GH20-9158-2
File No. S370-34

Program Product

Document Composition Facility and Document Library Facility General Information

IBM

GH20-9158-2
File No. S370-34

Program Product

Document Composition Facility and Document Library Facility General Information

Program Numbers 5748-XX9
5748-XXE

IBM

This publication was produced using the
IBM Document Composition Facility
(program number 5748-XX9)
and the master was printed on the
IBM 3800 Printing Subsystem.

Second Edition (September 1979)

This is a major revision of, and makes obsolete, GH20-9158-0, and its technical newsletter, GN20-9289.

This edition applies to Release 2 of the Document Composition Facility and the Document Library Facility, Program Products **5748-XX9** and **5748-XXE**, and to any subsequent releases until otherwise indicated in new editions or technical newsletters. It incorporates information formerly contained in Document Library Facility: General Information, GH20-9163-0, and its technical newsletters GN20-9274 and GN20-9290.

The changes for this edition are summarized under "Summary of Amendments" following the preface. Because the technical changes in this edition are extensive and difficult to localize, they are not marked by vertical bars in the left margin.

Changes are periodically made to the information herein; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 Bibliography, GC20-0001, for the editions that are applicable and current.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming or services which are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming or services in your country.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM Corporation, P.O. Box 50020, Programming Publishing, San Jose, California, U.S.A. 95150. IBM may use or 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.

This manual provides general information about the Document Composition Facility and the Document Library Facility program products. It is intended for the following audiences:

- The management and staff of groups that have potential users of the Document Composition Facility and the Document Library Facility - groups that prepare text documentation or publish such documentation
- The data processing management and staff who plan to install and maintain the Document Composition Facility and the Document Library Facility

This manual will help potential users:

- Evaluate the applicability of the Document Composition Facility and the Document Library Facility for their organization
- Evaluate the programming and machine requirements for using both of these products

This manual is arranged in the following chapters:

"Introduction" on page 1: Explains the term text processing and describes the text processing and document storage capabilities provided by the Document Composition Facility and the Document Library Facility.

"The Document Composition Facility" on page 7: Describes the Document Composition Facility and what it can do.

"The Generalized Markup Language" on page 15: Explains what generalized document markup is and what its application benefits are.

"The Document Library Facility" on page 19: Describes the Document Library Facility and what it can do.

"Document Compatibility and Conversion" on page 25: Describes the relationships between releases of these products and the Advanced Text Management System (ATMS).

"Operating Environment" on page 27: Identifies the programming and machine requirements for installing and using these program products.

"Publications" on page 29: Describes the publications that support these program products.

RELEASE 2

This revision documents the functional changes that have been made to the Document Composition Facility and the Document Library Facility for Release 2. It includes information that was formerly contained in Document Library Facility: General Information, GH20-9163-0.

LIST OF ILLUSTRATIONS

The Document Composition Facility (DCF) and the Document Library Facility (DLF) can assist you in handling your text processing applications.

- The Document Composition Facility provides a text formatter, called SCRIPT/VS. SCRIPT/VS can process documents marked up with Generalized Markup Language (GML) tags as well as those marked up with its own control words. (GML tags are a type of text markup that describes the parts or elements of the document being formatted rather than its specific format.)

  With the optional Foreground Environment Feature installed, the Document Composition Facility can be run in the interactive environments of VM/370 CMS, OS/VS2 TSO, and ATMS-III. It can also be run in the background environment under OS/VS2 MVS, OS/VS1, and DOS/VSE if the Document Library Facility is installed.

- The Document Library Facility is a storage facility that stores, archives (stores on a separate sequential data set), and retrieves data in a protected environment. It includes calling sequences that can be used to invoke the Document Library Facility as a subroutine of a user-written program, to call attribute processors to handle data conversion or manipulation, or to invoke user exit routines. As mentioned above, with the Document Composition Facility installed, the Document Library Facility can be used to run the SCRIPT/VS formatter in a background environment.

These products can help you handle your text processing more efficiently. Figure 1 on page 2 illustrates where these products fit in a text processing environment.

WHAT IS TEXT PROCESSING?

Text processing is the preparation of written material. This material could be intended for use as part of a printed document, such as a book or a memo, or as input for a data processing system data base, such as STAIRS/VS. Text processing can either be done manually or through the use of data processing programs.

No matter how the material is going to be used, when preparing a document using data processing programs, the following steps have to occur:

1. The document is created.
2. A text editor, such as the TSO system editor, or a word-processing system, such as the IBM Office System 6, is used to enter the document into a data set or file. (This same text editor or word-processing system is also used to update the material.)
3. Markup is added to the document to indicate how it is to be formatted.
4. In the case of host-attachable editors and word-processors, the document is transmitted to the host along with the appropriate JCL.
5. The document is processed by a text formatter that can properly interpret the control words or markup that has been included.
6. The formatted document is either displayed on a terminal, sent to an output device for printing, added to a data base (such as STAIRS/VS), or returned to the host-attachable text editor or word-processor environment.

TEXT ENTRY AND EDITING

Text entry and editing can be performed in both interactive and background environments depending on the editors being used. For example, data can be entered into a TSO file and edited using either the TSO system editor or the IBM Structured Program Facility-II (SPF-II) editor. Similarly, text can be prepared using a word-processing system and then submitted to the host with the appropriate JCL to invoke the Document Library Facility to store it or the Document Composition Facility to format it. (In this case, both the Document Library Facility and the Document Composition Facility process the data in a background environment.)

There are many text editors and word-processing systems available for entering and editing data. Figure 1 on page 2 illustrates the ones that can be used to produce documents that are going to be formatted by the Document Composition Facility. It also illustrates that many of these same editors and word-processing systems can use the document storage facilities provided by the Document Library Facility (in a background environment).

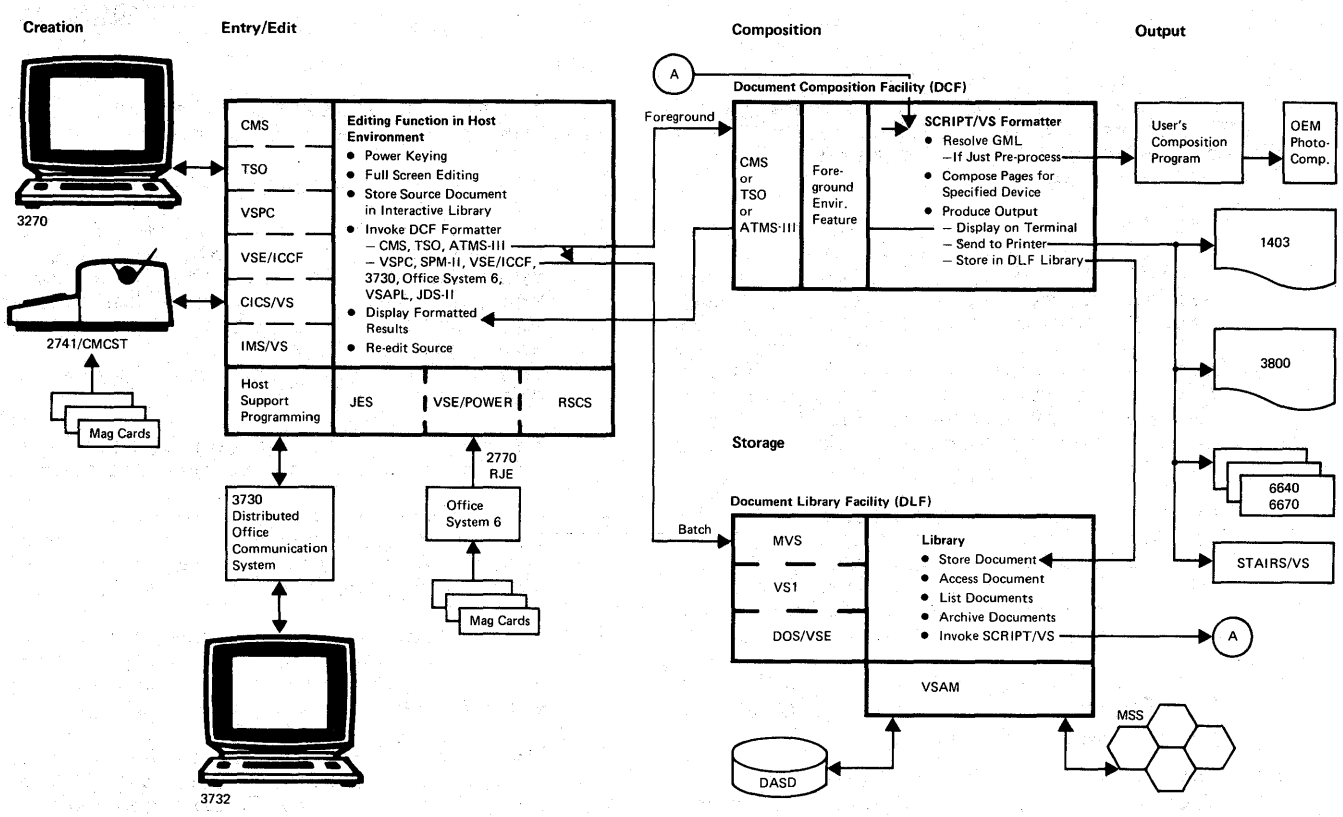


Figure 1. DCF and DLF Overview

TEXT MARKUP

Text markup is information that you add to the text of a document that enables a text processing system to process it in some way. Markup is considered:

- Specific, when you indicate exactly how a document is to be formatted, such as with SCRIPT/VS control words
- General, when you describe the specific elements of a document without specifying how these elements are to be formatted, such as with Generalized Markup Language (GML) tags

With general markup, during the formatting process, each of the GML tags is mapped to a series of SCRIPT/VS control words, symbols, and macro instructions called application processing functions (APFs). This enables you to change the format of an element of a document without changing the GML tag; only the APF that the tag is mapped to has to change.

General markup is designed to be more natural to use than specific markup, because thinking in terms of the content or purpose of the elements of a document is more natural than thinking in terms of how the elements should be processed or how they should appear when printed.

For example, a heading such as the one which begins this chapter could be created using a series of explicit SCRIPT/VS controls words, such as

```
.cp
.uc 1
.bf GB12
Introduction
.pf
.sp 2
```

However, instead of using this series of SCRIPT/VS control words, you could specify the single GML first level heading tag

```
:h1.Introduction
```

to produce the same chapter heading. This GML tag will also automatically generate a table of contents entry for the chapter, along with the proper page number. The APF that is mapped to this GML tag includes all of the SCRIPT/VS control words and symbols needed to produce these results.

Since the general markup tags do not specify a format, as specific markup does, you have more flexibility in choosing the final format for your document than if you had used specific markup. This flexibility enables you to:

- Enter and edit a document without regard to how it is going to be formatted.
- Change aspects of a document's format, such as the amount of indention in a list, by changing the APF that the GML tag is mapped to. (With specific markup, you would have to go through the entire document and make the change everywhere there was a list.)
- Standardize formats for different types of documents without requiring users to know what these standards are.
- Transfer a document to another location that uses GML tags without worrying about what style of formatting they use. As illustrated in Figure 5 on page 17, if they are not using your style, all they have to do is map the GML tags to APFs that reflect their formatting style before processing the document. (With specific markup, they would have to go through the entire document changing all of the control words that do not correspond to their formatting style.)

GML tags are discussed in more detail in "The Generalized Markup Language" on page 15.

TEXT FORMATTING

The processing formatter formats the text according to its interpretation of the markup that has been included. How a document is going to be used often determines how the document is marked up. For example:

- If you are going to display it on a terminal, you might want the left and right margins to line up with the left and right sides of the screen.
- If you are going to have it processed on a printer, you must be sure that the format you want can be produced by available printers.
- If you are going to use it as input to a data base, its format must adhere to special format requirements for that data base. (For instance, STAIRS/VS requires its input to be in Condensed Text Format.)

Therefore, in selecting a type of text markup, you must be sure that it can be interpreted to perform the type of applications you require.

BENEFITS OF USING A COMPUTER TO DO TEXT PROCESSING

Text processing programs can be of great assistance in producing documents. Some of the typical benefits include:

- Experimentation with different formats. It is possible to experiment with different document formats without altering a document's text or markup. (This is difficult to do using a manual method of text processing.)
- Captured keystrokes. Text that is entered once need not be reentered for subsequent drafts. Only changes or additions have to be entered.
- Reduced proofreading. Only changed portions of text need to be proofread, because the computer can reliably reproduce the unchanged portions.
- No separate composition for final copy. The make-up of pages for final printing is a direct result of the process of document creation and markup.
- Shorter time to produce finished material. Time savings come from all of the process savings mentioned above.
- Same text usable for multiple documents. Text stored in files or data sets can be included in multiple documents without having to maintain multiple copies of it.
- Same text usable for other text applications. Text stored in files or data sets can be searched by programs other than text processing programs.

COMPUTER ENVIRONMENTS FOR TEXT PROCESSING

To effectively use data processing equipment in text processing, you must have access to:

- A text editor or word-processing capability for entering and updating data

- A facility, such as a library, for storing and accessing the data
- A text formatter for formatting the data
- A device for displaying or printing the output

Depending on the types of documents being processed, more than one type of editor, as well as a variety of storage facilities and output devices, may be required.

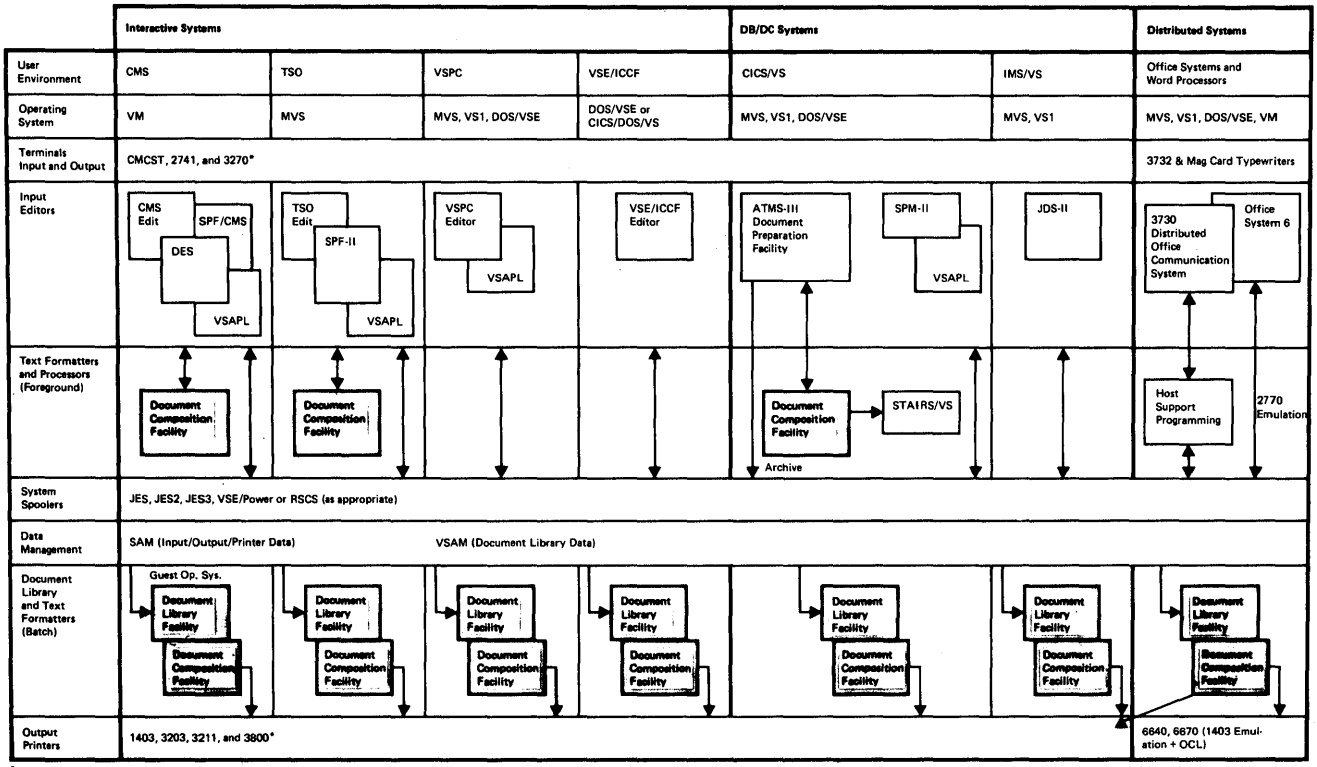
The Document Composition Facility and the Document Library Facility are capable of handling input from different sources and can submit output to a variety of output devices for final processing as illustrated in Figure 2 on page 5.

The editors that can be used with the Document Composition Facility in an interactive environment include:

- TSO, CMS, and VSPC System Editors: TSO, CMS, and VSPC each provides a set of editing facilities for text entry and editing as part of the base system.
- Advanced Text Management System-III (ATMS-III): ATMS-III provides a set of text-oriented entry and editing facilities. These facilities can be used to prepare documents for formatting by SCRIPT/VS, thus making SCRIPT/VS functions available to the ATMS-III user. In addition, these facilities make it possible for the ATMS-III user to use the Document Library Facility to archive ATMS-III documents.
- Structured Programming Facility-II (SPF-II), 5740-XT8, under either TSO or CMS: SPF-II includes text entry and editing capabilities beyond the base TSO and CMS editing facilities. This includes such things as taking greater advantage of the full-screen editing functions of the IBM 3270 Information Display System, and being able to simultaneously view a source document and its formatted results on a split screen.
- Display Editing System (DES), 5796-PJP, under CMS: DES includes text entry and editing capabilities beyond the base CMS editing facilities. This includes such things as taking greater advantage of the full-screen editing functions of the IBM 3270 and being able to simultaneously view a source document and its formatted results on a split-screen.

If the Document Library Facility is installed, it can also accept input that has been processed by the following host-based or host-attachable text editors and word-processing systems and submitted as a batch job in a background environment:

- VSE/Interactive Computing and (see Interactive Computing and Control Facility) Control Facility (VSE/ICCF), 5746-TS1, in DOS/VSE: VSE/ICCF's full-screen editor can be used to enter and edit text. The document can then be submitted as part of a job that invokes SCRIPT/VS. The resulting formatted output is routed back to the VSE/ICCF environment for viewing on a terminal or for printing.
- Source Program Maintenance On-Line-II (SPM-II), 5798-CFT, in CICS/VS: SPM-II can be used to enter and edit text. The document can then be submitted as part of a job that invokes SCRIPT/VS. The resulting formatted output is routed back to the SPM-II environment for viewing on a terminal or for printing.
- Job Development System-II (JDS-II), 5798-CLP, in IMS/VS: JDS-II provides a full-screen editing facility that can be used to enter and edit text. The document can then be submitted as part of a job that invokes SCRIPT/VS. The resulting formatted output can be routed back to the JDS-II environment for viewing on a terminal or for printing.
- 3730 Distributed Office Communications System, 5740-XY9: This system provides facilities for producing text that can be submitted to the host as part of a job, via remote job entry (RJE). A user-written program can place the document into a data set or file for subsequent host processing. With additional user-written programming, the document can be stored in the Document Library Facility library and storing data formatted by SCRIPT/VS.
- Office System 6 and Mag Card Typewriters: These systems provide facilities for producing text on magnetic cards or diskettes. This text can then be submitted to the host as part of a job via remote job entry (RJE), where it is stored in the Document Library Facility library for formatting by SCRIPT/VS. The output can be routed to any IBM product that utilizes the IBM 2770 bisync protocol. (It can also be routed to a system printer as long as that printer has been configured into the system as an IBM 2741 or 1403.)



* and other devices compatible at the data stream level and supported by the user environment

Figure 2. Available Operating Environments

The Document Composition Facility is a text processing program that is used in the preparation of printed materials. This program consists of a text formatter, SCRIPT/VS, that can format documents that include SCRIPT/VS control words and Generalized Markup Language (GML) tags along with the text. Once processed by SCRIPT/VS, these documents can be printed at any of a variety of output devices or used as input to the Storage and Information Retrieval System/Virtual Storage (STAIRS/VS) program product.

This program includes an optional Foreground Environment Feature which enables you to use SCRIPT/VS with:

- OS/VS2 TSO (Time Sharing Option). SCRIPT/VS runs in the TSO foreground. TSO's interactive editing (EDIT or SPF-II) and related facilities provide the text entry and editing and data set management that are used with SCRIPT/VS for text processing.
- VM/370 CMS (Conversational Monitor System). CMS's text editors, file system, and related facilities provide the text entry and editing, and file management facilities that are used with SCRIPT/VS for text processing.
- ATMS-III (Advanced Text Management System-III). The ATMS-III editor in a CICS/VS environment provides the text entry and editing facilities that are used with SCRIPT/VS for text processing.

To use SCRIPT/VS in a background environment (OS/VS1, OS/VS2 MVS, or DOS/VSE), you must install the Document Library Facility. (This product provides text storage and management for the documents SCRIPT/VS processes.) In a background environment you can also use:

- The ATMS conversion routine (that is provided with the Document Composition Facility). This conversion routine can be used to convert most explicit and implicit ATMS-II and ATMS-III markup to similar or equivalent SCRIPT/VS markup.
- SCRIPT/VS as a subroutine of a user-written program.

In a background environment, you can format documents that have been: communicated to the host system from offline word-processing systems, such as magnetic card typewriters or the IBM Office System 6; created in a data communication system, such as Job Development System-II (JDS-II); or created in an interactive system, such as VSE/Interactive Computing and Control Facility (VSE/ICCF), and have been submitted with JCL as a job. (See Figure 2 on page 5 for a summary of all of the text editors and word-processing systems supported by the Document Composition Facility.)

RELEASE 2 ENHANCEMENTS

Release 2 of the Document Composition Facility includes functions that:

- Enable SCRIPT/VS, with the Foreground Environment Feature installed, to operate under the control of ATMS-III in a CICS/VS environment.
- Enable SCRIPT/VS to produce formatted output suitable for input to the STAIRS/VS program product.
- Enable indexes to be automatically generated from index entries specified within the text at points of reference. Page numbers for these index entries are automatically generated.
- Provide dictionary and stem processing for hyphenation and spelling verification in United Kingdom English, Canadian English, Canadian French, Dutch, French, German, Italian, and Spanish, as well as in American English.
- Enable SCRIPT/VS users to build their own dictionaries which can be concatenated with the base dictionary in user-defined order.
- Enable a user to provide an algorithmic hyphenation routine that can be invoked to hyphenate a word if it is not to be hyphenated using the base dictionary. IBM provides an algorithmic hyphenation routine for English with this release of the product.
- Extend the Generalized Markup Language (GML) support to accept multiline markup.
- Make it easier to write application processing functions (APFs) when creating additional Generalized Markup Language (GML) tags.
- Extend many of the formatting capabilities of SCRIPT/VS to provide such facilities as widow control and top and bottom column and page floats.

SCRIPT/VS FUNCTIONS

The main function of SCRIPT/VS is to format text for display or printing on both impact and nonimpact printers, such as the IBM 3800 Printing Subsystem. SCRIPT/VS provides the capability to produce flexible composition for printing on a computer printer as an alternative to using independent typesetting machines or outside printers. (With the IBM 3800 Printing Subsystem, SCRIPT/VS can format text in multiple character styles and sizes, called fonts.)

The documents to be processed by SCRIPT/VS can be stored in the storage facility being used with the interactive editor or word-processor or they can be stored using the Document Library Facility. This facility, which is described in more detail in "The Document Library Facility" on page 19, provides document storage control in a secure library while providing authorized users easy access to their stored documents.

Besides being used as a formatter, SCRIPT/VS can be used as a preprocessor to prepare documents for processing by other formatting programs, such as formatters that support photocomposers. It can also be used to format a document so that the output is suitable for use as input to STAIRS/VS.

SCRIPT/VS can process documents that have been marked up for SCRIPT/370 Version 3 with little or no changes required to the markup in these documents. (For more information on the compatibility between SCRIPT/VS and SCRIPT/370, see the Document Composition Facility: User's Guide)

SCRIPT/VS processing can be divided into two general categories of functions; formatting, and general document handling.

FORMATTING FUNCTIONS

The formatting functions provide:

Page Layout: You can control the following aspects of page layout:

- Line Formatting. How input lines are processed for output; formatted or unformatted, centered, aligned left, aligned right, justified (the output lines are the same length), ragged-right, or ragged-left. (With ragged-right and ragged-left, the output lines can vary in length).
- Line Spacing. The amount of space between output lines, including the space left for illustrations or artwork that cannot be produced using your data processing equipment.
- Paragraph Formatting. The style of the paragraphs in your documents. For example, the amount of space between paragraphs and whether or not the first line is indented and, if so, how much.
- Fonts. Which font is used for different portions of text, both in the body and in the running headings and footings.
- Columns. The number of columns on each page (up to nine), the size of each one, and its placement on a page.
- Margins. The size of the top and bottom margins. Title lines can be defined that will be put into the top or bottom margins of even- and/or odd-numbered pages.
- Indention. The extent and duration of indention. This includes hanging indentions (a paragraph in which the indention of the first line is unchanged and subsequent lines are indented to a specified value), left or right margin indentions, the size of the indentions, and the number of lines to which the indentions apply.
- Headings and Footings. The placement of running headings and footings within your document. Running headings and footings can include such things as chapter or section titles, document titles or form numbers, page numbers, or security classifications, and can be emphasized using capitalization, underscoring, overstriking, and multiple fonts. They can also be different on even- and odd-numbered pages or completely suppressed for a series of pages.
- Reference Numbers. Whether or not line reference numbers are placed on any page of a document. (If reference numbers are requested, all nonblank lines in the body of the specified page will be serially numbered.) These numbers will appear in the binding to the left of the first column on the page.

Figure 3 on page 10 shows some sample page layouts that are different from the one used in this manual.

Head Levels: You can specify up to seven head levels for distinctive formatting of headings for different levels of topics, including before and after spacing, font selection, capitaliza-

tion, underscoring, and right or left alignment. For example, the run-in heading of this paragraph has a head level of 5.

Table of Contents: You can control whether or not a table of contents is automatically generated and where it is placed. SCRIPT/VS collects entries for a table of contents from the head levels in the text and supplies the page numbers. You can also specify phrases other than the text of head levels to appear in the table of contents. The table of contents of this manual was automatically generated by SCRIPT/VS.

Indexes: You can control whether a back-of-book index is to be generated. If an index is to be generated, you can include index entries in the body of your document at the points of reference. (These entries do not appear in the body of the document when it is formatted.) SCRIPT/VS will use these entries to generate the index and the appropriate page numbers for the index entries.

Highlighting Phrases: You can control how phrases are to be highlighted for emphasis. For devices that do not have multiple fonts, highlighting is done with underscores or capitalization, or both. For devices with multiple fonts, you can change the font for emphasis. For devices that support overstriking, you can also overstrike a phrase to highlight it.

Footnotes: SCRIPT/VS saves text indicated as a footnote and places it at the bottom of the page.[1] Subsequent footnotes are placed below it.

Hyphenation: You can control whether or not words are to be hyphenated at the end of output lines. SCRIPT/VS provides dictionary and stem processing support for the following languages:

- American English
- United Kingdom English
- Canadian English
- Canadian French
- Dutch
- French
- German
- Italian
- Spanish

You can either specify a specific language for a certain document or you can use the default language that has been established for your installation. Since you can specify within a document that a different language is to be used, it is possible for SCRIPT/VS to provide hyphenation support for multilingual documents as long as the languages are included among those that SCRIPT/VS supports.

You can also build additional dictionaries to meet the special requirements of your installation. These dictionaries can then be concatenated together with the base dictionaries supplied by IBM in the order you specify. Once built, these dictionaries can be updated as required.

SCRIPT/VS also allows you to provide an algorithmic hyphenation routine as an alternative for providing hyphenation support to meet the special requirements of your installation. This algorithmic hyphenation routine can be invoked whenever a word cannot be hyphenated using the base dictionary. (IBM provides an algorithmic hyphenation routine for English with the Document Composition Facility.)

Printing Parts of a Document: You can control whether every page of formatted text is included in an output document or only specified ranges of pages.

Tab Handling: You can specify up to 99 tab positions. When formatting output lines, SCRIPT/VS tabs to the right to the specified tab stop.

Box Drawing: You can construct boxes around text (which can still be formatted in the usual ways). You can also draw boxes within boxes and can draw vertical lines to separate columns of text within a box and horizontal lines to separate rows.

Keeping Text Together: You can use one of several methods provided by SCRIPT/VS to keep lines of text together to improve the appearance of output. This ensures that figures and examples are not broken between columns or pages and that headings are not separated from the text that follows them. SCRIPT/VS can also prevent widows, single lines at the beginning or end of a paragraph, from being left by themselves at the top or bottom of a column or page.

Floating Text to the Top or Bottom of a Page or Column: SCRIPT/VS provides a way to designate that blocks of text are to be kept together and floated to the top or bottom of a subsequent output page or column.

Using as a Subroutine: In a background environment, with the Document Library Facility installed, a program can call SCRIPT/VS as a subroutine (to format reports, for example).

---

[1] Like this.

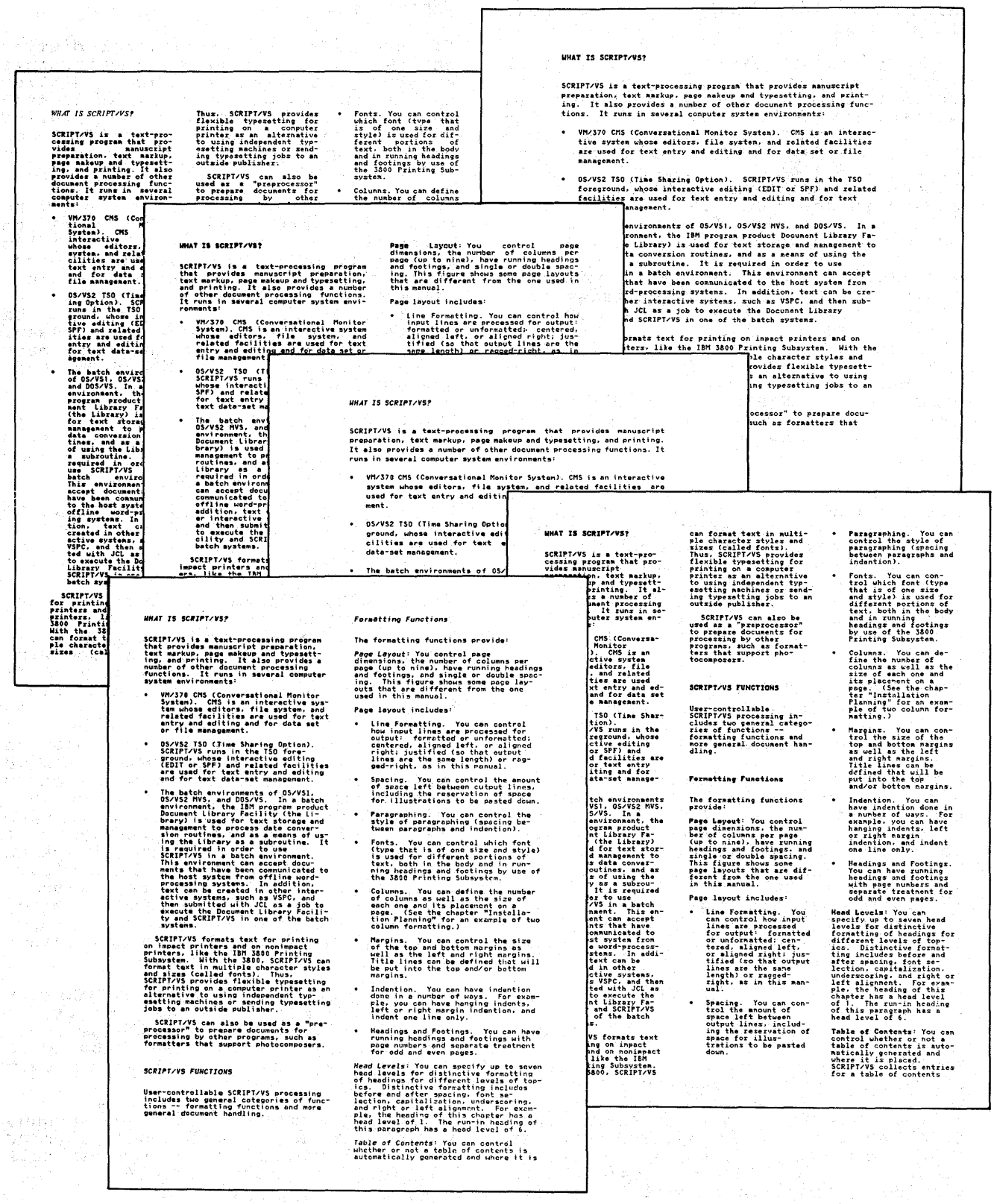


Figure 3. Samples of Alternative Page Layouts

DOCUMENT HANDLING FUNCTIONS

The document handling functions that SCRIPT/VS provides include:

Generalized Markup Language: You can use Generalized Markup Language (GML) to describe the elements of a document rather than SCRIPT/VS control words that indicate how the document is to be formatted. SCRIPT/VS recognizes GML tags as acceptable markup. ("The Generalized Markup Language" on page 15 discusses GML in more detail.)

Saving Input Lines for Subsequent Processing: You can control whether certain input lines will be written to a data set or file for subsequent processing or will be processed immediately.

Revision Codes: You can put up to nine revision codes in the left margin. These revision codes can be used to emphasize lines that may be of particular interest or to indicate lines that have changed since a previous edition of the document.

Spelling Verification: You can control whether or not the spelling of words is checked. Any dictionaries that you add for hyphenation purposes, along with the ones provided by IBM, are used for spelling verification.

Imbedding Separate Files: You can control how separate source documents are brought together for processing as a single document. Any number of source documents can be imbedded in the base source document. A source document that has been imbedded can itself have another source document imbedded in it, and so on, up to eight levels.

Conditional Processing: You can cause SCRIPT/VS to test for conditions that alter processing. For example, the symbol values set by a user might determine whether a block of input text is included in the output document or not. (SCRIPT/VS includes conditional testing as part of its normal processing. For example, it checks the amount of space left in a column before processing certain blocks of text. Much of this conditional processing can also be controlled by the user by defining macro instructions to supplement SCRIPT/VS control words.)

Interaction during Processing: In the CMS and TSO interactive environments, you can communicate with SCRIPT/VS while it is processing a document. In effect, the terminal can be treated as an input file. There are control words that you can place within your document that cause SCRIPT/VS to pause during processing and allow you to enter additional text or control words. This enables you to interactively specify the values of symbolic variables specified in the document or enter portions of text that vary from one formatting process to the next.

Destination of Output: You can control whether the output document is: (1) stored as a file for possible later editing, displaying, or printing, or for use as input to another program, (for example, a formatter that supports a photocomposer); or, (2) printed on the device specified (an impact or non-impact printer, or a display and typewriter terminal), provided the specified device is supported by the host.

Adding Text to Output: If your document is being printed at a typewriter terminal, you can halt the printing and type in additional information. When you have finished making your additions, you can signal SCRIPT/VS to resume its processing. (The information you add will appear in the output just the way it was entered; it will not be processed by SCRIPT/VS.)

Conversion of ATMS Documents: With the IBM Document Library Facility installed, you can use the ATMS conversion routine to convert most explicit and implicit ATMS-II and ATMS-III markup into equivalent or similar SCRIPT/VS markup. (The Document Library Facility provides a calling sequence that the Document Composition Facility uses to obtain the document to be converted. You can use this same calling sequence to obtain documents for other conversion routines that you may want to provide.)

Tracing of Processing Actions for Debugging: You can trace control word processing and the substitution of symbols and macro instructions in input lines. (This can be a valuable debugging aid when undesirable output is received.)

SCRIPT/VS FORMATTING APPLICATIONS

SCRIPT/VS can be used for in-house publishing. In general, this is the publishing of books, manuals, directories, letters, computer-generated Preparing to print a manual like this one is one of several in-house publishing applications SCRIPT/VS can perform. Other applications include:

- Technical Documentation. Technical professionals (such as programmers, engineers, and research scientists) often need to document their work in progress. This documentation takes many forms:

  - Functional specifications

- Engineering specifications
- Design specifications
- Laboratory reports
- Data processing standards manuals
- Technical reports
- Procedure manuals
- Conference papers
- Manuscripts for publication in journals
- Proposals

SCRIPT/VS enables the information in these documents to be developed over weeks or months as a project continues. It also enables many different people to contribute to it during the project's development.

- Administrative Documentation. Many documents produced in offices and administrative centers have the characteristics of publications; for example, department reports, office procedures, lists, directories, and bulletins. They are usually either typed or prepared on a "word-processing" machine of some sort. SCRIPT/VS, used with a word-processing system or an interactive environment that has editing and storage facilities, can be an effective tool in producing these kinds of documents.

- Program-Generated Reports. Printed reports of all kinds are prepared by programs that interrogate data bases to retrieve and accumulate data. The coding to format these reports is usually done in the programming language used for the rest of the program. Most programming languages, however, are not designed for text processing, so they are not always easy to use to format reports.

  In the background environments of OS/VS1, OS/VS2 MVS, and DOS/VSE, SCRIPT/VS can be used by programs to generate reports. A user-written program can use one of the calling sequences provided with the Document Library Facility to pass text and its associated markup from the Document Library to SCRIPT/VS for processing. Thus, the Document Composition Facility can be used as a subroutine of the user-written program.

  The text to be printed need not literally be generated by the program that calls SCRIPT/VS. The text can already exist as a separate document or file that is used by SCRIPT/VS during processing.

SCRIPT/VS APPLICATIONS OTHER THAN FORMATTING

SCRIPT/VS has other applications besides formatting documents. It can be used as:

- A processor of documents that are going to be used as input to STAIRS/VS
- A preprocessor for other text processing programs

SCRIPT/VS AS A PROCESSOR OF INPUT FOR STAIRS/VS

STAIRS/VS (Storage and Information Retrieval System/Virtual Storage) retrieves information from documents it has stored. STAIRS/VS bases this retrieval capability on a multilevel indexing scheme that you use through queries and prompts.

STAIRS/VS requires that documents to be indexed be in Condensed Text Format. When preparing a document for indexing by STAIRS/VS, SCRIPT/VS processes all of the markup that has been included and then formats the document into Condensed Text Format.

SCRIPT/VS can also format documents that will be indexed by STAIRS/VS in a "proof" format. This "proof" format is easier to check for errors than Condensed Text Format and can be used to proofread the document before it is formatted into Condensed Text Format.

SCRIPT/VS AS A PREPROCESSOR

SCRIPT/VS can be used as a preprocessor for other text processing programs, such as formatters that support photocomposers, or a user-written program. The symbol and macro capabilities of SCRIPT/VS can be used to process GML tags and to generate another program's text processing controls. This SCRIPT/VS output can then be processed by another text processing system.

SCRIPT/VS's logic capabilities (including symbols, arrays, condition-testing, branching, and imbedding of other documents) enable people with programming training to program a source document. For example, you can use SCRIPT/VS to write a program that structures text for computer-assisted instruction courses.

APPLICATIONS BENEFITS

For all of the preceding applications, the use of SCRIPT/VS has several benefits that are particularly evident in the following instances:

- When SCRIPT/VS is used with interactive text editors or word-processing systems, manuscript preparation can be easier and quicker than it is with other methods, because these systems provide text entry, editing, and storage facilities. Revisions require minimal rework. Only affected portions of the source document have to be updated. SCRIPT/VS produces the new version with no need for additional layout work, retyping, page numbering, or proofreading.

- Once the text is in a file or data set, it is available for other applications. It can be imbedded into multiple files in any order to produce a variety of different types of documents such as bulletins, memos, or technical reports.

- Copies of the work can be made quickly at each stage of the process:

  - Review copies for the preparation of the manuscript

  - Proof copies for evaluating the layout

  - Camera-ready copy for plate making (or multiple final copies, as printed by the computer)

  The text originally entered in the computer with markup serves as the source for SCRIPT/VS preparation of printed output at every step of the process.

- Final copies can be printed in the precise quantities needed. This retains the original quality for each copy. Additional copies can be printed as needed, rather than stocked.

- Printing by outside publishers and printers can be reduced. With SCRIPT/VS, composition and printing are integrated steps in the publishing process.

- SCRIPT/VS GML is helpful in publishing where consistency of publishing style is important. Besides standardization of style, the other GML benefits apply:

  - The ease and speed of marking up with GML.

  - The use of alternative functions for producing copies for different purposes. For example, early review copies might be printed on an impact printer, or typed or displayed on a computer terminal), and proof copies and final copies might be printed on the IBM 3800 Printing Subsystem, for different character styles and sizes.

- When the document is marked up for conditional processing, SCRIPT/VS can select sections of a document for printing as part of other documents.

- For users of word-processing systems, using the fuller composition capabilities of SCRIPT/VS provides access to such functions as multiple column page formatting, generating an automatic table of contents, generating an automatic index, and forward and backward cross-references. Input can be transmitted to the host system, formatted by SCRIPT/VS, and then sent back to be printed.

- Data from various source documents can be included in one output document.

USING SCRIPT/VS

SCRIPT/VS processing can be controlled by using:

- The SCRIPT command options that are specified when SCRIPT/VS is invoked

- Symbols and macro instructions

- SCRIPT/VS control words

THE SCRIPT COMMAND OPTIONS

The SCRIPT command can be issued in an interactive or a foreground environment. If the Document Library Facility is installed, it can also be issued in a background environment or as part of a user-written program when SCRIPT/VS is being used as a subroutine of that program.

When issuing the SCRIPT command, a number of options can be specified that determine:

- Values for symbols used in a document

- Symbol and macro libraries to be used
- The width of the binding margin
- Whether spelling is to be checked
- Whether the indexing control words will generate an index.
- The type of device or data base the output is being formatted for
- Whether the output is to be stored in a file or data set, or printed
- Ranges of page numbers to be included in the output

If none of the command options are included when the SCRIPT command is issued, SCRIPT/VS will use the default values that apply to the environment in which you are running.

"Appendix A: SCRIPT Command Options Summary" on page 31 summarizes the options of the SCRIPT command and describes the effect they have on SCRIPT/VS processing.

SYMBOL AND MACRO INSTRUCTION PROCESSING

You can define symbols and macro instructions to be used during SCRIPT/VS processing. Symbols can be used:

- In tests for conditional processing
- For cross-references headings or figure numbers
- For entering characters that are not on the keyboard
- As abbreviations for lengthy, repetitive phrases

You use SCRIPT/VS control words to establish the value of a symbol.

Macro instructions are often used to expand the capabilities of the SCRIPT/VS control words. This includes:

- Redefining SCRIPT/VS control words to extend their capabilities
- Defining a single control word to perform a frequently used function that otherwise requires a series of SCRIPT/VS control words, symbols, and macro instructions and some SCRIPT/VS logic
- Defining control words to perform a function that is not provided by SCRIPT/VS control words

Symbols and macro instructions can be defined within a document if they are going to be used only for that document. However, if they are going to be used by multiple documents, it is more efficient to store them in a symbol and macro library. Then, if you need to change a symbol definition or macro instruction, you only have to change it in one place.

Symbols and macro instructions are used to implement GML tags. (GML tags are discussed in more detail in "The Generalized Markup Language" on page 15.)

THE SCRIPT/VS CONTROL WORDS

SCRIPT/VS provides a device-independent, document-processing language that includes more than one hundred control words that control SCRIPT/VS processing. Through the macro facilities of SCRIPT/VS, you can redefine these control words or define new ones. You can also define symbols, which have many uses (for example, cross-references to pages or figure numbers and tests for conditional processing).

If no control words, macro instructions, or symbols are specified to override them, SCRIPT/VS's formatting defaults are used. However, in most cases, control words have to be added in order for the document to adhere to a particular formatting style. These control words can be specified in several ways:

- With the Generalized Markup Language (GML), in which tags are used to mark up a document, so that the actual control words can be located outside the document.
- The control words themselves may be part of the document markup.
- In interactive environments, control words (or other input) may be entered from the terminal when control words in the document cause SCRIPT/VS processing to pause and allow input from the terminal.
- The control words can be issued within a user's program that is using SCRIPT/VS as a subroutine (batch environment only). In this case, the markup is included in the document being passed to SCRIPT/VS by the user's program.

The SCRIPT/VS control words and the functions they perform are summarized in "Appendix B: SCRIPT/VS Control Word Summary" on page 33.

To mark up a source document, you add information to it that enables a person or system to process it in some way. This added information is usually referred to as markup. It consists of "control words" or "controls," or, as in the case of the Generalized Markup Language, "tags." Document markup is the primary means of indicating to a text processing system, such as SCRIPT/VS, how to process a document.

A document can be marked up in either of two ways:

- With specific markup, you indicate exactly how a document is to be formatted, such as with SCRIPT/VS control words.
- With general markup, you describe the specific elements of a source document without specifying how these elements are to be formatted, such as with Generalized Markup Language tags.

WHAT IS GENERALIZED MARKUP LANGUAGE?

The Generalized Markup Language (GML) is markup that identifies the parts of a document rather than its format. For example, a book might have front matter, body, and back matter as its major divisions and within these divisions have subdivisions of chapters, sections, headings, paragraphs, examples, figures, lists, items in a list, and so on. A memo, on the other hand, might have the addressee, date, sender, subject, and reference as its major divisions as well as some of the same subdivisions as a book.

General markup is designed to be more natural to use than specific markup because thinking in terms of the content or purpose of the parts of a document is more natural than thinking in terms of how the parts should be processed or how they should appear when printed. A person doing general markup can concentrate on the text, without thinking about format. Figure 4 on page 16 compares the GML tags with the SCRIPT/VS control words that are required to produce a given two-level list.

BENEFITS OF USING GML FOR SCRIPT/VS TEXT PROCESSING

Some of the benefits in using GML when you are using SCRIPT/VS for text processing include:

- Alternative GML interpretation. A GML tag need not be limited to a single SCRIPT/VS interpretation. For example, a tag might indicate that a group of words in the text is a title. For one application, you might want titles to be underlined, while for another you might want them to be italicized. Each application could be satisfied by alternative GML interpretations, with no change to the source document or to the markup of titles.
- Ease of markup. GML tags are easy to remember because they can consist of terms and abbreviations commonly used to describe a document. Also, GML generally requires fewer characters to be entered for markup than a corresponding sequence of control words. The result is faster markup and keying of the document.
- Ease of text update. Such things as the numbering of items in an ordered list are left to the formatter. Thus, when an item is inserted or deleted, subsequent items do not have to be renumbered as they would have had to be if the numbers had been part of the source text.
- Uniformity of formatting style. Use of GML for all documents of a particular kind results in a single format for all of them without the people doing the markup having to think about the format. Similarly, a change in format for all the documents can be made by changing the way in which the GML is interpreted.

BENEFITS OF GML FOR OTHER APPLICATIONS

For general processing with SCRIPT/VS, or other programs, the benefits of GML include:

- Alternative text processing programs. SCRIPT/VS can be used to convert GML into the processing controls of a text formatting program other than SCRIPT/VS. For example, they might be converted into the controls of a formatter that can be used to produce output on a photocomposer.

Consider the following nested lists:

1. First level item.

   a. Second level item.

   b. Second level item.

2. First level item.

   Following paragraph...

This result can be obtained using either explicit or generalized markup:

Explicit Markup

```
.ti ¬ 05
.tb 4m 8m
.of 4m
1.¬First level item.
.sk
.in +4m
.of 4m
a)¬Second level item.
.sk
.of 4m
b)¬Second level item.
.sk
.in -4m
.of 4m
2.¬First level item.
.of 0
.sk
Following paragraph ...
```

Generalized Markup

```
:ol
:li.First level item.
:ol
:li.Second level item.
:li.Second level item.
:eol
:li.First level item.
:eol
:p.Following paragraph ...
```

Figure 4. An Example of Explicit Versus Generalized Markup

- Alternative output devices. Because GML describes the parts of the document and not its format, it is possible to exploit the formatting capabilities of an alternative output device without changing any of the GML tags; only their interpretation has to change.

- Document exchange. Documents that have been marked up with GML tags can be processed by different groups or locations, regardless of their formatting style. If the formatting style changes, only the Application Processing Functions (APFs) to which the GML tags are mapped have to change.

- Data base applications. GML identifies the parts of documents. Programs can be written alone, or in conjunction with SCRIPT/VS processing, to extract information from documents, based on these parts, for storage in or retrieval from a data base.

HOW SCRIPT/VS INTERPRETS GML

In GML, "interpreting" a tag means performing the correct processing function on the associated text. Every GML tag is associated with an APF or macro that indicates how the text following the tag is to be processed.

Each APF is composed of some combination of SCRIPT/VS control words, symbol definitions, and macro instructions. Which APF is to be used for a given tag in a specific document depends upon the processing that's required. To specify which APF is to be used for a given tag, you include the appropriate mapping of the APF to the given tag in the document profile. Then, when a different type of processing is required, the tag does not have to change; only the mapping of an APF to that tag changes.

Many tags can be associated with the same APF. This will cause the same processing to occur for the text associated with each tag. Figure 5 on page 17 shows this relationship and the relationship that exists between the GML tags in a source document, the document profile, and the macro library containing the APFs that the GML tags are mapped to.

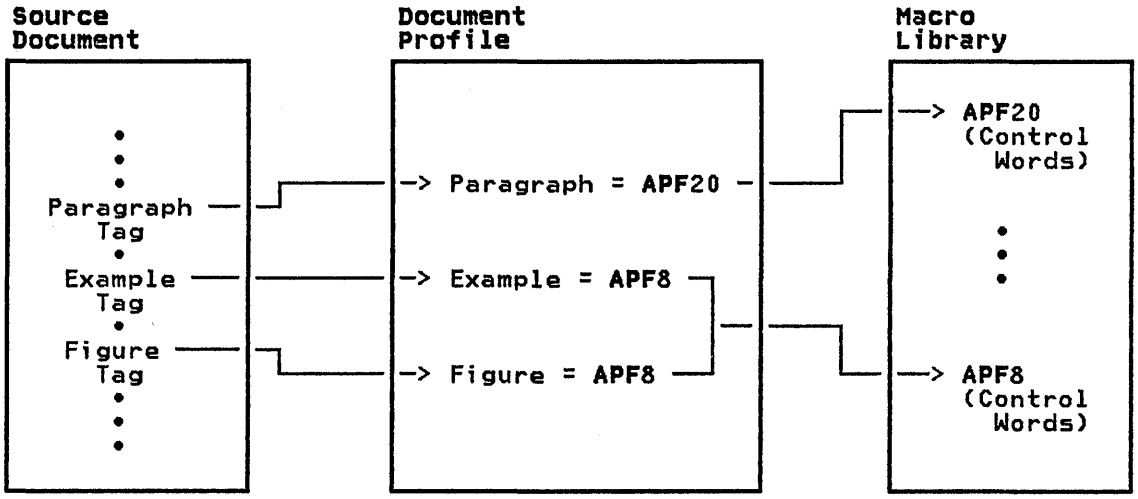


Figure 5. The Relationship Between GML tags, a Document Profile, and APFs

For more information about defining and modifying document profiles and APFs, see Document Composition Facility: User's Guide.

DEVELOPING GML FOR YOUR OWN USE

IBM provides a GML starter set with the Document Composition Facility to help you start using GML. This starter set consists of a collection of GML tags, their associated APFs, and the Document Profile that provides the GML tag-to-APF mapping. These GML tags identify the following parts of a document:

- Title page
- Front matter
  - Abstract
  - Preface
  - Table of contents
  - List of figures
- Body
  - Headings (seven levels)
  - Paragraphs
  - Examples
  - Index entries
  - Ordered lists (numbers or letters)
  - Unordered lists (bullets or dashes)
  - Simple lists (no numbers, letters, bullets, or dashes)
  - Definition lists (terms paired with definitions)
  - Items in a list
  - Quotations
  - Figures
  - Figure captions
  - Footnotes
  - Highlighted phrases
- Appendix
- Back matter
- Index

You can use these starter set tags with their associated APFs and document profile as examples of how APFs are implemented. The symbol and macro facilities that you use to create new sets of GML for the different kinds of documents you produce can also be used to modify the starter set. For each kind of document you produce, you can define the GML tags so that they are meaningful to the people who are creating and updating them. You can also define the associated APFs so that the desired formatting style is produced.

```
:h2 id='store'.Storing Your Documents
:p.The Library program
stores data in groups of sequential records,
called documents, on the one or more VSAM Entry-Sequenced Data Sets
(ESDSs) that make up a Document Library.
Data can be stored in a Document Library as long as it is in a
sequential format with a fixed or variable record length that
is acceptable to the IBM Sequential Access Method (SAM).
The data can be:
:ul
:li.Prepared on an interactive system,
such as TSO, VM/CMS, VSE/ICCF, or VSPC
:li.Prepared by a text processor, such as ATMS-III
:li.Prepared on a
word-processing system, such as the IBM 3730 Distributed
Office Communications System or the IBM Office System 6.
:li.Input to or output from an application program
that is using the Library program as a subroutine
:eul
:p.A Document Library includes a directory that is used
to keep track of authorized users and the location and
attributes of the documents stored there.
This directory resides in a VSAM Key-Sequenced Data Set.
:p.The Library program uses the information stored in the
directory to ensure that all users only access documents they are
authorized to access, and only issue commands they are authorized
to issue, and to obtain accounting information for the installation.
```

Figure 6. Text Marked Up with GML

The GML APFs in the starter set support formatting for all of the devices supported by SCRIPT/VS. (For more information on the devices supported by SCRIPT/VS, see "Operating Environment" on page 27.)

Different sets of GML tags can be defined to support applications other than text processing. Depending on what the other applications are, it is possible that a single set of GML tags can support both text processing and the other applications.

This manual was marked up using the GML starter set and a few additional tags that were required to meet IBM's publishing specifications. Figure 6 shows the source text, including the GML tags, for the section "Storing Your Documents" on page 20.

The Document Library Facility enables you to store, archive (store on a sequential data set outside of the Document Library Facility), and retrieve documents within a secure environment. It can be used to store many different types of data but is primarily intended for text documents.

The Document Library Facility also includes ways to:

- Use the Document Library Facility as a subroutine of an application program.
- Call attribute processors (routines that change the characteristics of a document) to handle such things as converting one type of text processing controls to another, or, data manipulation.
- Obtain copies of some of the user and document profile information stored in the Document Library directory. (The Document Library directory is a VSAM Key-Sequenced Data Set that contains profile records for all Document Library Facility users and for all documents that are stored in the Document Library. A Document Library is the VSAM Entry-Sequenced Data Sets in which the Document Library Facility stores documents.)

With the Document Composition Facility installed, a user can:

- Use the ATMS conversion routine, that IBM provides with the Document Composition Facility, to process documents stored in the Document Library. (This routine is described in more detail in "The Document Composition Facility" on page 7.)
- Perform SCRIPT/VS processing in a background environment.
- Access SCRIPT/VS as a subroutine of an application program.

THE DOCUMENT LIBRARY FACILITY RELEASE 2 ENHANCEMENTS

Release 2 of the Document Library Facility, hereafter referred to as the Library program, includes functions that:

- Make it possible for a Document Library to consist of multiple VSAM Entry-Sequenced Data Sets and still be controlled by a single VSAM Key-Sequenced Data Set.
- Enable an ATMS-III user who is also a Library program user to use a Document Library for ATMS-III archive storage.
- Enable Library program users to maintain multiple versions of the same document in the same Document Library.
- Enable Library program users to maintain multiple documents with the same name but different content attributes in the same user library.
- Enable Library program users to archive a document in a sequential data set that is not part of a Document Library and put it back into a Document Library when required. (These documents can be archived based on their content attributes, version numbers, date of storage in the library, or user number.)
- Enable Library Program users to map their Library program user numbers to their ATMS-III user numbers. (This enables them to use a Document Library for ATMS-III archive storage.)
- Allow control statements and input data to be mixed in the same Library program input stream.
- Enable a document to be processed by an attribute processor for data conversion or manipulation on EXPORT.
- Enable a Library program user to store up to 50 characters of information with a document when it is stored in the Document Library. (This data can be listed when required.)
- Enable a Library program user to lock or unlock a document. (Locking a document makes it inaccessible even to its owner.)
- Enable a library program user to change a document's content attributes or name.
- Enable the Resource Access Control Facility (RACF) to be used to provide additional security for the documents stored in a Document Library.

These enhancements also enable a library administrator to:

- Copy a complete Document Library from one set of VSAM data sets to another.

- List all of the user mapping records that have been defined in the Document Library directory.
- Delete a user number and all documents associated exclusively with that number. (This invalidates the mapping.)
- Delete an attribute processor, class, or cluster profile record from the Document Library directory.
- Delete a mapping that has been set up between a Library program user number and an ATMS-III user number from the Document Library directory.
- Write exit routines to control user activities within a Document Library and to obtain additional accounting information.

STORING YOUR DOCUMENTS

The Library program stores data in groups of sequential records, called documents, on the one or more VSAM Entry-Sequenced Data Sets (ESDSs) that make up a Document Library. Data can be stored in a Document Library as long as it is in a sequential format with a fixed or variable record length that is acceptable to the IBM Sequential Access Method (SAM). The data can be:

- Prepared on an interactive system, such as TSO, VM/CMS, VSE/ICCF, or VSPC
- Prepared by a text processor, such as ATMS-III
- Prepared on a word-processing system, such as the IBM 3730 Distributed Office Communications System or the IBM Office System 6
- Input to or output from an application program that is using the Library program as a subroutine

A Document Library includes a directory that is used to keep track of authorized users and the location and attributes of the documents stored there. This directory resides in a VSAM Key-Sequenced Data Set.

The Library program uses the information stored in the directory to ensure that all users only access documents they are authorized to access, and only issue commands they are authorized to issue, and to obtain accounting information for the installation.

PROTECTING DOCUMENTS STORED IN A DOCUMENT LIBRARY

The Library program provides several means for protecting documents stored in a Document Library. Since a Document Library resides in VSAM data sets, VSAM passwords can be established for these VSAM data sets. (For more information on how VSAM passwords are established, see the Access Method Services publication applicable to your installation.)

Documents stored in a Document Library can also be protected by using these security features:

- User password
- Document password
- The three types of libraries that exist in a Document Library

If an installation uses OS/VS2 MVS and has the Resource Access Control Facility (RACF), program number 5740-XXH, Program Product installed, its library users can use RACF's security features to provide additional security for their documents.

PASSWORD PROTECTION

The Library program enables library administrators to assign user passwords. (This password is assigned when a library administrator creates a user's profile.) This prevents an unauthorized user from using authorized users' user numbers and thereby accessing their documents. Once a user password has been assigned, the user or the library administrator can change it as often as is necessary to ensure the user ID's security.

For additional security, the Library program also provides facilities for assigning document passwords. A user can assign a password to a document containing sensitive information when it is initially created. A user then specifies this password, along with the document name, when accessing the document. This password can be changed as often as necessary.

THE THREE TYPES OF LIBRARIES

A Document Library contains three types of libraries:

- Private libraries
- Project libraries

- Public libraries

Each of these types of libraries limits who may access documents stored in them and who may add documents to them and replace or delete documents already stored there. Since a user is only allowed to own one library (control what documents are added, replaced, or deleted), the proper library should be selected based upon who is going to require access to the documents that are going to be stored there and the sensitivity of the information they contain.

Private Libraries: A private library is accessible only to its owner and to a library administrator. As the owner of a private library, a user is the only one who can add, replace, or delete documents that are stored there. The library owner is also the only one, with the exception of a library administrator, who can supply other users with copies of these documents.

If a user's private library contains documents that other users also require, the library owner can designate the required documents as sharable. This enables the other users, who have been authorized by the document owner to do so, to access these documents on a read-only basis.

Project Libraries: A project library can be shared among several users. The Library program allows users to access documents in a project library if their user profiles indicate that they are authorized to use that project library. If the library's profile indicates that it is controlled, only its owner can add documents to it and replace or delete documents that exist there. Therefore, an authorized user of a controlled project library, other than the library owner, can only read the documents stored there. However, if a project library's profile indicates that it is not controlled, any authorized user can add documents to it. In this case, only the person adding the document or a library administrator can replace or delete it.

Public libraries: A public library can be accessed by any user of the Library program. Like a project library, if the public library is controlled, only the library owner can add, replace, or delete documents that are stored there. If it is not controlled, anyone can add documents to the library. However, whoever adds a document to the library is the only one, besides the library administrator, who can then replace or delete it.

A public library should be used to store information that is of general use to all library users, such as SCRIPT/VS document profiles and macro libraries. An installation can designate a library administrator as the library owner for a public library containing this type of information, thereby preventing anyone else from replacing or deleting it.

ADMINISTERING THE DOCUMENT LIBRARY

After the Library program has been installed, the DEFINE USER command must be used to identify at least one library administrator to the Library program. Until a library administrator is defined, no one else can use the Library program. Library administrators are the only ones with the authority to add users to the Library program.

Library administrators are responsible for maintaining a Document Library and its directory. They are the only users of the Library program who should have the authority to:

- Read and copy any document stored in a Document Library.
- Change the password or share status of any document stored in a Document Library. (A document's share status indicates who has the authority to read or replace it.)
- Use the ACCOUNT command to produce accounting statistics based on information stored in the Document Library directory.
- Add new users to the Library program and set them up with user profiles and passwords.

It is also their responsibility to:

- Alter user and document profiles when required
- Ensure that backup copies of the documents in the Document Library are maintained
- Change the cluster, attribute processor, and class profiles in the Document Library and delete them when they are no longer needed
- Set up a mapping between a library program user number and that user's ATMS-III user number so that a Document Library can be used as ATMS-III archive storage

"Appendix C: Summary of the Library Administrator Commands" on page 41 summarizes the Library program commands that a library administrator uses to perform these functions. These commands can only be used by someone with library administrator authority.

USING THE LIBRARY PROGRAM

Anyone at your installation that has been authorized as a user of the Library program can store and access data using the Document Library. However, to use the Library program effectively requires a working knowledge of the library program commands and the job control language for the system on which the Library program is running.

USING THE COMMANDS

Once a library administrator has defined someone as an authorized user, that person is able to use the Library program commands that are not restricted in use to a library administrator. These commands enable an authorized user to:

- Read documents from sequential files and store them in a Document Library.
- Retrieve documents from a Document Library.
- Change the user password or the name, or the password, content attributes, or share status of any owned document.
- Make a document inaccessible to anyone other than the library administrator.
- Send copies of documents to other users.
- Make backup copies of documents.
- Store multiple versions of a document.
- Archive infrequently used documents in a sequential data set that is not part of a Document Library. (If the archived document is required, it can be retrieved and put back into the Document Library. It can then be accessed from there just like any other document.)
- Erase one or more documents.
- Store and access data when the Library program is being used as a subroutine of an application program.
- Invoke SCRIPT/VS to format documents from a Document Library or from sequential files. (This can only be done if the Document Composition Facility is installed.)

"Appendix D: Summary of the Library Program User Commands" on page 43 summarizes the commands a Library program user uses to perform these functions.

USING THE LIBRARY PROGRAM CALLING SEQUENCES

The Library program provides several calling sequences (a sequence of programming instructions required to invoke a specific facility) that enable you to:

- Invoke SCRIPT/VS to process documents in a background environment when the Document Composition Facility is installed.
- Provide data to attribute processors for data conversion or manipulation. These attribute processors can convert or manipulate data:
  - When a document is being stored in a Document Library
  - When a copy of a document that is in the Document Library is being EXPORTed to an authorized user or to an application program that is using the Library program as a subroutine
  - Before a document is processed by the SCRIPT/VS formatter

  IBM provides a conversion routine with the Document Composition Facility that converts ATMS-II and ATMS-III control words to equivalent or similar SCRIPT/VS control words. Other attribute processors can be provided by your installation.
- Use the Library program as a subroutine of a application program. When you use the Library program as a subroutine of an application program, all of the functions of the Library program are available to you. (If the Document Composition Facility is installed, this calling sequence enables you to use SCRIPT/VS as a subroutine of the application program, since SCRIPT/VS can be invoked while using the Document Library Facility.)
- Write user exit routines to obtain copies of information contained in the Document Library directory. You can use this information to:
  - Control the accessibility, storage, archiving, retrieval, and deletion of documents

- Place additional information in a User Mapping Record (UMR) or User Profile Record (UPR)

- Control the addition, changing, and deletion of user profiles

These calling sequences are described further in the Document Library Facility Guide.

SCRIPT/370: COMPATIBILITY WITH SCRIPT/VS

Documents marked up for Installed User Program SCRIPT/370, Version 3, program number 5796-PHL, can be processed by the SCRIPT/VS formatter with little or no change to source files. When variations do occur, equivalent functions can usually be achieved through the use of user-written SCRIPT/VS macro processing functions. EASYSCRIPT "tags" will be processed, but the long form of SCRIPT/370 control words are not supported by SCRIPT/VS.

ATMS-II AND ATMS-III: COMPATIBILITY WITH THE DOCUMENT COMPOSITION FACILITY

The Document Composition Facility provides a routine that enables ATMS-II and ATMS-III documents to be processed by SCRIPT/VS. This routine converts most ATMS-II and ATMS-III implicit and explicit markup to similar or equivalent SCRIPT/VS control words, macros, and GML.

The interface to this routine is included in the Document Library Facility.

DCF: COMPATIBILITY BETWEEN RELEASES 1 AND 2

With Release 2 of the Document Composition Facility, the FILE option is mutually exclusive with the PRINT and TERM options on the SCRIPT command. The DEVICE option should be used with the FILE option instead of the PRINT or TERM options. Also, the .UD control word will now allow only blanks to be exempted from automatic underscoring.

DLF: COMPATIBILITY BETWEEN RELEASES 1 AND 2

The Document Library Facility library data set structure has been changed for Release 2. Therefore, documents that are currently stored in Release 1 library data sets must be moved to the new library data sets.

The syntax for a number of the Library program commands have been changed with Release 2. Also, the SEQUENCE option is no longer supported on the IMPORT, EXPORT, and SCRIPT commands, and the DATED option on the COPY command has been replaced with the ARCHIVE function.

PROGRAMMING REQUIREMENTS

The Document Composition Facility with the optional Foreground Environment Feature can be installed in the following interactive environments:

- OS/VS2 TSO (Time Sharing Option) with OS/VS2 (MVS), Release 3.8
- VM/370 CMS (Conversational Monitor System), Release 5.0
- ATMS-III (Advanced Text Management System-III), Release 1.0, on CICS/VS Version 1, Release 4.0.

The Document Library Facility can be installed in the background environments of the following virtual storage operating systems, provided VSAM and the Access Method Services are available:

- OS/VS1, Release 7
- OS/VS2 MVS, Release 3.8
- DOS/VSE

For DOS/VSE, VSE/Advanced Functions Release 2 (5747-CC1), and VSE/VSAM (5746-AM2) are required.

The Document Composition Facility can also be installed with the Document Library Facility in these background environments.

Note: These licensed programs are designed to work with the specified release levels and any subsequent releases and modifications unless otherwise stated.

MACHINE REQUIREMENTS

Processors: The Document Composition Facility will execute in an interactive environment under the control of the specified releases of CMS, TSO, or ATMS-III and the processors on which they are designed to operate.

The Document Library Facility and the Document Composition Facility operate in a background environment on all IBM System/370 Models 125 and above, the IBM 4331 and 4341 Processors, and the IBM 3031 and above Processors that are supported by DOS/VSE, OS/VS1, or OS/VS2 MVS. (The Document Library Facility has to be installed in order to execute the Document Composition Facility in a background environment.)

Terminals: SCRIPT/VS can format output in the specified interactive environments for the IBM 2741 Communication Terminal and the IBM 3270 Communication Terminal, or any terminal that is functionally equivalent to these terminals at the data stream level that can be used with CMS, TSO or ATMS-III.

In a background environment, SCRIPT/VS cannot be used with terminals directly, but terminals attached to the operating system can use SCRIPT/VS via remote job entry (RJE).

In addition, in a background environment, using the job submission capability of the Distributed Office Support System/370 (program number 5740-XY9), a document can be entered on the 3730 Distributed Office Communications System and transmitted to the host with the job. A user program can then place the 3730 document into a data set for subsequent host-application processing. Using attribute processors, the document can be placed in a Document Library and then processed by the Document Composition Facility.

Printers: SCRIPT/VS output can be formatted for printing on the IBM 3800 Printing Subsystem. For DOS/VSE, use of the 3800 Printing Subsystem requires the DOS/VSE IBM 3800 Printing Subsystem Independent Release (5747-CC1).

Output can also be formatted for printing on the IBM 1403, 3202, or 3211 line printers or any printer functionally equivalent to these IBM printers at the data stream level.

Storage Devices: The Document Composition Facility uses storage devices that are supported by the environment in which it is operating.

The Document Library Facility resides on a direct-access device and uses VSAM to access data in its libraries and SAM to read or write to external data sets. Therefore, any direct-access devices that are supported by these access methods will be supported by the Document Library Facility.

STORAGE ESTIMATES

Real Storage: The factors determining how much real storage the Document Composition Facility or the Document Library Facility requires are numerous and dependent on the installation. Therefore, no precise statement of real storage requirements can be made. The person responsible for installing these programs will determine the

amount of real storage to be used. Guidelines for establishing these real storage requirements will be included in the Program Directory that is shipped with each of these products.

Virtual Storage: The Document Composition Facility requires approximately 200K bytes of virtual storage for the load modules. Since the code is reentrant, all users can share the same copy; additional users do not increase storage space requirements. The Document Composition Facility also requires working storage to format documents. The size of working storage required depends on the complexity of the documents being formatted. However, at least 64K bytes of virtual storage should be available.

The Document Library Facility requires approximately 150K bytes of virtual or working storage. If SCRIPT/VS is going to be used with the Library program, an additional 15K bytes of virtual storage should be allocated to the Library program.

Note: These storage estimates are in addition to what is required for the system control program with which these products are running.

Direct-Access Storage: The Document Composition Facility requires the following direct-access storage:

	CMS (Blocks)	Other* (Tracks)
Load Module Size	250	30
GML Macro Library	160	19
GML Profile	25	2

*These estimates are based on an IBM 3330-1

The Document Library Facility load module requires 18 tracks of IBM 3330 direct-access storage. Additional storage is required for any data conversion routines, the exact amount being determined by the size of the routine. It is up to each installation to determine how much additional direct-access storage is required for document storage.

REQUIRED PUBLICATIONS

The following publications document the Document Composition Facility and the Document Library Facility, and will be available with the products. Information on how to install these products will be in the Program Directories shipped with the products.

- Document Composition Facility and Document Library Facility Executive Overview and Product Summary, GX20-2332. This card summarizes the functions of both of these products and illustrates their potential operating environments.

- Document Composition Facility: Generalized Markup Language (GML) User's Guide, SH20-9160. This manual is for all users of the Document Composition Facility. It describes the Generalized Markup Language (GML) that is provided with this Program Product, how to use GML to mark up documents, and how to use the Document Composition Facility to process them. The "Markup Guide" portion of this manual is also distributed in machine-readable form so that it can be customized to reflect any GML tags that your installation develops. This material can be used as an installation verification test.

- Document Composition Facility: Generalized Markup Language (GML) Quick Reference, SX26-3719. This reference card summarizes the GML in the starter set.

- Document Composition Facility: User's Guide, SH20-9161. This manual describes all SCRIPT/VS control words in detail and explains SCRIPT/VS symbol and macro facilities, conditional processing, GML development, and how to use SCRIPT/VS. It also discusses differences between SCRIPT/VS Release 1 and Release 2, and between SCRIPT/VS and SCRIPT/370.

- Document Composition Facility: User's Quick Reference, SX26-3723. This reference card summarizes the SCRIPT/VS control words and how to use SCRIPT/VS.

- Document Library Facility Guide, SH20-9165. This manual describes what a Document Library is, how to set one up, and the functions that can be performed by using the Document Library Facility commands.

- Document Composition Facility (DCF) Diagnosis, LY20-8067. This manual provides customers and IBM program support representatives with information for them to use in diagnosing program problems.

- Document Library Facility (DLF) Diagnosis, LY20-8068. This manual provides customers and IBM program support representatives with information for them to use in diagnosing program problems.

The messages for the Document Composition Facility and the Document Library Facility are distributed in machine-readable form and are also available as part of the informal documentation that is shipped with the product. The documentation for these messages is also available on microfiche. The order numbers of these two sets of microfiche are LYB0-8070 (for Document Composition Facility messages) and LYB0-8071 (for Document Library Facility messages).

RELATED PUBLICATIONS

One of the following publications will be required during installation of the Document Library Facility:

- OS/VS2 Access Method Services, GC26-3841

- OS/VS1 Access Method Services, GC26-3840

- VSE/VSAM Access Method Services: User's Guide and Reference, SC24-5144

These manuals describe the commands used during the creation and reorganization of the storage used by the Document Library Facility.

You should use the following publications to evaluate the use of the Document Composition Facility in different operating environments:

- IBM Virtual Machine Facility/370: Introduction, GC20-1800. This manual contains an introduction to CMS (Conversational Monitor System), which is one of the interactive systems with which SCRIPT/VS operates.

- IBM Virtual Machine Facility/370: CMS User's Guide, GC20-1819. This manual gives detailed user information about the CMS file system and related facilities for text entry, text editing, and file man-

agement.

- OS/VS2 TSO Terminal User's Guide, GC28-0645. This manual gives detailed user information about OS/VS2 TSO (Time Sharing Option), which is one of the interactive systems with which SCRIPT/VS operates. It describes the TSO EDIT facility and related facilities for text entry, text editing, and data set management.

- Customer Information Control System/Virtual Storage (CICS/VS) General Information, GC33-0066. This manual provides an overview of the CICS/VS operating environment. ATMS-III under CICS/VS is one of the interactive systems with which SCRIPT/VS operates.

- Advanced Text Management System-III (ATMS-III) General Information Manual, GH20-2404. This manual provides an overview of the ATMS-III Program Product and the functions it can perform.

- Introducing the IBM 3800 Printing Subsystem and Its Programming, GC26-3829. This manual provides general information about the IBM 3800 Printing Subsystem. It describes what the 3800 is and provides information about the standard and optional features available.

- OS/VS2 MVS Resource Access Control Facility (RACF) General Information, GC28-0722. This manual provides general information about the security features of RACF.

- VSE/Interactive Computing and Control Facility (VSE/ICCF) GeneralInformation, GC33-6066. This manual provides general information about the editing facilities of VSE/ICCF.

- Display Editing System for CMS User's Guide, SH20-1965. This manual provides information on how to use the editing facilities of the Display Editing System.

- Source Program Maintenance Online-II Description/Operations, SB21-1700. This manual provides information on how to use the editing facilities of SPM-II.

- IMS/VS Job Development System-II Description/Operations, SB21-1979. This manual provides information on how to use the editing facilities of JDS-II.

- TSO-3270 Structured Programming Facility-II General Information, GH20-1638. This manual provides general information about the editing facilities of SPF-II.

The following figure summarizes the options of the SCRIPT command and describes the effect they have on SCRIPT/VS processing. These options are described in more detail in Document Composition Facility: User's Guide.

Option	For Release 2	Description
BIND		Shifts the page image to the right.
CHARS		Specifies the fonts to be used.
CONTINUE		Continues processing after a nonsevere error occurs.
CTF	New	Specifies that output is to be prepared in STAIRS/VS Condensed Text Format.
DEST		Specifies a remote output station (TSO only).
DEVICE		Specifies a logical output device.
DUMP		Enables the .ZZ "Diagnostic" control word.
FILE		Specifies a disk file for output.
INDEX	New	Enables the indexing control words to be used to create an index.
LIB	Extended	Specifies symbol and macro libraries (CMS, TSO, and ATMS only).
MESSAGE		Controls message printing.
NOPROF		Suppresses the document profile.
NOSPIE		Prevents entering the SPIE exit routines (TSO and CMS only).
NOWAIT		Prevents prompting for paper adjustment on typewriter terminals.
NUMBER		Prints file name and line number of the last output line.
OPTIONS	Extended	Specifies a file that contains SCRIPT options (CMS and ATMS only).
PAGE		Selects the pages that are to be printed.
PRINT		Produces printer output.
PROFILE		Specifies the document profile.
QUIET		Suppresses the formatter's identifier message.
SEARCH		Specifies a library or partitioned data set that is to be searched for imbedded files (not valid in CMS).
SPELLCHK		Enables the .SV [Spelling Verification] control word.
STOP		Prints separate pages at the typewriter terminal.
SYSVAR		Sets symbol values for &SYSVARn.

Figure 7. Summary of the SCRIPT Command Options (Part 1 of 2)

Option	For Release 2	Description
TERM		Displays the output at a user's terminal (CMS or TSO only).
TLIB		Specifies spelling checking and hyphenation dictionaries.
TWOPASS		Uses two formatting passes to produce the output.
UNFORMAT		Prints all input lines without formatting.
UPCASE		Folds lowercase letters to uppercase for printing.

Figure 7. Summary of the SCRIPT Command Options (Part 2 of 2)

The SCRIPT/VS control words and the functions they perform are summarized in the following figure. These control words are described in more detail in Document Composition Facility: User's Guide.

Control Word	For Release 2	Description
... [Set Label]		Inserts a line that can be used as the target of the .GO control word.
.AA [Associate APF]	New	Associates a GML tag with the macro instructions that are invoked to perform the tag's processing functions.
.AN [And]	New	Works with the .IF or .OR control word to conditionally process SCRIPT/VS input lines.
.AP [Append]	Extended	Allows an additional file to be appended to the file just processed.
.BC [Balance Columns]		Causes SCRIPT/VS to attempt to balance the columns when a page eject occurs or when the column definition is changed.
.BF [Begin Font]	Extended	Causes SCRIPT/VS to use a new font.
.BM [Bottom Margin]		Specifies the amount of space in the bottom margin area.
.BR [Break]		Prevents the concatenation of the following text line with preceding text.
.BX [Box]		Draws horizontal and vertical lines around subsequent output text.
.CB [Column Begin]		Causes an eject to the next column or page.
.CC [Conditional Column Begin]		Causes a column eject if less than a specified amount of space remains in the column.
.CD [Column Definition]		Specifies the number of columns on the page and the position of each one.
.CE [Center]		Centers the text lines between the left and right margins.
.CL [Column Width]		Specifies the width of each column.
.CM [Comment]		Identifies a comment line.
.CO [Concatenate Mode]		Causes output lines to be formed by concatenating input lines.
.CP [Conditional Page Eject]		Causes a page eject if less than a specified amount of space remains on the page.
.CS [Conditional Section]		Allows conditional inclusion of input in the formatted output.

Figure 8. SCRIPT/VS Control Word Summary (Part 1 of 7)

Control Word	For Release 2	Description
.CT [Continued Text]	New	Causes the given line to be treated as a continuation of the previous text line.
.CW [Control Word Separator]		Defines the control word separator character.
.DC [Define Character]	Extended	Specifies characters to be used for special functions.
.DD [Define Data File-id]	Extended	Specifies the id of a file to be used with the .IM, .AP, or .WF control words.
.DF [Define Font]	New	Specifies the name of a font which will be invoked by the .BF control word.
.DH [Define Head Level]		Specifies the format and characteristics of the section headings produced by the .Hn control words.
.DI [Delay Imbed]		Delays the processing of text input lines until the next page eject occurs.
.DL [Dictionary List]	New	Adds or deletes dictionaries from the list of dictionaries to be used for hyphenation and spelling verification.
.DM [Define Macro]		Defines a macro using text, SCRIPT/VS control words, and special symbols.
.DS [Double Space Mode]	Changed	Causes subsequent output lines to be doublespaced.
.DU [Dictionary Update]		Adds words to, or deletes words from, the addenda dictionary.
.EC [Execute Control]		Executes the input line as a control word even if there is a macro of the same name.
.EF [End of File]	Extended	Simulates an end of file condition.
.EL [Else]	New	Works with the .IF control word to conditionally process SCRIPT/VS input lines.
.EM [Execute Macro]		Executes the input line as a macro even if macro substitution is off.
.EZ [EasySCRIPT]		Enables or disables EasySCRIPT processing functions.
.FL [Float]	New	Designates a block of text to be formatted and placed at the top or bottom of a subsequent output page or column.
.FM [Footing Margin]		Specifies the amount of space between the last line of text in the page's body and the first bottom title line.
.FN [Footnote]		Saves formatted text and prints it at the bottom of the page in single-column format.

Figure 8. SCRIPT/VS Control Word Summary (Part 2 of 7)

Control Word	For Release 2	Description
.FO [Format Mode]		Controls concatenation and justification of input lines.
.FS [Footing Space]		Specifies the number of lines in the bottom margin area that can contain bottom title lines.
.GO [Goto]		Causes SCRIPT/VS to locate the input line identified with the specified label and resume processing with that input line.
.GS [GML Services]	New	Performs various services that may be required when writing GML APFs.
.HM [Heading Margin]		Specifies the amount of space between the top title lines and the first line of text (or running heading) in the body of the page.
.HS [Heading Space]		Specifies the number of lines in the top margin area that contain top title lines.
.HW [Hyphenate Word]		Specifies hyphenation points for a word that might need to be hyphenated during formatting.
.HY [Hyphenate]	Extended	Controls the SCRIPT/VS automatic hyphenation function.
.IE [Index Entry]	New	Formats single index entries.
.IF [If]	Extended	Processes an input line if a specified relationship exists.
.IL [Indent Line]		Indents the next output line the specified amount of horizontal space.
.IM [Imbed]	Extended	Processes the named file at this point.
.IN [Indent]	Extended	Specifies the amount of space subsequent output lines are to be indented from the current margin.
.IR [Indent Right]	Extended	Specifies the amount of space subsequent output lines to be indented from the current right margin.
.IT [Input Trace]		Provides a trace of processing for each SCRIPT/VS control word and macro, as well as symbol substitution.
.IX [Index]	New	Causes the automatically generated back of book index to be imbedded and printed.
.JU [Justify Mode]		Causes left and right justification of output lines as needed to make the end of each line even with the current right margin.
.KP [Keep]	Changed	Ensures that a group of output lines are kept together in the same column.

Figure 8. SCRIPT/VS Control Word Summary (Part 3 of 7)

Control Word	For Release 2	Description
.L8 [Leading Blank]		Processes an input line in which the first character is a blank.
.LI [Literal]		Ensures SCRIPT/VS treats the input lines as text lines.
.LL [Line Length]		Specifies the length of each subsequent output line.
.LS [Line Spacing]		Specifies the number of blank lines between each subsequent output line.
.LT [Leading Tab]		Processes input lines in which the first character is a tab.
.LY [Library]		Specfies whether a library is to be used to resolve symbol and macro definitions.
.MC [Multicolumn Mode]		Restores column definition saved by a previous .SC control word.
.ME [Macro Exit]	New	Causes SCRIPT/VS to terminate processing of a macro.
.MG [Message]		Produces a message similar in format to the SCRIPT/VS error messages.
.MS [Macro Substitution]		Causes SCRIPT/VS to recognize and process macros.
.NL [Null Line]		Processes input lines in which there are no characters.
.OC [Output Comment]	Extended	Specifies a line that is to be inserted into the output document as an output comment without being formatted.
.OF [Offset]		Causes a hanging indention (a paragraph in which the indention of the first line is unchanged and subsequent lines are indented to the offset value).
.OR [Or]	New	Works with the .IF control word to conditionally process SCRIPT/VS input lines.
.PA [Page Eject]		Causes a page eject and can set the page number of the next page.
.PF [Previous Font]		Causes the last stacked font to become the current font.
.PI [Put Index]	New	Saves the specified lines for use in building a back-of-book index.
.PL [Page Length]		Specifies the amount of space, including top and bottom margins, for each output page.
.PM [Page Margins]	New	Causes SCRIPT/VS to shift the formatted output of each page to the right.

Figure 8. SCRIPT/VS Control Word Summary (Part 4 of 7)

Control Word	For Release 2	Description
.PN [Page Numbering Mode]		Controls external and internal page numbering.
.PP [Paragraph Start]		Begins formatting the output line as the start of a paragraph after a skip.
.PS [Page Number Symbol]		Sets a page number symbol.
.PT [Put Table of Contents]		Places the input line into the file used to accumulate table of contents entries.
.QQ [Quick Quit]		Causes SCRIPT/VS processing to terminate immediately without completing the current page.
.QU [Quit]		Causes SCRIPT/VS processing to terminate after completing the current page.
.RC [Revision Code]	Extended	Specifies a revision code symbol that is to be printed to the left of output lines containing updated material.
.RD [Read Terminal]	Extended	Allows user to type in one or more text lines while a file is being formatted.
.RE [Restore Status]		Restores environment that has been previously saved with the .SA control word.
.RF [Running Footing]	Extended	Specifies input lines that are to be saved as a running footing and processed at the bottom of each appropriate page.
.RH [Running Heading]	Extended	Specifies input lines that are to be saved as a running heading and processed at the top of each appropriate page.
.RI [Right Adjust]		Formats the output lines as unconcatenated input lines that are aligned with the right-hand margin.
.RN [Reference Numbers]	New	Controls output line reference numbering.
.RT [Running Title]		Defines running title lines for the top and bottom of even, odd, or all output pages.
.RV [Read Variable]	Extended	Allows user to assign a value to a symbol name by entering it at the terminal in response to an interactive request made while SCRIPT/VS is processing the input file.
.SA [Save Status]		Saves the current values and parameters of the formatting environment.
.SC [Single Column Mode]		Causes SCRIPT/VS to save the current column definition and format subsequent input lines in a single column.

Figure 8. SCRIPT/VS Control Word Summary (Part 5 of 7)

Control Word	For Release 2	Description
.SE [Set Symbol]	Extended	Defines a symbol name and assigns a value to it.
.SF [Save Font]	Changed	Saves the current font-id.
.SK [Skip]		Specifies the amount of space to insert before the next text output line.
.SL [Set Line Space]		Specifies the vertical distance between base lines of output lines.
.SP [Space]	Changed	Specifies the number of blank output lines to insert before the next text output line.
.SS [Single Space Mode]		Causes subsequent output lines to be single-spaced.
.SU [Substitute Symbol]		Controls the substitution of symbols with their previously assigned values.
.SV [Spelling Verification]		Defines the start and functions of the SCRIPT/VS spelling verification routine.
.SX [Split Text]	Extended	Produces an output line of three parts - "left" is a string of text that is aligned with the current left margin; "right" is a string of text that is aligned with the current right margin; and "fill" are the characters that fill the remaining space between the other two strings.
.SY [System Command]		Causes SCRIPT/VS to pass the input line to the host system for processing.
.TB [Tab Setting]	Extended	Specifies the tab settings to be used when the input file is formatted.
.TC [Table of Contents]		Imbeds the table of contents file which consists of entries that have been automatically generated by the .Hn control words or inserted using the .PT control word.
.TE [Terminal Input]		Allows user to enter lines interactively from the terminal when the file is formatted.
.TH [Then]	New	Works with the .IF control word to conditionally process SCRIPT/VS input lines.
.TI [Translate Input]		Specifies character translations to be performed on input lines before SCRIPT/VS processing begins.
.TM [Top Margin]		Specifies the amount of space in the top margin area.
.TR [Translate Character]		Specifies character translations to be performed on output.

Figure 8. SCRIPT/VS Control Word Summary (Part 6 of 7)

Control Word	For Release 2	Description
.TS [Translate String]	New	Translates input character to a character string.
.TU [Translate Uppercase]	New	Allows you to specify the output representation of each character in the source text when uppercase has been requested by the .UC, .UP, or .BF control words.
.TY [Type on Terminal]		Types the input line on the user's terminal during formatting.
.UC [Underscore and Capitalize]		Underscores and capitalizes one or more subsequent input lines.
.UD [Underscore Definition]	Changed	Defines the characters to be underscored when the .UC and .US control words are used.
.UN [Undent]		Causes the next output line's indention to change. (It is moved to the left of the current left margin.)
.UP [Uppercase]		Prints one or more subsequent input lines in uppercase characters.
.US [Underscore]		Prints one or more subsequent input lines with underscored characters.
.UW [Unverified Word]	New	Indicates that a misspelled word has been found during spelling verification. (Is generated by SCRIPT/VS.)
.WF [Write To File]		Writes one or more input lines to the output file.
.WZ [Widow Zone]	New	Specifies that the formatter is to suppress single line widows.
.ZZ [Diagnostic]		Turns on or off the diagnostic trace function and selects the type of data to be traced.

Figure 8. SCRIPT/VS Control Word Summary (Part 7 of 7)

APPENDIX C: SUMMARY OF THE LIBRARY ADMINISTRATOR COMMANDS

The following figure summarizes the library program commands that the library administrator uses to perform the functions described in "The Document Library Facility". More extensive descriptions of these commands can be found in the Document Library Facility Guide.

Command	For Release 2	Function
ACCOUNT		Creates accounting records (SMF record format type 47) from information contained in the user profiles in the Document Library directory.
ALTER CLASS	New	Changes the cluster name associated with a class.
ALTER CLUSTER	New	Changes cluster profiles and formats the cluster if new space has been added to it.
ALTER MAP	New	Changes the mapping of an external identifier to a Library Program user number.
ALTER PROCESS	New	Changes the entry point name of an attribute processor.
ALTER SYSTEM	Extended	Formats any additional space acquired since the last DEFINE SYSTEM or ALTER SYSTEM command was issued.
ALTER USER	Extended	Alters the description of a user profile.
COPY LIBRARY	New	Copies a complete Document Library from one set of VSAM data sets to another.
COPY USER	New	Copies all of the documents from one user's library to another user's library.
DEFINE CLASS	New	Defines the class content attribute of a document.
DEFINE CLUSTER	New	Defines new VSAM source clusters.
DEFINE MAP	New	Maps a Library program user number to that user's ID for an external system.
DEFINE PROCESS	New	Defines attribute processors that are going to be used by the Library program.
DEFINE SYSTEM		Defines the VSAM data sets that are going to be used by the Library Program.
DEFINE USER	Extended	Defines a new user library to the Library program.
LIST MAP	New	Lists all of the user mapping records that are defined in the Document Library directory.
PURGE CLASS	New	Deletes a class content attribute definition from the Document Library directory.
PURGE CLUSTER	New	Deletes a cluster profile from the Document Library directory.

Figure 9. Summary of the Library Administrator Commands (Part 1 of 2)

Command	For Release 2	Function
PURGE MAP	New	Removes the mapping of an external identifier to a Library program user's number.
PURGE PROCESS	New	Deletes processor profile records (PPRs) from the Document Library directory.
PURGE USER	New	Deletes a user number and all of the documents that are associated only with that number from the Document Library.

Figure 9. Summary of the Library Administrator Commands (Part 2 of 2)

APPENDIX D: SUMMARY OF THE LIBRARY PROGRAM USER COMMANDS

The following figure summarizes the commands any authorized Library program user can issue to perform the Library program functions described in "The Document Library Facility". More information about these commands can be found in the Document Library Facility Guide.

Command	For Release 2	Function
ALTER DOCUMENT	New	Locks or unlocks a document or changes its content attributes or name. A library administrator can use this command to alter any document in the Document Library.
ARCHIVE	New	Moves a document, along with its directory entry record, from the Document Library to a different sequential data set while retaining a copy of the Document Entry Record (DER) in the Document Library directory.
AUTH		Identifies someone as an authorized user of the Library program.
CLOSE		Terminates the reading of a document or the use of the Library program when the Library program is being used as a subroutine of an application program.
COPY DOCUMENT	New	Copies a document and its versions from one user's library to another user's library.
COPY IN	New	Restores all or part of a user's library using the backup copy that was created using the COPY OUT command. A library administrator can use this command to restore all or part of the Document Library.
COPY OUT	New	Provides a backup copy of all or part of a user's library. A library administrator can use this command to provide a backup copy of all or part of the Document library.
ENVIRONMENT		Describes the characteristics of a DOS/VSE sequential file that is being used for input or output.
EXPORT	Extended	Provides an external copy of a document being stored in the Document Library, and indicates if an attribute processor is to process the document before the external copy is made available to the calling program.
IMPORT	Extended	Reads documents from sequential files and stores them in the Document Library.
LIST CLASS	New	Lists all of the Class Entry Records (CERs) that are defined in the Document Library directory.
LIST CLUSTER	New	Lists the Cluster Profile Records (CPRs) that have been defined in the Document Library directory.

Figure 10. Summary of the Library Program User Commands (Part 1 of 2)

Command	For Release 2	Function
LIST DOCUMENT	Extended	Lists the attributes of any documents that are stored in the user's private, project or public library. A library administrator can use this command to list the attributes of any document in the Document Library.
LIST PROCESS		Lists the Attribute Profile Records (APRs) for all of the attribute processors that are defined to the Library program.
LIST USER	Extended	Lists the contents of the user's User Profile Record (UPR). A library administrator can use this command to list the User Profile Records of all of the users of the Library program.
PASSWORD		Changes a user's password. A library administrator can use this command to change any user's password.
PROTECT	Extended	Changes the password or share status of a document or a version of a document. A library administrator can use this command to change the password or share status of any document stored in the Document Library.
PURGE DOCUMENT	Extended	Deletes a document from a user's library. A library administrator can use this command to delete any document from the Document Library.
READ	Extended	Obtains records from the Document Library when the Library program is being used as a subroutine of an application program.
RETRIEVE	Extended	Copies a document that has been archived on a sequential data set back into the Document Library.
SCRIPT	Extended	Invokes the SCRIPT/VS formatter when the Document Composition Facility is also installed.

Figure 10. Summary of the Library Program User Commands (Part 2 of 2)

A

B

C

D

E

F

G

H

U

V

W

Document Composition Facility and
Document Library Facility General Information
GH20-9158-2

Reader's Comment Form

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. This form may be used to communicate your views about this publication. They will be sent to the author's department for whatever review and action, if any, is deemed appropriate. Comments may be written in your own language; use of English is not required.

IBM may use or 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.

Note: *Copies of IBM publications are not stocked at the location to which this form is addressed. Please direct any requests for copies of publications, or for assistance in using your IBM system, to your IBM representative or to the IBM branch office serving your locality.*

Note: Staples can cause problems with automated mail sorting equipment. Please use pressure sensitive or other gummed tape to seal this form.

List TNLs here:

If you have applied any technical newsletters (TNLs) to this book, please list them here:

Last TNL _____

Previous TNL _____

Previous TNL _____

Fold on two lines, tape, and mail. No postage necessary if mailed in the U.S.A. (Elsewhere, any IBM representative will be happy to forward your comments.) Thank you for your cooperation.

Reader's Comment Form

Fold and tape | Please do not staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
P.O. Box 50020
Programming Publishing
San Jose, California 95150

Fold and tape | Please do not staple | Fold and tape

DCF and DLF General Information (File No. S370-34) Printed in U.S.A. GH20-9158-2

IBM®

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601

DCF and DLF General Information (File No. S370-34) Printed in U.S.A. GH20-9158-2

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, N.Y. 10604

IBM World Trade Americas/Far East Corporation
Town of Mount Pleasant, Route 9, North Tarrytown, N.Y., U.S.A. 10591

IBM World Trade Europe/Middle East/Africa Corporation
360 Hamilton Avenue, White Plains, N.Y., U.S.A. 10601