhead projection.

Since each character in Double Width font takes up 2 character positions, you must only place half as many characters on an input line.

```
.IM FONTS
.FO OFF;.LL 40
.BF DBL1
.CE ON
College History
.CE OFF;.SP 2
.BF DBL
o Founded 1892
.SP
o Privately Endowed
.SP
o Church Affiliated
.SP
o Independent Now
.SP
o Business Oriented:
.SP
  o Bachelors Degree
.SP
  o Masters Degree
```

results in:

```
College History

o  Founded  1892

o  Privately  Endowed

o  Church  Affiliated

o  Independent  Now

o  Business  Oriented

o  Bachelors  Degree

o  Masters  Degree
```

Now that you have been exposed to examples on how
to use printer options, you may want to experiment
further with other fonts and combinations. A very
handy place to do this is in a practice session. Create
your FONTS.SCT file with an editor, set up the printer
with your Options Profile for the practice session and
then experiment with the variety of fonts set up in the
FONTS.SCT file. When you are confident that you can
now control your printed output, you have reached
another plateau in your understanding of the
SCRIPT/PC text processor.

Printer Font Restrictions

The following list shows the combination of fonts that
are not allowed:

• Emphasized Print (EM) is not allowed with
  Compressed Characters (CM).

- Double Strike (DS) is not allowed with the Subscript (SB) or Superscript (SP) character sets

- Subscript Character Set (SB) is not allowed with the Superscript (SP) Character Set.

    Note: When you have completed printing a document, removed the paper and are ready to end SCRIPT/PC, you may notice a delay in returning to DOS if the printer has been left "offline". This is normal operation. The delay will be reduced considerably if you make the printer ready and online prior to returning to DOS.

Where Do You Go From Here

This book has attempted to introduce you to the world of text processing by giving you a chance to practice with SCRIPT/PC commands (both descriptive and declarative). It is not an in-depth training course in all the techniques possible with this product. The intent has been to give you a basic understanding of text processing, enough to create a smooth beginning.

If you have done the practice sessions and have read and understood the chapters on how to use the formatting commands, you should have gained a great deal of knowledge about formatting a document. The additional chapters on page layout and controlling printing should give you an appreciation of how to set up your final output.

This book has prepared you to get started. Book 2 shows you all the control words and tags associated with SCRIPT/PC. Right now, there is a big gap in your knowledge about all the other functions shown in Book 2 that you have not been taught. However, as your experience increases and your curiosity is aroused, you will start to reach for those "other" control words or tags. Book 3 contains the detailed description of

each of the control words and tags. Book 3 will become your main reference manual for all the detailed information you may want to learn from time to time.

If you are using SCRIPT/PC for business, school, or home, enjoy it!

Index

Special Characters

See Period (.)

.AP
See Append control word

.BF
See Begin Font control
word

.BM
See Bottom Margin control
word

.BR
See Break control word

.BX
See Box control word

.CE
See Center control word

.CL
See Column Width control
word

.DD
See Define Data File-id
control word

.DF
See Define Font control
word

.EF
See End Of File control
word

.FO
See Format control word

.IL
See Indent Line control
word

.IN
See Indent control word

.LL
See Line Length control
word

.MC
See Multicolumn control
word

.OF
See Offset control word

.PF
See Previous Font control
word

.PL
See Page Length control
word

.PM
See Page Margin control
word

.RH
See Running Heading
control word

.RT
See Running Title control
word

.SE
See Set Symbol control
word

.SK
See Skip control word

.SP
See Space control word

.TB
See Tab Setting control
word

.TE

A

L

M

description of 5-15
up-arrow key
   using, options profile 2-13

Y

Year
   See System Year

IBM

The Personal Computer
Software Library

Reader's Comment Form

SCRIPT/PC   1502414
Book 1 How to Use

Your comments assist us in improving the usefulness of
our publication; they are an important part of the input
used for revisions.

IBM may use and distribute any of the information you
supply in any way it believes appropriate without
incurring any obligation whatever. You may, of course,
continue to use the information you supply.

Please do not use this form for technical questions
regarding the IBM Personal Computer or programs for
the IBM Personal Computer, or for requests for
additional publications; this only delays the response.
Instead, direct your inquiries or request to your
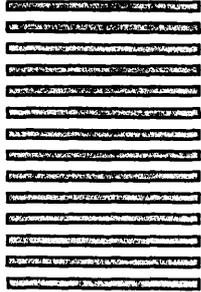authorized IBM Personal Computer dealer.

Comments:

BUSINESS REPLY MAIL

FIRST CLASS   PERMIT NO. 321   BOCA RATON, FLORIDA 33432

POSTAGE WILL BE PAID BY ADDRESSEE

IBM PERSONAL COMPUTER
SALES & SERVICE
P.O. BOX 1328-C
BOCA RATON, FLORIDA 33432

- -

Fold here

Please do not staple   Tape   Tape

Software included:

Diskette

System requirements:

IBM 80-column
display

128KB of memory
One diskette drive

IBM Printer