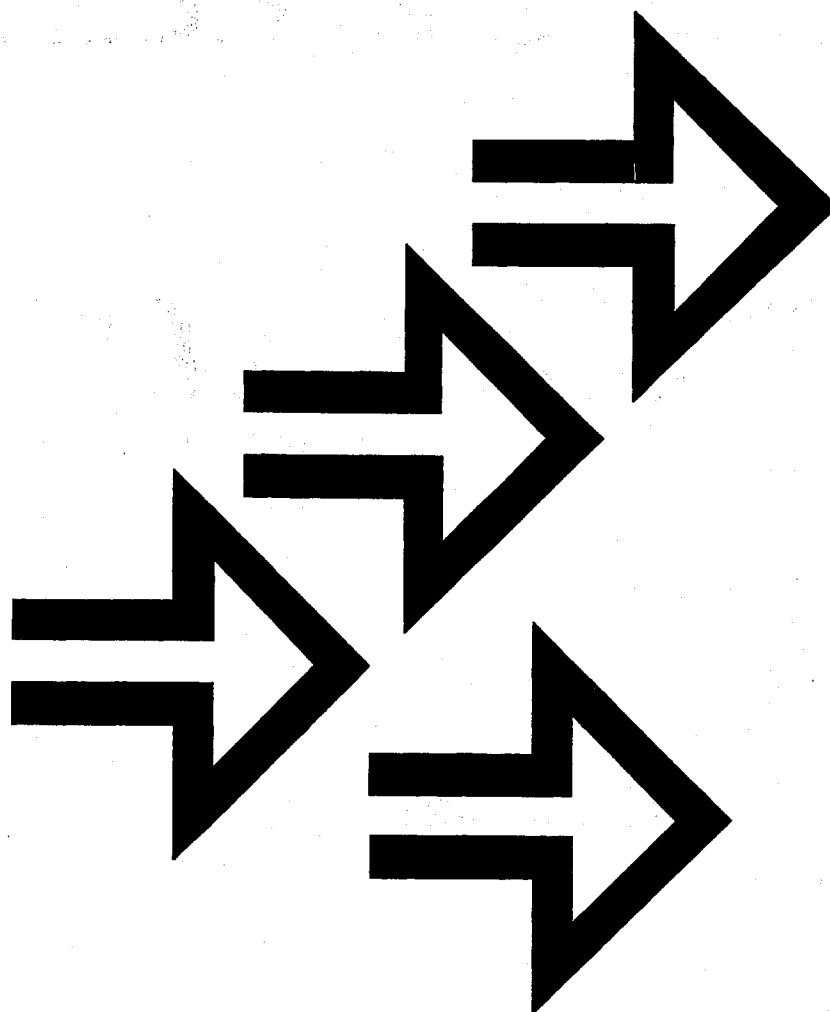


**Document Composition Facility**

**Diagnosis Guide and Reference**

**Release 4.0**





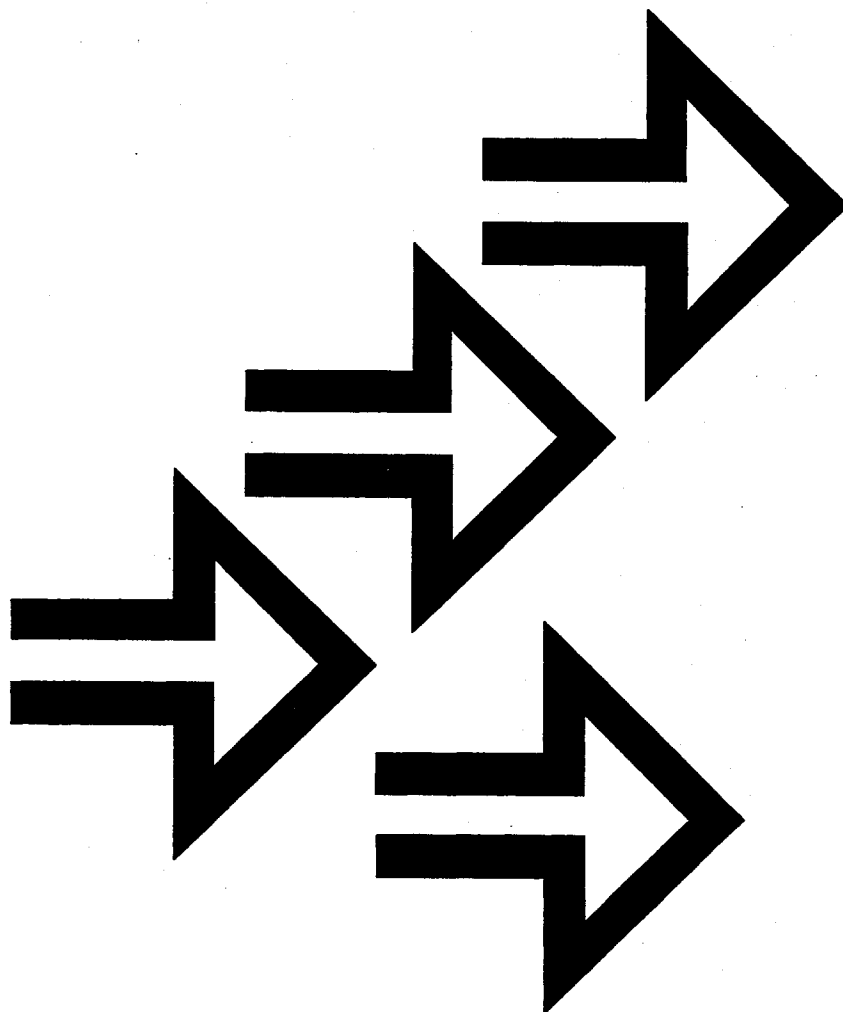


Document Composition Facility

LH40-0209-00

## **Diagnosis Guide and Reference**

Release 4.0



**Note!**

Before using this information and the product it supports, be sure to read the general information in "Notices" on page xi.

**First Edition (December 1990)**

This edition contains information from and makes obsolete the *Document Composition Facility: Diagnosis Guide and Reference*, LV32-0523. Changes or additions to the text and illustrations made to this manual are indicated by a vertical line to the left of the change.

This edition applies to Release 4.0 of the IBM Document Composition Facility Licensed Program, 5748-XX9, and to all subsequent releases and modifications until otherwise indicated in new editions or Technical Newsletters. Be sure to use the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for reader's comments appears at the back of this publication. If the form has been removed, address your comments to:

IBM Corporation  
Information Development  
Department 588  
P.O. Box 1900  
Boulder, Colorado 80301-9191

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 1984, 1990. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

# Contents

<b>Summary of Amendments</b> .....	ix
Release 4.0 .....	ix
<b>Notices</b> .....	xi
Programming Interface .....	xi
Trademarks .....	xi
<b>Chapter 1. Introduction</b> .....	1
Purpose .....	1
How This Manual Is Organized .....	1
Prerequisite Knowledge .....	2
<b>Chapter 2. Diagnosis Guide</b> .....	3
Diagnosis and Correction Process Personnel .....	3
Diagnostician .....	3
IBM Support Center Representative .....	3
IBM Specialist .....	4
IBM Change Team .....	4
Using Keywords .....	5
Developing the Keyword String .....	6
Getting Started .....	6
Eliminating Command Usage Errors .....	6
Gathering Diagnosis Information .....	7
Component Identification Keyword .....	7
Type-of-Failure Keyword .....	8
MSGx .....	8
ABEND .....	10
WAIT .....	10
LOOP .....	11
INCORROUT .....	11
DOC .....	12
PERFM .....	13
Module Keyword .....	14
Function Keyword .....	14
Subfunction Keyword .....	16
Release Level Keyword .....	17
Maintenance Level Keyword .....	18
Search Argument Procedure .....	19
Search Argument Technique .....	19
Preparing Authorized Program Analysis Reports (APARs) .....	20
Initiating An APAR .....	20
APAR-Supporting Documentation .....	20
Submitting Tapes or Cartridges as APAR Documentation .....	21
<b>Chapter 3. Program Overview</b> .....	23
General Description .....	23
Input Files .....	25
Types of Input .....	25
SCRIPT/VS Program Control Flow .....	27
Operating Environment .....	30

Programming Systems	30
Hardware Requirements	30
Storage Requirements and Estimates	32
Structural Overview	34
Program Structure	34
Program Conventions	37
Communication between Modules of the DCF program	38
<b>Chapter 4. Service Aids</b>	<b>39</b>
Text Input Service Aids	39
SCRIPT Command Options	39
The .IT Control Word	40
Terminal-Oriented Control Words	40
The .MG Control Word	40
Messages and Codes	41
Internal Program Service Aids	42
Internal Programming Errors	42
FE Patch Areas	42
The .ZZ Control Word	43
The .SY Control Word	44
User-defined exit for .SY	45
Module Identification	48
The Module Call Stack	48
Common Entry and Exit Logic	49
System Tools	49
List of Abbreviations	50
<b>Appendix. Related Publications and Products</b>	<b>51</b>
Publication Library Guide for the Document Composition Facility	52
DCF Publications	53
Related Products	53
Optional Features	54
Related Publications	55
<b>Index</b>	<b>57</b>

---

## Figures

1. DCF Message Identifier . . . . .	9
2. Function Names and Descriptions . . . . .	15
3. DCF Program Overview . . . . .	24
4. SCRIPT/VS Overview . . . . .	24
5. Font Initialization, Definition, and Usage . . . . .	28
6. SCRIPT/VS Sequence Flowchart . . . . .	29





## **Tables**

1. DCF Module Save Area .....	49
-------------------------------	----



---

## Summary of Amendments

### Release 4.0

Document Composition Facility (DCF), Release 4.0, includes the following:<sup>1</sup>

- **Shading:** Allows for shading of boxes, areas, and table cells on Advanced Function Printing (AFP) printers and PostScript\*\* devices.
- **MVS and VM Extended Architecture Exploitation:** Allows the DCF program to run and use storage above 16 megabytes.
- **Online Command Help:** Online help for the SCRIPT command is now available for the VM/CMS and MVS/TSO systems.
- **Online Help for Error Messages:** Online GML and SCRIPT error messages are now available for VM/CMS and MVS/TSO systems.
- **Separation Masters:** Allows construction of masters for multiple color printing and multi-part forms.
- **Including Overlays:** Allows you to merge page overlays in a DCF document that is formatted for an AFP printer.
- **Page Segment Enhancements:** DCF allows page segments containing Image Object Content Architecture (IOCA) Image Objects to be included when formatting for some AFP page printers.

The new ABSOLUTE parameter for the .SI [Segment Include] control word allows the user to specify the depth and width of the space to be reserved on the .SI control word instead of using the size of the segment.
- **File Size Limit:** SCRIPT/VS now processes VM input files containing more than 65,535 records.
- **PRINT Option:** Allows for use of the PRINT option on the SCRIPT command in VM. Files sent to AFP page printers<sup>2</sup> will go directly to SPOOL saving processing time and DASD.
- **IBM LaserPrinter 4028 Support:** Adds support for the IBM LaserPrinter 4028.
- **Font Character Set Size:** Raises the limit on the number of characters allowed in a font character set from 292 to approximately 1000.
- **Language Attribute on the :GDOC tag:** Allows for the selection of the language of the document, which includes translations for literals and messages, and using the correct language for spelling verification, hyphenation, and index sorting sequences in 15 different languages when using the starter set.
- **Hyphenation:** Adds six new languages for spelling verification and hyphenation, and algorithmic hyphenators for all 15 languages.
- **Revision Code Font:** Allows for selecting the fonts used in revision code characters.

---

<sup>1</sup> Release 4.0 of the Document Composition Facility may be used with Release 3 of the Document Library Facility.

<sup>2</sup> The PRINT option is still ignored for the IBM 4250 printer and PostScript devices.

- **Revision Character Alignment:** Allows for specifying revision codes on the left or the right or on alternate columns. For a two column format, the revision code character can appear to the left of the first column and to the right of the second column.
- **Table Storage Relief:** The new BREAK parameter for the .TA [Table] control word allows the user to break a table, which places the portion of the table that has been formatted, and releases the storage used for formatting that portion.
- **Page Number Limit:** Allows for the page numbering limit to increase to 99,999,999.
- **Output Comment:** Allows for use of the .OC [Output Comment] control word with AFP printers. The .OC control word is still ignored when formatting for the 4250 printer.
- **Schedule Tag Set:** Provides a set of tags to allow printing of schedules with start dates, end dates, projected dates, actual dates, slip dates, actions, responsible parties, and schedule changes.
- **Overhead Transparency Tag Set:** Provides a set of tags to allow printing of overhead transparencies with running headers, running footers, and a large, easy to read font.
- **Memo Tag Set:** Provides a set of tags to allow printing of memos in a consistent style.
- **TSO Changes to DCF:** Changes have been made to DCF to improve usability in the TSO environment. The changes can be summarized as follows:
  - Addition of TSO command options
  - Addition of TSO DDNAMES for SCRIPT command files
  - Changes to the Font Library Index Program (FLIP) to allow concatenated font libraries.
- **Module Renaming:** Allows a VM system programmer to change the names used for the DCF disk-resident module, any discontinuous shared segment, and the bootstrap module at installation time.
- **Font Library Index Program (FLIP) Enhancement:** In MVS/TSO, FLIP will search all of the data sets that have been designated as part of a concatenated font library. In VM/CMS, FLIP searches disk extensions for fonts of the specified filetype.

---

## Notices

References in this publication to products or services of IBM do not suggest or imply that IBM will make them available in all countries where IBM does business or that only products or services of IBM may be used. Noninfringing equivalents may be substituted, but the user must verify that such substitutes, unless expressly designated by IBM, work correctly. No license, expressed or implied, to patents or copyrights of IBM is granted by furnishing this document.

---

## Programming Interface

This manual is intended to provide to customers diagnostic information about the Document Composition Facility (DCF). This manual documents information which is Diagnosis, Modification, and Tuning Information provided by DCF.

**Warning:** Do not use this Diagnosis, Modification, and Tuning Information as a programming interface.

---

## Trademarks

The following terms are trademarks of IBM Corporation in the United States or other countries:

IBM®	Advanced Function Printing™
PSF™	Print Services Facility™
S/370™	IBM 3800 Printing Subsystem Models 3 and 6™
S/390™	IBM 3900 Advanced Function Printer™
AFP™	IBM 3825 Page Printer™
IBM 3827 Page Printer™	IBM 3828 Advanced Function MICR Printer™
IBM 3835 Page Printer™	IBM 3812 Page Printer™
IBM 3816 Page Printer™	IBM LaserPrinter 4028™
IBM 4224-2xx Page Printer™	IBM 4234 Page Printer™
IBM 3820 Page Printer™	IBM LaserPrinter 4019™

The following terms are trademarks of other companies as follows:

PostScript	Adobe Systems Incorporated
------------	----------------------------

Central Programming Service support and maintenance are available for the following Generalized Markup Language (GML) profiles and macros:

- The GML starter set profile: DSMPROF4
- The GML schedule profile: DSMSPROF
- The GML overhead transparency profile: DSMTPROF
- The GML memo profile: DSMMPROF
- The GML bar-code profile: DSMBPROF
- The GML SCRIPT Mathematical Formula Formatter profile: DSMFPROF
- The GML office document feature profile: DSMOPROF
- The GML office document feature macro-instruction library: DSMOGML
- The GML SCRIPT Mathematical Formula Formatter macro-instruction library: DSMFMAC
- The GML macro-instruction library: DSMGML4

Support and maintenance, however, *are not* available for these profiles and macros if they have been modified in any way. If you modify these items, it is recommended that you also maintain an ***unmodified*** version of these items for diagnostic purposes.

An authorized program analysis report (APAR) may not be submitted regarding the translation of literals, messages, and phrases in the starter set.

---

## Chapter 1. Introduction

This chapter provides general information about the purpose of this manual, how the information is organized, and what prerequisite knowledge you need to use this manual.

---

### Purpose

The *Document Composition Facility: Diagnosis Guide and Reference* provides information to help you isolate failures in the Document Composition Facility (DCF) and communicate those failures to an IBM Support Center representative. This manual allows you to understand the program organization and flow of control, but it does not provide you with enough detail to change or correct the program logic. The information in this manual is to be used only for diagnostic purposes.

This manual describes a systematic way of selecting keywords, so that you can describe a suspected program failure. By using the appropriate keywords, you can search the Early Warning System (EWS) microfiche or the Information Systems (Info-Sys) data base to discover if your problem has already been reported.

If you are an IBM customer, you can give the keyword sets to your IBM Support Center representative, who will access the Info-Sys data base for you.

**Note:** You can use this manual with or without access to Info-Sys or EWS. You can expedite the diagnosis of any problem by using keyword sets when communicating with your IBM Support Center representative.

---

### How This Manual Is Organized

This manual was designed to help you isolate a failure in the Document Composition Facility (DCF) and communicate that failure to an IBM Support Center representative.

The information in this manual is presented in four chapters and one appendix:

- Chapter 1, "Introduction" explains the purpose of the publication, how the information is organized, and what prerequisite knowledge is necessary to use this publication to your best advantage.
- Chapter 2, "Diagnosis Guide" on page 3 shows you how to build a keyword string that describes a program failure in DCF. You can use this string to find out if a correction exists for that failure.
- Chapter 3, "Program Overview" on page 23 describes the program structure of DCF, control flow, principal control and data areas, and relationships of the DCF functions.
- Chapter 4, "Service Aids" on page 39 describes text input and internal program service aids. It also contains a "List of Abbreviations" on page 50 that lists and spells out the names and abbreviations used in this manual.
- Appendix, "Related Publications and Products" on page 51 lists not only the books in the DCF Library and other publications related to this publication, but related products as well.

- An index has been added to the back of the book to be used as another means of retrieving information.

---

## Prerequisite Knowledge

Prerequisite knowledge for using this manual is an understanding of DCF capabilities as described in:

- *Document Composition Facility: SCRIPT/VS User's Guide*, S544-3191, which contains examples using SCRIPT/VS control words.
- *Document Composition Facility: SCRIPT/VS Text Programmers Guide*, SH35-0069, which contains information about how to use DCF.
- *Document Composition Facility: SCRIPT/VS Language Reference*, SH35-0070, which contains information about the functions of DCF.



---

## Chapter 2. Diagnosis Guide

The purpose and scope of this chapter is to help you:

- Build a string of keywords that describes a program failure in the Document Composition Facility (DCF)

**Note:** A keyword is a word or abbreviation that describes one aspect of a program failure.

- Use the string to find out if a correction exists for that failure
- Help your IBM Support Center representative in reporting a new problem if no correction exists.

---

### Diagnosis and Correction Process Personnel

The people involved in the diagnosis and correction process are listed below, along with a brief description of their tasks.

#### Diagnostician

The diagnostician identifies the IBM program (DCF in this case) causing the failure and describes the failure using a keyword string. The diagnostician then uses the string to search an online data base called Information Systems (Info-Sys) or a microfiche copy of the Info-Sys called the Early Warning System (EWS). The purpose of this search is to determine if the same problem has been described previously and, if so, whether a correction exists for the problem. The diagnostician can contact an IBM Support Center representative for help in doing the search.

#### IBM Support Center Representative

The Support Center representative has three functions in the diagnosis and correction process:

- To help the diagnostician search Info-Sys or EWS
- To provide a correction if one exists
- To refer the problem to an IBM specialist if the problem is new.

## IBM Specialist

The IBM specialist contacts the diagnostician. Together they:

- Verify that the diagnostician used the correct keywords in building the string
- Gather additional information about the failure. See "Preparing Authorized Program Analysis Reports (APARs)" on page 20 for the information that might be needed.

After confirming that the problem has not been reported before, the IBM specialist tries to develop a bypass to help the customer continue productive work. The customer is asked to send needed documentation to the IBM change team. The Support Center representative prepares an APAR on the new problem and sends it to the IBM change team.

## IBM Change Team

The tasks of the IBM change team are:

- To develop corrections for program problems reported on APARs
- To make the correction available to the customer reporting the problem
- If needed, to modify the keyword string to describe the failure more accurately
- To add the keyword string and the program correction to Info-Sys.

---

## Using Keywords

Each keyword describes one aspect of a program failure. The first keyword identifies the component name of DCF. You can search Info-Sys or EWS with just this keyword and detect all APARs written for DCF. Adding other keywords to the search argument reduces the number of matches and increases the chances of locating a match for your particular problem.

Remember that you will be most successful when you search Info-Sys or EWS if you follow these rules:

- Use the search keywords given in this chapter
- Spell keywords as they are spelled here each time you give them
- Include the appropriate keywords in any APAR.

If circumstances make it difficult to follow the instructions for a particular keyword, you can omit that keyword and go on to the next. In general, though, you will not want to start a search until you have a complete set of keywords.

A complete set of DCF keywords for your search argument consists of the following:

- Component name
- Type of failure
- Module name
- Function
- Subfunction
- Release level
- Maintenance level.

## Developing the Keyword String

Follow the instructions in the keyword procedures. If no specific step is indicated (a branch), follow the next sequential step. Information necessary to perform the numbered step is presented as bulleted information immediately following the step.

The keyword string is highlighted throughout the procedures. Occasionally, a step will provide a partial keyword string that describes what is known at that time about the program problem. A partial keyword string may require that a specific piece of information be inserted, such as replacing the x of the MSGx keyword with the identifier of the message received. The information to be substituted will be known at that time. Continue to develop the keyword string until you are instructed to use it as a search argument. A search of Info-Sys or EWS will be recommended as soon as the keyword string is complete enough to make a search reasonably productive, without requiring excessive research of the problem. If an early search is unsuccessful, instructions are given as to how to continue diagnosing the problem.

## Getting Started

Before developing a keyword string as described in this chapter, thoroughly check the application program for usage errors. If your problem persists, then begin developing your keyword string.

## Eliminating Command Usage Errors

Use the following procedure to verify that the problem is not caused by incorrect use of DCF commands, functions, GML tags, or control words.

1. Examine the parameters specified on each command, macro, GML tag or control word to verify that they have been specified correctly. See the *Document Composition Facility: SCRIPT/VS Language Reference*, the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*, and the *Document Composition Facility: SCRIPT/VS User's Guide* for correct command, control word, and macro usage information. See the *Document Composition Facility: Generalized Markup Language Starter Set Reference* for Generalized Markup Language (GML) usage information. See the *Bar Code User's Guide* for information on using the Bar Code tags. See the *Document Composition Facility: Generalized Markup Language Applications Guide* for information on using the Applications tags.
2. Check to ensure that the error is not a usage problem; if it isn't, rerun the job in an attempt to reproduce the problem before continuing.
3. If you have determined that your problem is not a usage error, then you will need to know how to gather diagnosis information and how to develop the keyword string. If all command, macro, control word, and GML parameters appear to be correctly specified, go to "Gathering Diagnosis Information" on page 7.

---

## Gathering Diagnosis Information

This procedure is designed to gather the diagnostic information that is needed to develop the keyword string to be used as a search argument for Info-Sys or EWS.

1. If DCF has been changed since your document last ran successfully, examine the changes. If the error is occurring as a result of the changes made in the code and cannot be corrected, note the change that caused the error.
2. Correct all problems diagnosed by messages and ensure that any messages previously generated have nothing to do with the problem being worked on.
3. Rerun SCRIPT/VS with the MESSAGE(ID) option of the SCRIPT command to yield maximum diagnostic information.
4. If the symptoms change, go back to step 2
5. Note the sequence of events that lead to the error condition. This information may be useful in developing a keyword string and will be needed if an APAR is required.
6. Reduce the size of the failing document as much as possible, to eliminate extraneous symptoms and to demonstrate the failure more clearly.
7. Go to the "Component Identification Keyword" to begin developing the keyword string.

---

## Component Identification Keyword

The component identification number is the first in a set of keywords describing any problem. It is derived from the order number of DCF and serves to identify the Info-Sys data base library that contains APARs for this product. Use this keyword whenever you suspect that DCF is the failing component. Used alone, it produces a full listing of APARs against the DCF component.

### Procedure

1. For DCF, the component identification number, which is also the format of the first partial keyword set, is:  
  
5748XX9
2. If the error resulted in a dump during the running of SCRIPT/VS, register 11 will point to the common data area. At offset X'2F8' from the beginning of this area you should find 5748-XX9, which identifies DCF.
3. Go to "Type-of-Failure Keyword" on page 8 to continue developing the keyword string.

**Note:** Refer to the *Document Composition Facility: Office Document Feature Reference* for diagnosis information for the Office Document Feature (ODF).

## Type-of-Failure Keyword

The type-of-failure keyword identifies the type of program failure that occurred and is a required part of the keyword string. The seven type-of-failure conditions are:

- MSGx
- ABEND
- WAIT
- LOOP
- INCORROUT
- DOC
- PERFM.

If you are not certain which of two keywords to use, use MSGx first. The situations that govern each type-of-failure keyword are explained below.

## MSGx

Use the MSGx keyword when a message:

- Is not documented or is documented incorrectly in the *Document Composition Facility Messages* manual
- Is generated when it should not have been generated
- Is not generated when it should have been generated
- Warns of an unrecoverable program error detected by DCF or a DCF-related program.

The GML starter set messages are sent by the .MG control word and are identified by three plus signs (+++). These plus signs are followed by an identifier in the form DSMGMLnnx, where *nnn* is a message number and *x* is a severity code.

DCF messages are identified by a set of 10 characters in the form DSMmmmmnnnx, as shown in Figure 1.

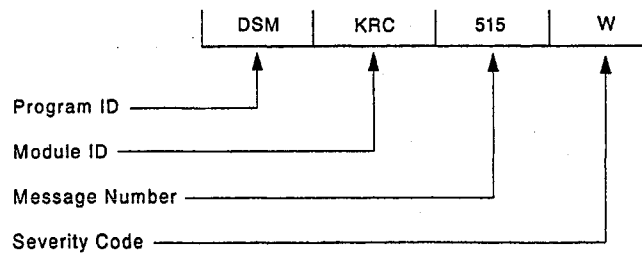


Figure 1. DCF Message Identifier

### Procedure

You need the message identifier to complete the MSGx keyword string. Use the MESSAGE (ID) on the SCRIPT command to generate the message identifier.

1. Replace the x in the MSGx keyword with the DCF message identifier.
2. Add the module name to the MSGx keyword.

The format of the partial MSGx keyword string is:

5748XX9 MSGx module-name

For example, the following message:

DSMKRC515W .RC MODE WAS ON OR OFF ALREADY

produces the following partial MSGx keyword string.

5748XX9 MSGDSMKRC515W DSMLPKRC

3. Go to "Function Keyword" on page 14 and follow the instructions.

## ABEND

Use the ABEND keyword when the host system, DCF, or any program that services DCF, stops without a message.

Do not use this keyword if the abend was forced by the host system or by the operator because of too much time spent in a wait state or an endless loop; for those situations, see the WAIT and LOOP keywords.

### Procedure

1. If the ABEND identification includes a code, add it to the keyword. For example, use ABENDOC4 if an OC4 abend occurred.

**Note:** If a program interrupt occurred in SCRIPT/VS and the NOSPIE option was specified in the SCRIPT command, SCRIPT/VS will abend. If NOSPIE was not specified and the default option, SPIE, was used, a message will be sent. In either case, rerunning with the other option may give you an alternative way to describe the failure.

2. If you have a complete storage dump, locate the address at which the ABEND occurred. An instruction address appears as the last six characters of the program status word (PSW) at the beginning of the dump. This address will be used later as the starting point in developing the module keyword string.

The format of the partial ABEND keyword string is:

5748XX9 ABENDxxxx

3. Go to "Module Keyword" on page 14 and select a module name for your keyword string.

## WAIT

Use the WAIT keyword when:

- The program status word (PSW) for the host system has the wait bit on
- The host system, DCF, or some program that services DCF has suspended activity, without issuing a message, while waiting for some condition to be satisfied.

Do not use this keyword if the wait occurs after an abend, as the result of an unanswered message, or because of an endless loop in DCF. (Use the ABEND or LOOP keyword.)

### Procedure

1. If you have a complete storage dump, locate the address at which the WAIT occurred so you can determine the module in which the WAIT occurred. All SCRIPT/VS modules contain a 8-character module ID near the start of the module. Immediately ahead of this 8-character string is a branch instruction that branches around the module ID and a 3-byte hexadecimal date. Use the full module name, including the DSM prefix, as part of the keyword string.



2. If possible, select from the following list a keyword to describe the resource for which the program is waiting.

- STORAGE
- DISK
- INPUT
- OUTPUT
- BUFFER

If the wait is for a resource that is not described by one of these keywords, you can choose some other descriptive word—but make it specific.

The format of the partial WAIT keyword string is:

5748XX9 WAIT module-name resource

3. Go to "Function Keyword" on page 14 and select a function name.

## LOOP

Use the LOOP keyword if some part of the program code is executed endlessly; include situations where some part of the output (for example, a blank page) is repeated endlessly.

Do not use this keyword for an endlessly repeated message (use the MSG keyword) or an intentional loop used to wait for some resource (use the WAIT keyword).

### Procedure

1. If the DCF program suspends activity for no apparent reason, it may be in either a loop or a wait state. A loop may be evident from a repetition of names in the module call stack (the two most-recent links of a chain of module save areas) or from some other symptom, such as a page of output being printed repeatedly. If a loop is not evident, use the WAIT keyword.
2. If you have a complete storage dump, locate the address where the LOOP occurred. All SCRIPT/VS modules contain a 8-character module ID near the start of the module. Immediately ahead of this 8-character string is a branch instruction that branches around the module ID and a 3-byte hexadecimal date. Use the full module name, including the DSM prefix, as part of the keyword string.

The format of the partial LOOP keyword string is:

5748XX9 LOOP module-name

3. Go to "Function Keyword" on page 14 and select a function name.

## INCORROUT

Use the INCORROUT keyword when the output is incorrect.

Do not use this keyword for text that is repeated endlessly (use the LOOP keyword).

## Procedure

1. Be especially careful at this point that the output really is incorrect, in terms of DCF commands and control words used, and not merely undesirable.
2. It may be necessary at this point to trace the expansion of GML tags, macros, symbols, and control words, to determine exactly what SCRIPT/VS was told to do.
3. Check Chapter 4, "Service Aids" on page 39 for suggestions on analyzing the effect of control words.

The format of the partial INCORROUT keyword string is:

5748XX9 INCORROUT

4. Go to "Module Keyword" on page 14 and select a module name for your keyword string.

## DOC

Use the DOC keyword when a programming problem appears to be caused by incorrect, missing, or ambiguously stated information in one of the DCF manuals (except for *Document Composition Facility Messages*—in that case, use the MSG keyword).

## Procedure

1. If the documentation problem is in any DCF publication, place the publication's order number after the DOC keyword.

**Note:** Do not use hyphens as part of the keyword. For example, if the order number of the book is S544-3191-01, the DOC keyword is:

DOC S5443191 1

2. The format of the partial DOC keyword string is:

5748XX9 DOC ordernumber

**Note:** You should pad the first digit of the suffix with a zero (0) so that the suffix is two digits long. Because the DOC string is so general, it may result in your getting too many matches to your problem. If it does, you may want to add the number of a page or figure, or the text (excluding special characters) of a section heading to your DOC order-number keyword string to more specifically limit the search.

In any case, try to add the most unique information that you can so that you narrow the search as much as possible.

3. To determine if this is a known problem, go to "Search Argument Procedure" on page 19 and follow the instructions. If the search is unsuccessful, return here to continue.
4. If this is a documentation error that you believe will cause lost time for other users, contact an IBM Support Center representative for help in initiating a documentation APAR.
5. If the documentation problem is not severe, use the Reader's Comment Form at the back of the manual to suggest improvements to the publication.

## PERFM

This keyword is used when the performance problem could not be corrected by system tuning.

### Procedure

1. Use the PERFM keyword to identify situations in which some part of DCF fails to meet explicitly stated expectations. Because most performance problems relate to system tuning, they should be handled by system engineers and system programmers.

The format of the partial PERFM keyword string is:

5748XX9 PERFM

2. Go to "Function Keyword" on page 14 and follow the instructions.

## Module Keyword

Use the **MODULE** keyword to identify the module you suspect contains the program failure.

### Procedure

1. Use the instruction address you found in the ABENDxxxx or the WAIT/LOOP procedure as a starting point. The leftmost column of numbers will indicate addresses within the dump.
2. Scan backward in the dump until you see a 5-character module ID followed by a module date. For example, LPKRC is a DCF module ID, and 092582 is the module date.
3. If there is a branch instruction (a 4-byte instruction with 47F0 as the first two bytes) just ahead of the module ID, you have found the beginning of a DCF module.
4. Use the 8-character module ID as the module keyword.

The format of the partial **MODULE** keyword string is:

```
574XX9  module-name
```

and the partial set of keywords now might be:

```
574XX9  MSGDSMKRC515W DSMLPKRC
```

5. Go to "Function Keyword" to continue with the keyword-selection process.

## Function Keyword

The procedure in this section describes how you can add another keyword to your search argument by determining the DCF function that was processing when the failure occurred. Selecting an accurate function keyword provides a tighter focus on the area of the program code that contains the error.

### Procedure

1. From the list in Figure 2 on page 15, select the name of the function that was processing when the program failed.
2. If the failure occurred during page layout, for example, the function keyword is **PAGE**.

The format of the partial **FUNCTION** keyword string is:

```
5748XX9  function
```

and the partial set of keywords now might be:

```
5748XX9  MSGDSMKRC515W DSMLPKRC PAGE
```

3. Go to "Subfunction Keyword" on page 16 and follow the instructions.

**Note:** The arrangement of failure symptoms into the categories shown in Figure 2 on page 15 is not meant to imply that the program code is organized in the same way. It is merely an attempt to provide standard descriptions for situations in which it may be difficult to assign a program failure to a single control word.

---

<b><u>Function Name</u></b>	<b><u>Function Description</u></b>
<b>COLUMN</b>	Column layout: arrange text in columns; balance columns.
<b>EASY</b>	EasySCRIPT.
<b>ENVIRONMENT</b>	Environment-oriented functions and input/output services; specific environments can be coded as TSO, CMS, CICS, and so on.
<b>FONT</b>	Provide the proper type font, including capitals, underscoring, and boldface.
<b>GML</b>	GML tag or APF processing.
<b>HEADING</b>	Chapter and section headings: arrange headings of different levels; assemble into a table of contents.
<b>HYPHENATION</b>	Hyphenation.
<b>INDEX</b>	Index processing support.
<b>INIT</b>	Initialization: initialize work areas.
<b>LINE</b>	Line layout: justify lines, center lines, tab, or indent.
<b>LOGIC</b>	Logic flow: maintain the proper sequence of events in processing the document; include and exclude appropriate pieces of text.
<b>MACRO</b>	Macro definition, expansion, or use.
<b>PAGE</b>	Page layout: arrange text on a page; provide proper margins at top, bottom, and sides; provide titles, running headings, and running footings; start page breaks.
<b>SPACE</b>	Provide space between lines.
<b>SPELL</b>	Spelling checking.
<b>STAIRS</b>	STAIRS processing support.
<b>SYMBOL</b>	Symbol substitution.

---

*Figure 2. Function Names and Descriptions*

## Subfunction Keyword

The control word and GML tag keywords specify two of the basic subfunctions of SCRIPT/VS.

### Procedure

1. If SCRIPT/VS has begun to process text, you may be able to associate the program failure with a specific control word or GML tag. If so, the control word or GML tag itself can be used as a keyword. (See the *Document Composition Facility: Text Programmer's Quick Reference* card or the *Document Composition Facility: SCRIPT/VS Language Reference* for a list of control words and the *Document Composition Facility: Generalized Markup Language Starter Set Reference* for a list of GML tags.

**Note:** Special characters impose an exception to this rule. Info-Sys will not accept the initial period that signals a control word, nor will it accept the initial colon that signals a GML tag. Depending upon the particular control word or GML tag, you can either use the two digit form or you can spell out the control word or GML tag completely. For example, with the .FN [Footnote] control word you can enter either

FN or FOOTNOTE

or both. With the :FIG tag you can enter either

FIG or FIGURE

or both. You should use whichever method results in the least number of matches in your search.

Do include the period and colon, though, when writing an APAR, to allow others to distinguish SCRIPT/VS control words and GML tags from two-character codes that may be used by other programs. For control words using special characters, do this:

- For Set Label ('...'), use LABEL.
- For Comment ('.\*'), use COMMENT.

The format of the partial SUBFUNCTION keyword string is:

5748XX9 subfunction

and the partial set of keywords now might be:

5748XX9 MSGDSMKRC515W DSMLPKRC PAGE RC

2. Go to "Release Level Keyword" on page 17 and follow the instructions.

---

## Release Level Keyword

This keyword identifies the release level of DCF currently installed on your system. The release level keyword is one of the following three-digit codes:

- 010 for release level 1
- 020 for release level 2
- 300 for release level 3
- 310 for release level 3.1
- 320 for release level 3.2
- 400 for release level 4.0.

The release level keyword is optional in the Info-Sys search argument, but it is required on APAR forms.

**Note:** Ensure that the release level you specify is supported.

### Procedure

1. Select the keyword that applies to the DCF release level currently installed on your system.

The format of the partial RELEASE LEVEL keyword string is:

5748XX9 release-level

and the partial set of keywords now might be:

5748XX9 MSGDSMKRC515W DSMLPKRC PAGE RC 2

2. Go to "Maintenance Level Keyword" on page 18 and follow the instructions.

## Maintenance Level Keyword

As a general rule, use the most current PTF number as a maintenance level keyword to search Info-Sys or EWS *only* if you feel that the description found in this PTF matches the problem you are having; if it doesn't, the match you are looking for may have been reported in some other PTF.

### Procedure

1. Determine the number of the latest PTF applied to DCF.
  - For OS/VS environments, process the System Modification Program (SMP) PTF list.
  - For VSE environments, process the Maintain System History Program (MSHP).
  - For CMS environments, inspect the VMSERV output listing.

2. You now have all the necessary information for an effective search of known problems in Info-Sys or EWS. Your complete set of keywords might look like this:

<b>Component Identification</b>	5748XX900
<b>Type-of-Failure</b>	MSGDSMKRC515W
<b>Module</b>	DSMLPKRC
<b>Function</b>	PAGE
<b>Subfunction</b>	RC
<b>Release Level</b>	400
<b>Maintenance Level</b>	UP11357

Having described each aspect of the program failure with these keywords, you are ready to search the Info-Sys or EWS data base with the following keyword string (search argument):

5748XX9 MSGDSMKRC515W DSMLPKRC PAGE RC 32 UP11357

3. If you do not have direct access to Info-Sys or EWS, contact your IBM Support Center and have them search Info-Sys or EWS with the search argument keyword set you have developed.
4. If you have access to Info-Sys or EWS, go to the section "Search Argument Procedure" on page 19 and follow the instructions.



---

## Search Argument Procedure

Each Info-Sys or EWS keyword describes one aspect of a program failure. The more precisely the keyword describes the program failure, the more selective the resulting search will be, thus yielding fewer matches or hits.

The following procedure explains how to use the set of keywords you developed as a search argument in Info-Sys or EWS.

1. Using the full set of keywords you developed, access Info-Sys or EWS and search the data base or microfiche. This will produce a number of matches (reported failures with characteristics similar to those of your problem).
2. Inspect each of the matches and eliminate those APAR corrections already applied to your system.
3. Compare each remaining APAR-closing description with your failure symptoms.
4. If you find a match, correct the problem by applying the correction.
5. If you find a match but a correction for the problem is not available, ask your IBM Support Center representative to contact you when a correction becomes available.

## Search Argument Technique

If you do not find a correction, broaden the search by using the following techniques in the order they are given:

1. Omit the release level keyword (xxx) from the search argument, thereby broadening the search to include similar failures on other release levels.
2. Begin dropping keywords from the right of the search argument to broaden the search even more.

## Preparing Authorized Program Analysis Reports (APARs)

Initiate an APAR only after you:

1. Check for usage errors
2. Develop a valid keyword string
3. Complete an unsuccessful search argument procedure.

## Initiating An APAR

As the support center helps you diagnose a DCF problem, they will gather the information needed to complete an APAR. If your problem is a new one, an IBM Support Center representative will issue an APAR number for the problem and begin the APAR procedure. At that time, be prepared to supply the following information:

- Customer number
- Operating system installed at your facility
- Current service level
- Keyword strings used to search Info-Sys or EWS
- CPU model and serial number.

## APAR-Supporting Documentation

Depending on the type of problem you are having, the Support Center representative might ask you to provide information that describes the DCF functions used, data base, environment, or operating instructions in effect when the problem occurred.

Although the documentation requirements vary, depending on the nature and severity of the failure, IBM does need the following documentation for all DCF-related problems:

- Console listing.
- A list of all specified SCRIPT command options.
- Complete text of all messages, including ID NUMBER.
- A failing test case, made as short as possible while still demonstrating the failure. This should be on tape if it is too long to allow easy keyboard entry. The user's profile, maclib, and command options should also be sent. The job to dump the tape should be included.
- Example of the failing output and the expected output.
- For MVS, a consolidated software inventory (SMPCSI) data set listing. For CMS, the latest PUT level plus any additional APARs or PTFs. For VSE, the MSHP list of maintenance.
- If your output device is a page printer, such as a 4250, a 3800 Printing Subsystem Model 3\*, or a 3820 Page Printer\*, submit a Font Library Index Program Report Listing with the APAR. Also, if a page segment is involved in the failure, you must submit the page segment with the APAR.
- If your output device is a PostScript\*\* device, submit a copy of your PostScript DCFINDEX file. If a PostScript image file is involved in the failure, you must

submit the image file also. The Support Center representative may require a copy of the AFM files and codepages files.

- For ABEND, WAIT, and LOOP problems, IBM also needs a storage dump.

With this information in hand, IBM can try to reproduce the problem and observe the failure symptoms.

If the Support Center representative is able to reproduce your problem before you submit an APAR, no documentation is usually needed.

**Note:** Be sure to put the APAR number the Support Center representative gives you on the top right-hand portion of each piece of documentation you submit. The Procedures Section of *Field Engineering Programming General Information* fully describes the necessary steps for submitting APAR documentation.

## Submitting Tapes or Cartridges as APAR Documentation

When you submit tapes as part of the APAR documentation, always use the smallest reel possible (that is, mini or 1200 feet). This keeps shipping costs to the minimum.

Before you ship user tapes or cartridges to IBM as part of APAR-supporting documentation, fill out the label (form number G229-2186) that IBM provides. Be sure to include:

1. Name—APAR submitter
2. Region—region number
3. Branch Office—branch office number
4. Customer Number—tape owner number
5. APAR Serial—assigned serial number
6. Mode and Density
7. Label—STD (standard), non STD (non-standard), No (no label)
8. File Format—fix blocked, unblocked, and so on
9. BLK Size—physical record size
10. REC Size—logical record size
11. Prog Used to Create—program used to create the tape, for example, DEBE, OS DUMP/RESTORE, and so on.

This label is intended to help prevent the loss of user tapes, should they become separated from other items of APAR documentation. IBM will return the media if requested to the submitter.



---

## Chapter 3. Program Overview

This chapter provides a general description of the Document Composition Facility (DCF), its operating environment, structure, and function.

---

### General Description

As shown in Figure 3 on page 24, DCF consists of SCRIPT/VS, the Generalized Markup Language (GML), the Dictionary Maintenance Program (DMP), and the Font Library Index Program (FLIP).

SCRIPT/VS is a text-formatting program. It reads input files with special controls and arranges them in a format appropriate for printing. Figure 4 on page 24 shows the SCRIPT/VS structure.

GML is a type of text markup that uses tags to describe the parts or elements of the document being formatted. A *starter set* of GML tags is supplied with DCF.

DCF provides an Application Programming Interface (API) to the SCRIPT/VS formatter for CICS applications, such as Advanced Text Management System-III (ATMS). Although the interface was designed specifically to support ATMS, it can be used by any CICS/MVS or CICS/VSE application to call the SCRIPT/VS formatter as an independent subroutine. For more information about this API, see the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*.

The DMP provides users with a means of creating and maintaining their own dictionaries to be used for hyphenation and spelling verification. This program generates three types of reports:

- |                                |  |
|--------------------------------|--|
| <b>Alpha dictionary report</b> | An alphabetical list of all words in the alpha dictionary file                                     |
| <b>Hyphenation report</b>      | An alphabetical list of all words in the alpha dictionary file with their hyphenation points shown |
| <b>Oldest usage report</b>     | A list of words that have not been frequently updated during dictionary maintenance.               |

The Font Library Index Program (FLIP) creates an index of the IBM page printer font licensed programs in a font library. SCRIPT/VS uses the index to determine whether a requested font is available for the current device and text orientation.

Certain kinds of gross program failures will be obvious if they occur: error messages, system abends, and so on. Most suspected failures of SCRIPT/VS, however, will probably appear as incorrect output, resulting from the use of functions in some unusual combination. For a complete description of the program's functions, see the following publications:

- *Document Composition Facility: SCRIPT/VS Text Programmer's Guide*
- *Document Composition Facility: SCRIPT/VS User's Guide*
- *Document Composition Facility: SCRIPT/VS Language Reference*
- *Document Composition Facility: Generalized Markup Language Starter Set User's Guide*

• *Document Composition Facility: Generalized Markup Language Starter Set Reference*

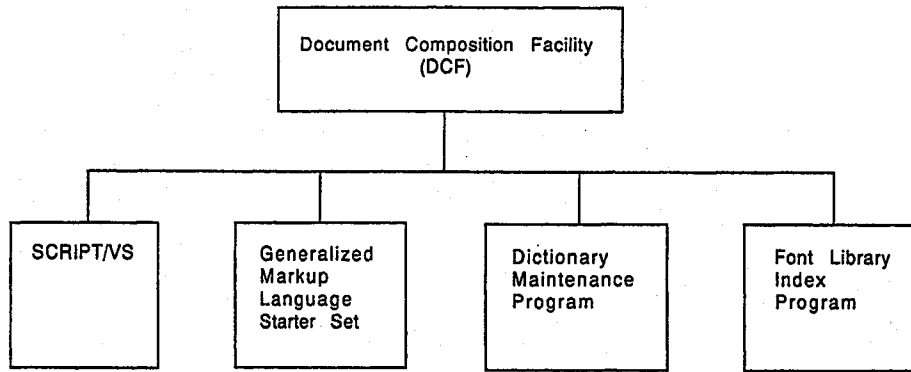


Figure 3. DCF Program Overview

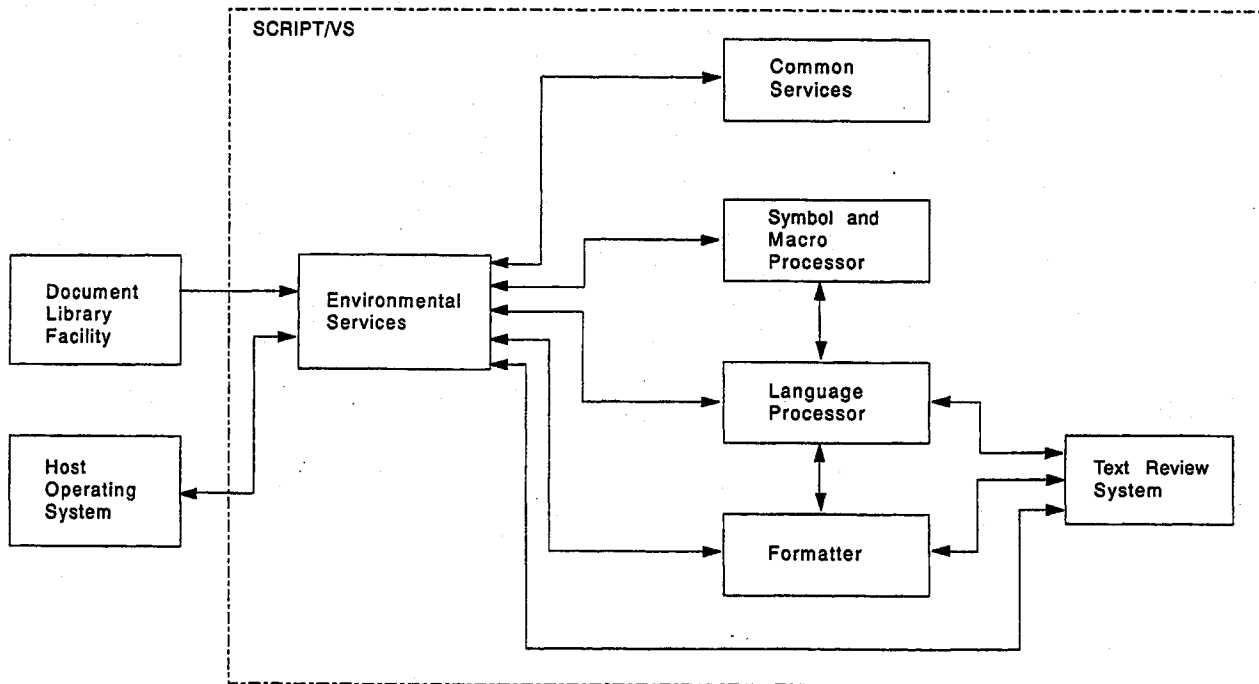


Figure 4. SCRIPT/VS Overview

## Input Files

SCRIPT/VS operates on a set of input files. The primary input file is named in the SCRIPT command that invokes SCRIPT/VS, and this file may imbed or append secondary files, which in turn may imbed or append still other files, and so on. In addition, there are three other ways to access files:

**Profiles:** The PROFILE option of the SCRIPT command specifies the name of a file to be processed before the primary input file. Typically this will contain symbol and macro definitions and will establish the working environment of the formatter. If the PROFILE option is not used, the program will search for a file named PROFILE to imbed. It may find one you did not want; you can prevent this by using the NOPROF option.

**The LIB Option:** The LIB option of the SCRIPT command must be used to identify where symbols and macros reside that are not defined in a document or in a profile. For TSO and CMS environments, the LIB option identifies a library name.

**The SEARCH Option:** In CMS, the SEARCH option of the SCRIPT command identifies where imbedded files reside. For TSO and batch environments (non-CMS), the SEARCH option identifies a library name.

## Types of Input

In addition to text, which is the basic data the program operates on, there are five types of input you can expect to see in an input file. Their most recognizable features are summarized below:

**Control Words:** Control words begin with a period (.), followed optionally by a single quotation mark ('), followed by a 2-character name.

Control words invoke the basic functions of DCF. A list of all SCRIPT/VS control words appears in the *Document Composition Facility: SCRIPT/VS Language Reference* and in the *Document Composition Facility: SCRIPT/VS Text Programmer's Quick Reference*.

**Macros:** Like control words, macros begin with an initial period, but their names may be from 1 to 10 characters long.

Macros represent sequences of control words, other macros, symbols, and text. They are defined inline with the .DM control word, or their definitions may be found in a library (as described in "The LIB Option" on page 25).

**Note:** Unintentional macro substitution is a common usage error.

**Symbols:** A symbol begins with an ampersand (&), followed by a name, with an end delimiter: a period (.), or a blank, or the end of an input line. Symbol names are case sensitive, that is, &AB and &ab are different symbols. Symbols may be compound; for example, if 'B' is set equal to 'xy', then &A.&B will be interpreted in the same way as &A.xy, which may in turn be a symbol.

Symbol substitution replaces symbols with other character strings, which may in turn be interpreted as control words, macros, symbols, or text. Symbols are defined with the .SE control word, or their definitions are sought in a library (as described previously).

**Note:** Omission of the trailing symbol delimiter may cause undesirable output and is a common usage error.

**GML Tags:** GML tags can appear anywhere in a document and begin with the GML delimiter which, by default, is a colon (:). A GML tag name can be up to 8 characters long and can consist of letters, numbers (except as the first character), and the special characters @, #, and \$. The end delimiter for a GML tag is a period (.), a blank, or the end of an input line. The starting and ending delimiters can be changed by the .DC or .GS control words. GML tags are typically associated with an Application Processing Function (APF); see below. The APF describes what is to be done with the tagged portion of the document.

**APFs:** IBM supplies a set of APFs, called the GML starter set, with DCF. These APFs are provided with the program in one of the following ways:

- For TSO, in a partitioned data set (PDS) called SCRIPT.R40.MACLIB
- For CMS, in a file called DSMGML4 MACLIB
- For background operation, in a document that the user has requested using the IMPORT command of DLF.

APFs, and the GML tags associated with them, are described in the *Document Composition Facility: Generalized Markup Language Starter Set User's Guide* and the *Document Composition Facility: Generalized Markup Language Starter Set Implementation Guide*.



## SCRIPT/VS Program Control Flow

1. After SCRIPT/VS is invoked by the SCRIPT command, initialization of the host environment is carried out under the control of DSMESBEG. (See "Module Naming Convention" on page 37 for a description of the abbreviated module names used in this section.)

DSMESBEG may call several modules, including:

- DSMLPLIT, which initializes the common communication area
- DSMESSCA, which scans the SCRIPT command and controls the processing of its options and defaults
- DSMLPDEV, which provides common device initialization.

Figure 5 on page 28 illustrates the module flow that passes control to the appropriate routine for font initialization and illustrates the control flow for font definition and usage.

2. Next, DSMLPPGO performs initialization needed for each pass through the program. If multiple formatting passes are specified with either the TWOPASS or FPASSES option of the SCRIPT command, DSMESBEG calls DSMLPPGO once for each pass. DSMLPPGO calls DSMTFMDE to initialize the formatter.
3. The program now enters its main loop, in module DSMLPMLP. At this point, three module names remain in the call stack: DSMESBEG, DSMLPPGO, and DSMLPMLP. DSMLPMLP controls the reading of text until all input files are exhausted and then returns to its caller.
4. From here on, the sequence of events is controlled by the text, with its controls, and by the results of laying out text in a formatted page. The program allows for extremely varied calling sequences: Modules can call each other, or themselves, or modules from a different functional area of the program; and it is possible to recognize only a few highlights as relatively constant features of program execution.

Figure 6 on page 29 shows the control flow within the SCRIPT/VS program.

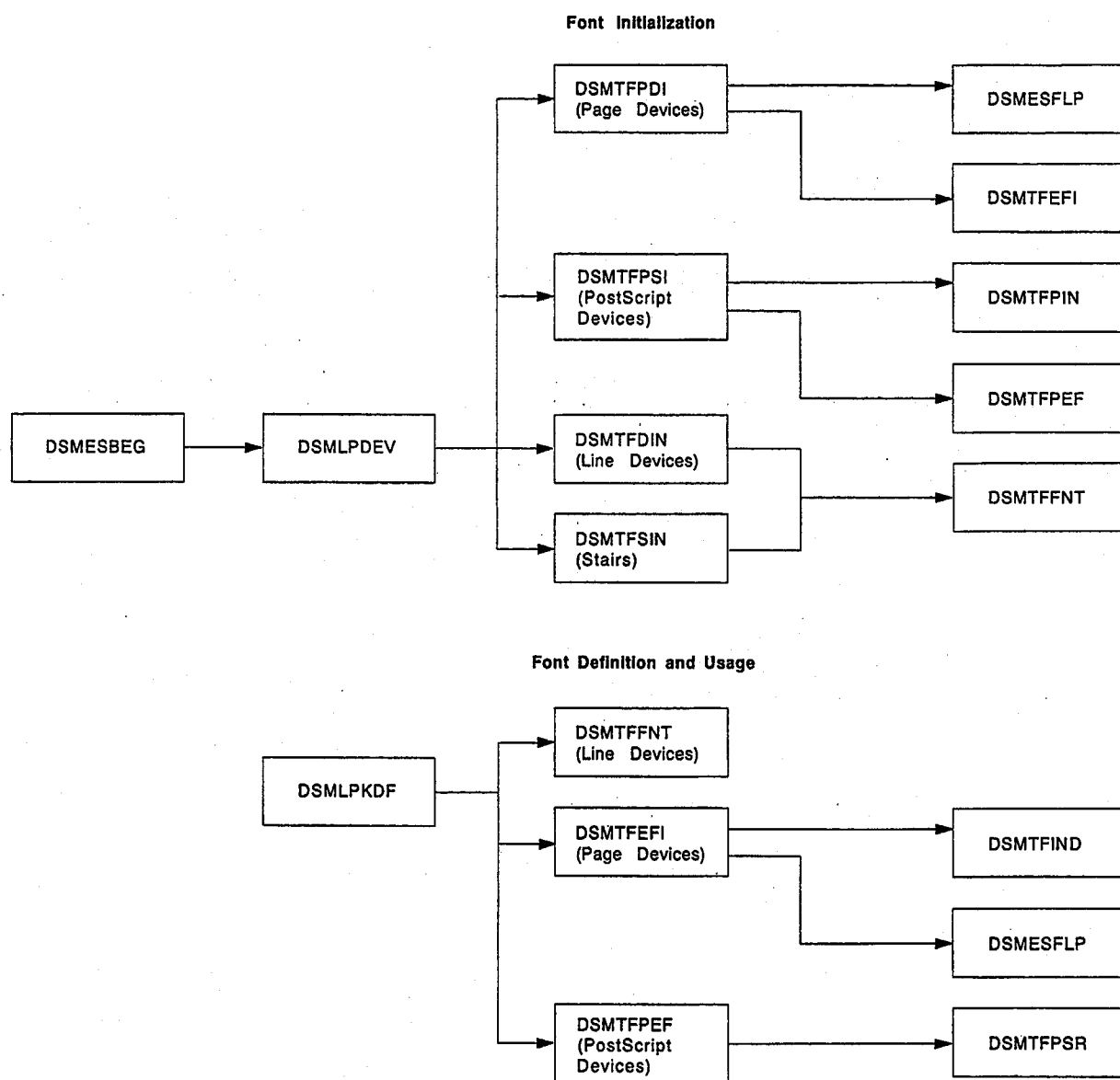


Figure 5. Font Initialization, Definition, and Usage

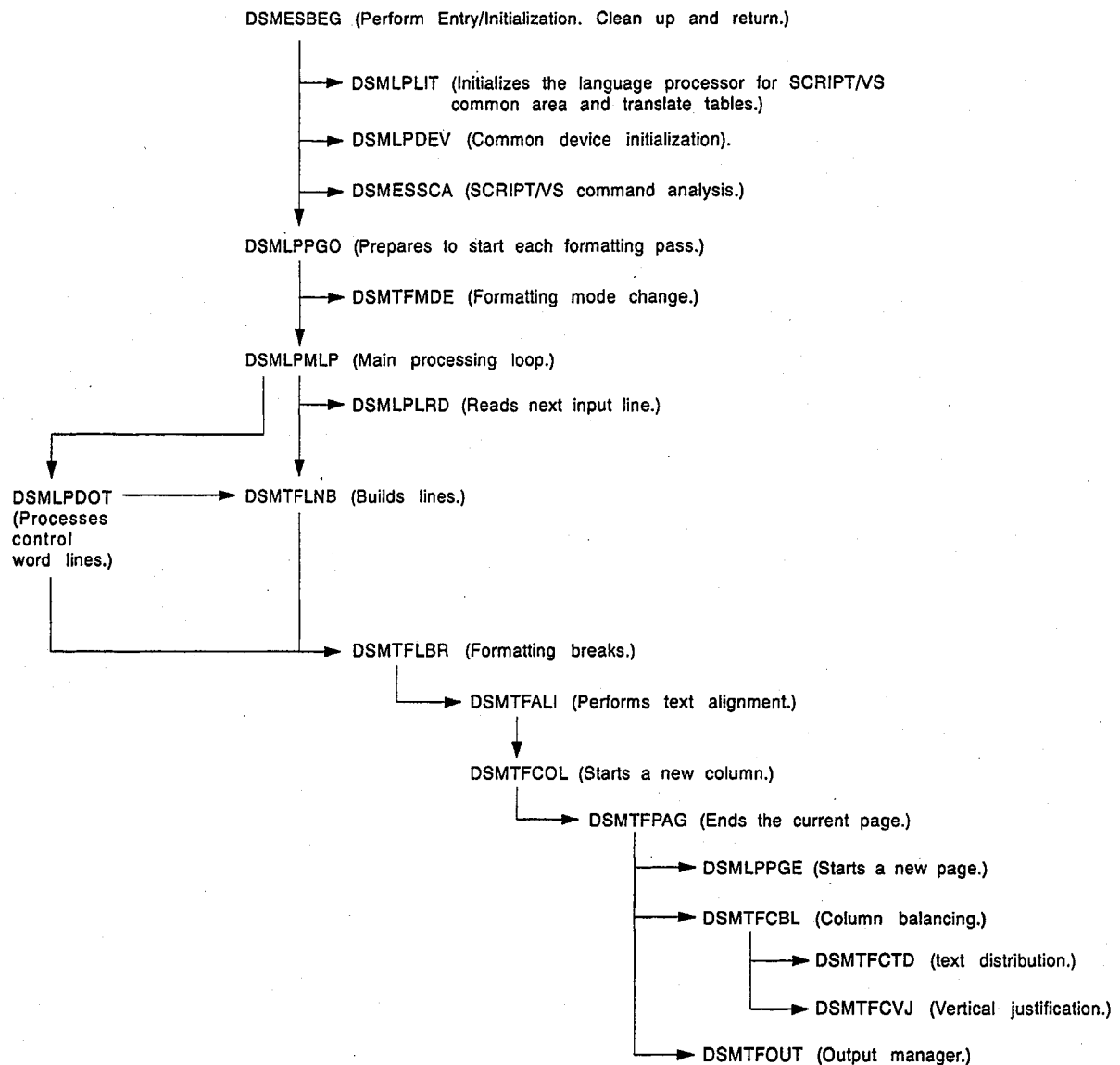


Figure 6. SCRIPT/VS Sequence Flowchart

## Operating Environment

This section explains DCF's programming systems, hardware requirements, and storage requirements.

## Programming Systems

DCF can be installed in the following foreground environments:

**CICS** CICS/MVS Version 2.1, under MVS/ESA, CICS/MVS Version 2.1, under MVS/SP, CICS/DOS/VS, Version 1.7, under VSE

**CMS** VM/SP Release 5.0 and above, VM/SP HPO Release 5 and above, VM/XA SP Release 2.0 and above, and VM/ESA Version 1.0 and above.

**Note:** In order to run DCF 4.0 with VM/SP5, you need to have the following programming temporary fixes (PTFs) installed:

- UV90415
- UM90047.

**TSO** MVS/SP 1.3.5 and above, MVS/SP, Release 2.2.0 and above, MVS/SP, Release 3.1 and above, and MVS/ESA SP 4.1 and above.

The ODF feature of DCF 4.0 is not supported under MVS/SP, Release 1.3.5. ODF can be used only with the VM and remaining MVS systems listed above. SMFF can be used only with the VM and MVS/TSO systems listed above. Neither ODF nor SMFF supports the VSE system.

ODF requires the IBM C for System/370 (Program Number 5688-039) licensed program for CMS and MVS.

DLF is run as a batch job or invoked as a callable subroutine. DLF can be installed in the batch environments of the following virtual storage operating systems, provided VSAM and the Access Method Services are available:

**MVS** MVS/370 SP, Release 1.3.0 and above, and MVS/XA Release 2.1.1 and above

**VSE** VSE/AF Version 2.1.3 and above (VSE/SP Version 2), VSE/VSAM Release 2 (Program Number 5746-AM2) and above, and VSE/ESA Version 1.1 and above.

DCF can also be installed with DLF in the batch environment.

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

## Hardware Requirements

**PROCESSORS:** DCF executes in an interactive environment under the control of the specified releases of CMS, TSO, or CICS and the processors on which they are designed to operate.

DLF and DCF operate in a batch environment on all IBM System/370 Models 138 and above, the System/390 and above, the IBM 3032 Processor and above, and the IBM 4332, 4341, 4361, 4381, and 9370 processors that are supported by VSE and MVS. Floating point hardware is required.

**TERMINALS:** In the foreground environments of CICS, CMS, and TSO, SCRIPT/VS can accept input from and produce output for the IBM 2741 Communications Terminal and the 3270 family of displays. In CICS/VS, ATMS-III also provides support for the 328x printers.

In a background environment, SCRIPT/VS cannot be used directly with terminals. However, programmable terminal systems, such as 8100 DOSF, can be attached to a supported host operating system through VTAM and can invoke SCRIPT/VS through remote job entry (RJE).

**PRINTERS:** The IBM family of Advanced Function Printing (AFP) page printers are all-points-addressable (APA) printers. AFP page printers allow you to use:

- Typographical fonts of varying sizes and rotations
- Electronically-designed forms
- Graphics, such as line art, illustrations, photographs, logos, and signatures.

You can generally print the same document on any one of IBM's AFP family of page printers without changing the source, although there are some functional differences among these printers (for example, the valid printable area on the page varies from printer to printer). The IBM family of AFP page printers includes:

- IBM 3800 Printing Subsystem Model 3 and 6
- IBM 3820 Page Printer
- IBM 3825 Page Printer
- IBM 3827 Page Printer
- IBM 3828 Advanced Function MICR Printer
- IBM 3835 Page Printer
- IBM 3812 Page Printer
- IBM 3816 Page Printer
- IBM LaserPrinter 4028
- IBM 3900 Advanced Function Printer
- IBM 4224-2xx Page Printer
- IBM 4234 Page Printer

and any other printer compatible at the datastream level.

IBM AFP page printers are supported under MVS, VM, or VSE. The Print Services Facility (PSF) licensed program (Program Number 5665-275 for MVS, Program Number 5664-198 for VM, or Program Number 5666-319 for VSE) is required for you to print DCF formatted page output on page printers.

The IBM 4250 Printer is an electroerosion printer that produces camera-ready copy. The Composed Document Print Facility (CDPF) licensed program is required to print DCF formatted output on the 4250.

**Note:** Although you can use DCF to format for a 4250 printer, the 4250 Printer and CDPF are no longer available from or supported by IBM.

SCRIPT/VS can format documents for PostScript devices configured to accept 8-bit ASCII, such as the IBM LaserPrinter 4019.

SCRIPT/VS documents can be formatted for printing on the IBM 3800 Printing Subsystem Model 1. For VSE, use of the 3800 Printing Subsystem Model 1 requires the VSE IBM 3800 Printing Subsystem Independent Release licensed program (Program Number 5747-CC1).

Documents can also be formatted for printing on the IBM 1403 line printer or any printer functionally equivalent to this IBM printer at the data stream level.

**STORAGE DEVICES:** DCF uses storage devices that are supported by the environment in which it operates.

## Storage Requirements and Estimates

**Real Storage:** Real storage requirements for DCF or DLF are dependent on each 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.

**Direct-Access Storage:** DCF requires (approximately) the following direct-access storage:

DCF Storage Estimates	CMS (4K Blocks)	3380 Tracks
Load Module Size	157	35
GML Macro Library	273	40
GML Profiles	47	6
Hyphenation and Spelling Verification Dictionaries	628	55

The DLF load module requires 30 tracks of 3330-11 direct-access storage. Additional storage is required for any data-conversion routines, the exact amount being determined by the size of the routine. Each installation must determine how much additional direct-access storage is required for document storage and directory space.

**Virtual Storage:** DCF requires approximately 435K bytes of virtual storage for the load modules. Each spelling-verification dictionary included in the SCRIPT/VS load module increases its size by an additional 27K to 162K bytes.<sup>3</sup> The SCRIPT Mathematical Formula Formatter (SMFF) adds 50K bytes to load the module size. Because the code is reentrant, all users can share the same copy; additional users do not increase storage space requirements. DCF also requires working storage to format documents. The complexity of the documents being formatted determines the size of working storage required. In the MVS environment, the minimum amount of storage recommended to execute SCRIPT/VS is one megabyte when formatting for advanced printer functions. The VSE environment requires a minimum background partition size of 2000K bytes to execute SCRIPT/VS and GML when formatting for advanced printer functions.

DLF requires approximately 200K bytes of virtual or working storage. When SCRIPT/VS is to be used with the library program, an additional 15K bytes of virtual storage (a total of 215K bytes) must be allocated in addition to the storage necessary for SCRIPT/VS.

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

---

<sup>3</sup> Dictionaries are dynamically loaded as needed in all environments. Previous CICS storage management restrictions preclude link-editing any spelling verification dictionaries with the SCRIPT/VS load module in the CICS environment.

## Structural Overview

This section explains DCF's program structure and conventions.

## Program Structure

The DCF program structure consist of load modules and source modules.

### DCF Load Modules

DCF load modules are obtained by link-editing the various subcomponent object files. Each environment has a set of these load modules.

#### • CMS

<b>SCRIPT</b>	Bootstrap Loader (invokes either shared segment or DSM4)
<b>DSM4</b>	Disk-Resident SCRIPT/VS Program

**Note:** The above modules may be renamed by your systems programmer at installation time.

<b>DSMCMF40</b>	Font Library Index Program
<b>DSMSEG4</b>	A discontinuous shared segment (DCSS) designed to run on CMS 5.0.
<b>DSMSEG4X</b>	A DCSS designed to run on CMS 5.5 and above (intended to be installed above the line).
<b>DSMSEG4B</b>	A DCSS designed to run on CMS 5.5 and above (intended to be installed below the line).

#### • MVS

<b>DSMTSS40</b>	TSO SCRIPT/VS Program
<b>DSMTXS40</b>	TSO SCRIPT/VS Program
<b>DSMLOS30</b>	DLF SCRIPT/VS Program
<b>DSMLXS30</b>	DLF SCRIPT/VS Program
<b>DSMATS30</b>	CICS SCRIPT/VS Program
<b>DSMBOF40</b>	Font Library Index Program
<b>DSMDAVER</b>	Danish Dictionary
<b>DSMDUVER</b>	Dutch Dictionary
<b>DSMEAVER</b>	American English Dictionary
<b>DSMECVER</b>	Canadian English Dictionary
<b>DSMEUVER</b>	U. K. English Dictionary
<b>DSMFCVER</b>	Canadian French Dictionary
<b>DSMFIVER</b>	Finnish Dictionary
<b>DSMFNVER</b>	French National Dictionary
<b>DSMGEVER</b>	German Dictionary
<b>DSMICVER</b>	Icelandic Dictionary
<b>DSMITVER</b>	Italian Dictionary
<b>DSMNOVER</b>	Norwegian Dictionary
<b>DSMPOVER</b>	Portuguese Dictionary
<b>DSMSPVER</b>	Spanish Dictionary
<b>DSMSWVER</b>	Swedish Dictionary

#### • VSE

<b>DSMLDS30</b>	DLF SCRIPT/VS Program
<b>DSMATS30</b>	CICS SCRIPT/VS Program
<b>DSMBDF40</b>	Font Library Index Program
<b>DSMEXITD</b>	GML Macro to DLF Library Exit



<b>DSMEXITA</b>	GML Macro to CICS Exit
<b>DSMDAVER</b>	Danish Dictionary
<b>DSMDUVER</b>	Dutch Dictionary
<b>DSMEAVER</b>	American English Dictionary
<b>DSMECVER</b>	Canadian English Dictionary
<b>DSMEUVER</b>	U. K. English Dictionary
<b>DSMFCVER</b>	Canadian French Dictionary
<b>DSMFIVER</b>	Finnish Dictionary
<b>DSMFNVER</b>	French National Dictionary
<b>DSMGEVER</b>	German Dictionary
<b>DSMICVER</b>	Icelandic Dictionary
<b>DSMITVER</b>	Italian Dictionary
<b>DSMNOVER</b>	Norwegian Dictionary
<b>DSMPOVER</b>	Portuguese Dictionary
<b>DSMSPVER</b>	Spanish Dictionary
<b>DSMSWVER</b>	Swedish Dictionary

## DCF Source Modules

The following assembler language files are provided in source and object form:

- **DSMCSDM (Message Definitions).** The tables in this file give the text of messages issued by SCRIPT/VS.
- **DSMTSDM (Message Definitions).** The tables in this file give the text of additional messages issued by SCRIPT/VS in the TSO environment.
- **DSMCSPDB (Phrase Definitions).** The tables in this file give the text of literal phrases used by SCRIPT/VS in formatting documents and displaying messages.
- **DSMxxUOT (Environment-Dependent Options).** The tables in this file define the names of the SCRIPT command options and establish default values for those options. There is a separate file for each operating system environment under which SCRIPT/VS operates:

<b>DSMATUOT</b>	CICS/VS
<b>DSMCMUOT</b>	CMS with VM/370
<b>DSMLDUOT</b>	DLF with VSE
<b>DSMLOUOT</b>	DLF with MVS
<b>DSMTSUOT</b>	TSO with MVS
<b>DSMVXUOT</b>	CMS with VM/XA or MV/ESA

- **DSMLPBCS (Box Character Sets).** The tables in this file define the box character sets that can be specified with the CHARS parameter of the .BX control word, and establish the characters they will use to construct boxes and rules on line devices.
- **DSMLPLDT (Logical Device Tables).** The tables in this file define the logical devices that can be specified with the DEVICE option of the SCRIPT command, and establish the characteristics of those logical devices.
- **DSMLPPDT (Physical Device Tables).** The tables in this file define the physical devices used by the SCRIPT logical devices, and describe the physical characteristics of each device.
- **DSMTFFIB (Line Printer Font Information).** The tables in this file define the fonts that can be specified with the CHARS option of the SCRIPT command for line devices, and establish the characteristics of those fonts.

- **DSMTRNLT (National Language Character Tables).** The tables in this file define the correspondence between the internal (code points) and external (terminal or printer) representations of accented characters, determine which characters will be treated as word delimiters or punctuation for purposes of spelling verification and hyphenation, and define the index sort sequences for the 15 IBM-supplied languages.

A macro library is also provided for use in assembling these source files. It contains the following macros:

DSMACTT  
 DSMATDEF (CICS only)  
 DSMBCS  
 DSMCMDEF (CMS only)  
 DSMDEF  
 DSMECOB  
 DSMEICB  
 DSMFIB  
 DSMHHCC  
 DSMIXTT1  
 DSMIXTT2  
 DSMLDT  
 DSMLLDEF (DLF only)  
 DSMMSG  
 DSMPDB  
 DSMPDT  
 DSMSCO  
 DSMSLDT  
 DSMSMSV  
 DSMSPDT  
 DSMTSDEF (TSO only)  
 DSMWDPT

If you want to run DCF in XA exploitation mode, you must use the H Assembler to assemble these modules.

If you don't have the H Assembler, you can still assemble these modules with the F Assembler, but you must delete the A mode, R mode compiler options. The resulting code will not run in an XA environment.

## Program Conventions

There are 2 types of program conventions: Register Conventions and Module Naming Conventions.

### Register Conventions

The following register conventions are used in DCF:

<u>Register</u>	<u>Usage</u>
<b>R0</b>	Parameter register Reason code on return (where defined) Message number
<b>R1</b>	Parameter register
<b>R2-R10</b>	Work registers
<b>R11</b>	Address of beginning of common data area (DSMSCOM)
<b>R12</b>	Module base register
<b>R13</b>	Address of module save area (DSMSAVD) in the call stack
<b>R14</b>	Return address
<b>R15</b>	Entry point address on entry Return code on return (where defined).

Among the work registers (R2 through R10), lower-numbered registers are generally used for computation and higher-numbered registers are generally used for work area base registers.

### Module Naming Convention

Every module in DCF has an 8-character name beginning with DSM. The next two characters name the functional area to which the module belongs; the last three characters uniquely identify the module. For example, the module name DSMLPMLP can be broken down into the following parts:

<b>DSM</b>	Program identifier
<b>LP</b>	Functional-area identifier
<b>MLP</b>	Module identifier.

In this example, DSMLPMLP is identified as a DCF module from the Language Processor functional area that contains instructions for the main processing loop.

## Communication between Modules of the DCF program

With a few exceptions, the program modules link to each other through code that is contained in the common data area (DSMSCOM). The first instruction of the module branches to an offset in DSMSCOM, which is pointed to by register 11. The common code saves the caller's registers, sets up a new module save area, loads the module base address into register 12, and branches to an offset of X'18' from the beginning of the called module.

The 24 bytes between the module's entry point and the return from the common entry code are occupied by an unconditional branch instruction, an 8-character EBCDIC module name (for example, DSMLPMLP) and the assembly date.

---

## Chapter 4. Service Aids

SCRIPT/VS provides service aids at both the text input level and at the internal programming level:

- “Text Input Service Aids” describes service aids provided by options of the SCRIPT command and by SCRIPT/VS control words.
- “Internal Program Service Aids” on page 42 describes service aids provided within the SCRIPT/VS program.

---

### Text Input Service Aids

The following facilities of SCRIPT/VS are available to all users and are used to identify errors in the specification of control words, macros, and GML tags.

### SCRIPT Command Options

Some of the SCRIPT command options are useful for diagnostic purposes when they are specified to format an input file that might contain errors:

- UNFORMAT causes SCRIPT/VS to print all input lines as they appear in the input file. The lines that are produced in an unformatted listing represent all (and only those) lines that will be processed by SCRIPT/VS. Lines not processed as a result of a .GO control word or ignored because of a .CS control word, for example, are not shown in the unformatted listing. The UNFORMAT option is ignored for page printers.
- MESSAGE(ID) causes message identifiers to be printed with all messages.  
**Note:** The message identifier is required whenever you contact IBM for help.
- MESSAGE (TRACE) causes, for some messages, a trace of the sequence of imbedded files to be printed along with the message. For more details on messages, see the *Document Composition Facility: Messages*.
- MESSAGE (DELAY) causes error messages to be printed at the end of the output document instead of being displayed at a terminal during processing. (In the batch processing environment, messages are always delayed.)
- NOSPIE inhibits the special program-interrupt exit in SCRIPT/VS, leaving the system's standard exit in its place. Using this option will give you host-system messages and dumps for interrupts. The SPIE option generates messages identifying the program the interrupt occurred in and at what offset. (The SPIE option cannot be used in background or ATMS III.)

These options should always be used before you call IBM to obtain HOST symptoms for keywords that relate to abnormal end (ABEND) situations only.

- NUMBER causes the file name and line number of the last input line read to be printed to the right of each output line. The NUMBER option is ignored for page devices.
- PAGE allows you to print selected pages of output.
- PSOUT allows you to produce PostScript\*\* output in EBCDIC for debugging on the host.

**Note:** You must produce ASCII output to actually print on a PostScript\*\* device.

## The .IT Control Word

The input substitution trace should be used whenever a failure is suspected in symbol or macro substitution, or whenever a control word is suspected of failing. The trace can be used locally around the section in question to avoid unmanageable amounts of output. See the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for a description of this tool and suggestions for its use.

## Terminal-Oriented Control Words

The following control words enable you to debug control-word sequences interactively:

- .IT allows a trace with interactive substitution or insertion of controls.
- .TE and .TY allow reading and writing to a terminal.
- .SY allows you to use trace and debugging aids (see "The .SY Control Word" on page 44 for more information about the .SY control word).

## The .MG Control Word

The .MG control word can be used to write out messages with severity codes that are appropriately acted upon by SCRIPT/VS. This facility can be used to provide diagnostic messages for macros, user-written APF's, and for "debugging" complicated user written text logic. See the *Document Composition Facility: SCRIPT/VS Text Programmer's Guide* for a description of this tool and suggestions for its use.

## Messages and Codes

DCF provides extensive, automatic error checking of input controls and data. Over 300 explicit error messages are issued. Those messages are documented in *Document Composition Facility: Messages*.

**Program Module Issued Messages:** The messages are identified by a 10-character label of the form **DSMmmmmnnnx**. The elements of this code word are as follows:

1. **DSM** identifies DCF.
2. **mmm** is a three-character identifier (the last three characters of the module name) of the program module that caused the message to be issued.
3. **nnn** is a three-digit message number.
4. **x** is a message severity code that has one of the following values and associated return code:

<i>Severity Code</i>	<i>Description</i>	<i>Return Code</i>
R	response required	0
I	informational	0
W	warning	4
E	error	8
S	severe error	12
T	terminating error	16

**Message (.MG) Control Word Issued Messages:** Messages sent by use of the .MG control word are distinguished from program messages by a prefix of three plus signs (+++). Messages of this type that are sent by IBM-supplied APFs are documented in the *Document Composition Facility: Generalized Markup Language Starter Set Reference*.

## Internal Program Service Aids

These facilities of the SCRIPT/VS program are available to development and maintenance personnel for use in identifying internal programming errors.

## Internal Programming Errors

Internal programming errors may be reported in two different ways:

**Program Logic Checks:** When SCRIPT/VS detects internal errors, such as invalid control sequence codes, it can issue an error message or execute an operation exception. The error condition is documented at the point where it is detected.

**System Checks:** When the operating system detects an internal error, such as invalid chaining, the SCRIPT/VS interrupt exit receives control; it issues an error message identifying the type of program exception, the failing module, and the offset location of the error.

## FE Patch Areas

**DSMCSFEP:** A single 4096-byte module, named DSMCSFEP, has been included to serve as a patch area for future program repairs. There are no patch areas in the individual modules.

**Control Block Areas:** An eight-byte area (FEPATCH) in DSMSCOM (Common Control Block) is reserved for FE patch and diagnostic purposes.

**Note:** All FE patch areas are reserved for FE maintenance use only.

**Work Buffer Fences/Patch Areas:** One word fields are provided for FE patch/diagnostic purposes after the following work buffers:

- BUFF2
- HIDDENBF
- ERRBF
- TOKBUF
- TOKABUF

**Common Module Entry and Exit:** An eight-byte area in the Common Module Entry and Exit Code section of the Global Communications Area is provided for FE use.



## The .ZZ Control Word

The .ZZ control word is used to obtain certain diagnostic information.

<b>.ZZ</b>	{MODNAME <i>name</i> } {DICT}
------------	----------------------------------

**MODNAME** Will return the offset of the module *modname* from the beginning of the SCRIPT load module and the date on which it was assembled. The date is displayed either as "yymmdd" or as "yy.nnn" where "yy" is the year and "nnn" is the Julian, or shop, date. For example, "January 1, 1991" would be displayed as "91.001." The *modname* can be an 8-character name starting with "DSM."

**DICT** Will return information about the word most recently found during a dictionary search. Such a search could be for spelling verification of a word or for finding the hyphenation points of a word.

This information is returned as follows:

- The most recent word as it was entered by the user (if hyphenation was also requested, slashes in the word indicate the returned hyphenation points)
- The form of the word as it was found in the dictionary
- The dictionary in which the word was found.

**Remarks:** Use of MODNAME will find only the modules that reside in the main SCRIPT/VS load module.

1. In all environments, modules that won't be found include FLIP, dictionaries that aren't linked, and the following modules:

DSMAAAAA  
DSMLPBCS  
DSMLPLDT  
DSMLPPDT  
DSMTFFIB  
DSMTF66F  
DSMZXXXX  
DSMCSMDM  
DSMCSPDB  
DSMCSFEP  
DSMCSVRS

2. In MVS/DLF, since the following modules are not contained in the main SCRIPT/VS load module, they are also not found:

DSMESSSL  
DSMESFLP  
DSMESLIB  
DSMESPPSP  
DSMESSRC  
DSMESUTL  
DSMESUOT

3. In MVS/TSO, since the following modules are not contained in the main SCRIPT/VS load module, they are also not found:

DSMTSDMG  
DSMTSDSR  
DSMTSMAD  
DSMTSTSO  
DSMTSMDM  
DSMTSUOT  
DSMESFLP  
DSMESPSP  
DSMXTKSY

## The .SY Control Word

The .SY [System Command] control word is only supported in the interactive environments of CMS, CICS, and TSO. In CMS, SCRIPT/VS passes a line to CMS for processing as a CMS or CP command line. In TSO, the line is stacked until the end of SCRIPT/VS processing.

In ATMS, the .SY control word is passed to an installation provided host command processor. No system or data integrity exists unless the installation code introduces a problem.

Use the .SY control word if you want to perform some CMS or CP command when your SCRIPT file is processed, or, in TSO if you want some command to be performed after formatting is completed.

<b>.SY</b>	line
------------	------

**line** Is a CMS, CP, or TSO command line. In CMS, if line is omitted, CMS subset is entered.

### Remarks:

1. NO CMS command or user program is allowed that requires the use of the same area of storage that is being used by SCRIPT/VS. CMS commands that are valid in CMS SUBSET are valid on the .SY command. An invalid SUBSET command results in a return code of -2.
2. To test whether a command executed successfully in a SCRIPT file, you can test the value of the reserved system &\$RET. For example:  

```
.if &$ret ne .qu
```

causes SCRIPT/VS to terminate processing if the return code from the last executed CMS command is not zero.
3. If the command does not exist or was not executed at all, &\$RET is set to a negative value. This would be the case for nonexistent commands in CMS, and for all commands in environments other than CMS.
4. In the TSO environment, if the .SY control word is used more than once, the commands will be executed in the order in which they were encountered.
5. No command or program should be used that will free the storage controlled by SCRIPT/VS (for example, XEDIT in CMS).

## User-defined exit for .SY

The user-defined exit module (DSMXTKSY) allows you to write an assembler routine to determine the action your system will take when an .SY control word is encountered. To use this exit, you must write an assembler program called DSMXTKSY that will be given control by SCRIPT/VS before processing the .SY control word.

Possible actions that can be taken when the .SY control word is encountered are as follows:

- Process the .SY control word only if SYON is specified on the SCRIPT command
- Process the .SY control word, even if SYOFF is specified on the SCRIPT command
- Ignore the .SY control word completely
- Disallow the .SY control word, even if SYON is specified on the SCRIPT command.

## Register Conventions

When SCRIPT/VS calls DSMXTKSY, the following general registers contain significant information:

- |            |   |
|------------|---|
| <b>R1</b>  | Register 1 contains the address of the parameter list sent to DSMXTKSY by SCRIPT/VS.  |
| <b>R11</b> | Register 11 contains the address of the SCRIPT command control block. This register should not be altered by your DSMXTKSY routine or by any subroutines your routine may invoke.     |
| <b>R12</b> | Register 12 is, by convention, the base register used by all SCRIPT/VS modules. Your routine should copy the contents of Register 15 into Register 12 and use it as a base register.  |
| <b>R13</b> | Register 13 contains the address of a preallocated, prechained register save area. This register should not be altered by your routine or by any subroutines your routine may invoke. |
| <b>R14</b> | Register 14 contains the address to which your routine should return when finished.   |
| <b>R15</b> | Register 15 contains the entry point address of your routine. This value should be copied into Register 12.   |

## Input Parameter List

The system command parameter list, which is addressed by Register 1 on entry to the routine, must contain the following fields:

- |                    |  |
|--------------------|--|
| <b>bytes 1-60</b>  | On entry, these 15 fullwords have not been filled in. DSMXTKSY should save the contents of the 15 registers in this area. DSMXTKSY MUST ensure that the contents of the registers are the same upon exit from DSMXTKSY as they were upon entry.  |
| <b>bytes 61-64</b> | On entry, this fullword has been filled in with the address of a 256-byte work area which DSMXTKSY can use. All storage used by DSMXTKSY should be from this work area.  |
| <b>bytes 65-66</b> | On entry, this halfword has been initialized to zero. On exit, this halfword should contain the return code from DSMXTKSY to SCRIPT/VS, to tell SCRIPT/VS what action to take for the current .SY control word. The recognized return codes are: |

- 0 = Process .SY control word according to SYON/SYOFF setting. If SYOFF is in affect, message 313S will be issued.
- 4 = Process .SY control word regardless of SYON/SYOFF setting.
- 8 = Ignore the .SY control word regardless of SYON/SYOFF setting. No further processing of the .SY control word will be done.
- 12 = Disallow .SY regardless of SYON/SYOFF setting. Message 313S will be issued, regardless of the SYON/SYOFF setting.
- 16 = Indicate an unexplainable condition has happened. DCF will issue message 400 and terminate.

**bytes 67-68** On entry this halfword contains the length of the .SY control word parameter list.

**bytes 69-324** On entry this field contains the .SY control word command line to be processed. The first four bytes, ".SY" have been stripped out and only the command remains.

### Routine Entry and Exit Protocol

The following entry protocol is suggested for the system command routine:

```
DSMXTKSY CSECT
    USING PARMLIST,R1
    STM R14,R12,PARMRV SAVE ALL THE REGISTERS
    USING DSMXTKSY,R12
    LR R12,R15
    L R2,PARMWA GET ADDRESS OF THE WORK AREA
    USING WORKA,R2
    MVC PARMRC,=H' ' INITIALIZE RETURN CODE TO ZERO
```

The above assumes the parameter list has been defined as:

```
PARMLIST DSECT
PARMRV DS 15F SAVE AREA FOR REGISTERS
PARMWA DS F ADDRESS OF WORK AREA
PARMRC DS H RETURN CODE
PARMSYLG DS H LENGTH OF PARMSY
PARMSY DS CL256 .SY CONTROL WORD TO PROCESS
```

A WORKA DSECT has been defined. The following is an example:

```
WORKA DSECT
WORK DS CL256
```

The following EXIT protocol is suggested for the system command routine:

```
LM R14,R12,PARMRV RESTORE REGISTERS
BR R14 RETURN TO CALLER
```

### Sample DSMXTKSY Processing

One possible use of the DSMXTKSY interface is to allow a certain subset of system commands even if SYOFF has been specified. To do this, a table of allowable system commands must be created, and DSMXTKSY must be used to search that table for the command given on the .SY control word. In conjunction with suggested entry and exit protocol, the following shows sample logic for accomplishing this task:

1. Parse the system command from the .SY control word line in the parameter list pointed to by register 1
2. Translate the system command to uppercase
3. Look for the system command in a table of system commands which are allowed in SYOFF mode
4. If the system command is found, set return code to 4.

**Note:** If the system command is not found, the return code has already been initialized to zero to indicate to SCRIPT/VS to process the .SY control word according to the SYON/SYOFF setting.

### Installation Instructions

The following procedure is recommended for replacing an IBM supplied stub routine with your routine in the CMS environment:

1. Compile or assemble DSMXTKSY to produce a TEXT file. If your routine calls any other modules, compile or assemble them also.
2. Link and access the SCRIPT/VS Release 4.0 installation minidisk in READ/WRITE mode as your A-disk.
3. Move the IBM-supplied DSMXTKSY to another disk to save it.
4. Copy your DSMXTKSY TEXT file onto the A-disk. If your routine calls any other modules, copy them also.
5. Issue the following CMS command:

```
dsmgnd4
```

This EXEC creates a new DSM4 MODULE (disk resident SCRIPT MODULE).

6. If your installation has defined a shared segment for SCRIPT/VS, make sure the new SCRIPT module will fit entirely within the defined shared segment. Then issue the following CMS command:

```
dsmgns4
```

This EXEC reloads the SCRIPT shared segment.

7. Copy the new SCRIPT MODULE file onto the public minidisk where SCRIPT/VS normally resides at your installation.

The following procedure is recommended for replacing an IBM-supplied stub routine with a routine in the TSO environment:

1. Compile or assemble DSMXTKSY to produce an object file suitable as input to the OS/VS linkage editor. If your routine calls any other modules, compile or assemble them also, and append them to the DSMXTKSY object file.
2. Find out the name of the DCF Release 4.0 load library that contains the installed IBM-supplied dictionary routines. This library is named

DCF.R40.DCFLOAD in the example below. DCF may have been placed in another load library, such as SYS1.CMNDLIB, at your installation.

3. Execute the following job to replace the stub routine with your routine in the DCF load library:

```
//INSTALL JOB ,SCRIPTVS,MSGLEVEL=(1,1)
//LINKALG EXEC PGM=HEWL,REGION=768K
//          PARM='RENT,XREF,LIST,LET,
//          NCAL,SIZE=(46 K,96K)'
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(5,1))
//OBJSY DD DSN=DSMXTKSY.OBJ,DISP=SHR
//DCF DD DSN=DCF.R4 .DCFLOAD,DISP=SHR
//SYSMOD DD DSN=DCF.R4 .DCFLOAD,DISP=OLD
//SYSLIN DD *
INCLUDE OBJSY
INCLUDE DCF (DSMTSS4 )
      (insert order statements here)
ENTRY DSMTSTSO
ALIAS SCRIPT
NAME DSMTSS4 (R)
/*
//
```

**Note:** The current correct linkage editor ORDER statements should be obtained from the DCF installation sample DSMTSIVL in DCF.R40.DCFSAMP.

**Examples:** The .IM [Imbed] control word issues an error message if the designated file is not found. The CONTINUE option of the SCRIPT command allows SCRIPT/VS to continue processing after this error. The following .CIM [Conditional Imbed] macro would allow SCRIPT/VS to test for the existence of a file in CMS before attempting to imbed it, and only imbed it if it is available:

```
.dm cim () /.sy state &*1 * *
.dm cim () /.if &$ret eq .im &*
```

The macro would be invoked as follows:

```
.cim filename
```

## Module Identification

Each module uses a macro (DSMENTRY) which provides the module name, the last modification date, and the version, release, and modification-level of the module. These parameters are compiled as constants in the object code so that each module can be easily located and its modification date determined in a storage dump.

## The Module Call Stack

As the program executes, each entry to a new module is recorded in the two most recent links of a chain of module save areas, called the module call stack. One save area is diagrammed in Table 1 on page 49.

At entry, the contents of registers 14 through 12 are saved in the previous area, register 13 is loaded with the address of the next save area in the chain, and the last three characters of the module name are inserted in the module name field (bytes 2 through 4). When control returns to the calling module, the called module releases its save area for reuse. If a dump is taken during the execution of DCF, it is possible to tell, by tracing back up the call stack, the sequence of calling

modules within which the current execution is imbedded. It is *not* possible, though, to tell the entire sequence of modules that have been called because modules that have processed successfully and returned to their callers have released their save areas.

Offset decimal(hex)	Length decimal(hex)	Description
0 (0)	1 (1)	(Reserved)
1 (1)	3 (3)	Module identifier
4 (4)	4 (4)	Address of previous save area
8 (8)	4 (4)	Address of next save area
12 (C)	60 (3C)	Save area for registers 14 through 12
72 (48)	56 (38)	Scratchpad work area for this module

Table 1. DCF Module Save Area. Addressed by Register 13.

## Common Entry and Exit Logic

Most modules branch to the common entry and exit code provided in DSMSCOM to perform standard initialization and termination functions, such as saving and restoring registers, establishing base registers for module addressability, and setting return codes.

The common entry and exit code provides a single point of control for most module-level activity and can be used for module flow tracing, examination of parameter and return code register values, and so on.

## System Tools

The following system tools are applicable for DCF problem determination:

- CICS
  - CICS Debug Facility
  - CICS Dump Facilities
- VSE
  - SDAIDS
  - VSE Dump Facilities
- MVS
  - Generalized Trace Facility
  - MVS Dump Facilities
- TSO
  - TSO TEST Command
  - TSO Debug Facilities
  - MVS Dump Facilities
- VM/370 CMS
  - CP TRACE Command
  - CMS SVCTRACE Command
  - CMS DEBUG Command

---

## List of Abbreviations

<b>AD</b>	Alpha Dictionary
<b>APAR</b>	Authorized Program Analysis Report
<b>APF</b>	Application Processing Function
<b>ATMS</b>	Advanced Text Management System
<b>BR</b>	Branch
<b>CDS</b>	Control Data Set
<b>COB</b>	Command Option Block
<b>DCF</b>	Document Composition Facility
<b>DLF</b>	Document Library Facility
<b>DRM</b>	Digital Reference Matrix
<b>EWS</b>	Early Warning System
<b>FENV</b>	Formatting Environment
<b>FISD</b>	Imbed Stack Frame
<b>FTBD</b>	Font Table
<b>FWRK</b>	Formatter Work Area
<b>GLY</b>	Galley Definition Block.
<b>GMBD</b>	General Markup Block
<b>GML</b>	Generalized Markup Language
<b>GOB</b>	Galley Object
<b>HRM</b>	Hyphenation Reference Matrix
<b>Info-Sys</b>	Information Systems
<b>LCBD</b>	Line Control Block
<b>MSHP</b>	Maintain System History Program
<b>PTF</b>	Program Temporary Fix
<b>PTR</b>	Pointer
<b>SAFD</b>	Save Frame DSECT
<b>SAVD</b>	Register Save Area
<b>SCBD</b>	Stack Control Block
<b>SCOM</b>	Common Data Area
<b>SCT</b>	Section Definition
<b>SEFI</b>	External Font Information Block
<b>SLDT</b>	Logical Device Table
<b>SMP</b>	System Modification Program
<b>SPDT</b>	Physical Device Table
<b>SSF</b>	Software Support Facility
<b>STRT</b>	Translate Table DSECT
<b>THDS</b>	Text Handling Data Structure
<b>THRC</b>	Text Handling Return Code
<b>TRTH</b>	Text Handling Data Structure
<b>WAD</b>	Writeable Alpha Dictionary



---

## **Appendix. Related Publications and Products**

## Publication Library Guide for the Document Composition Facility

The following table lists the Document Composition Facility publications by number as they relate to user tasks. "DCF Publications" on page 53 lists the titles and the order numbers that correspond to the numbers listed in the table.

Number	User Tasks	Typical Audience	Brief Description
(1) (2) (3) (20)	Planning and introducing DCF/DLF	Users, system planners	Provide a general overview of text processing, library facility, and available books.
(3) (4) (5) (12) (16) (20)	Formatting documents (using the GML starter set)	Novice to experienced end users	Provide an introduction to the Generalized Markup Language (GML) starter set and describes the GML starter set tags and SCRIPT/VS messages.
(6)	Creating bar codes with DCF/GML	Experienced end users	Provides information about using GML to create bar codes.
(9) (10) (11) (12) (17) (18) (19) (21)	Formatting documents (using SCRIPT/VS control words)	Knowledgeable to experienced end users	Describe the function and use of all SCRIPT/VS control words, macros, diagnostic aids, and the formatting features and messages.
(14) (15)	Converting RFTDCA for SCRIPT/VS formatting	Novice to experienced RFTDCA users	Describe the function and use of the optional Office Document Feature, including diagnostic aids and messages.
(4) (5) (7) (9) (10) (11) (19)	Modifying GML starter set <sup>4</sup>	Document administrator and text programmer <sup>5</sup>	Contain material about GML starter set tags, SCRIPT/VS control words, and GML starter set modifications.
(4) (5) (7) (8) (9) (10) (11) (16) (19)	Creating GML application processing functions	Document administrator and text programmer <sup>5</sup>	Provide information about designing your own GML and about GML concepts, GML starter set tags, SCRIPT/VS control words, and usage guidelines.
(10) (11) (12) (13) (19) (24) (25) (26) (27) (28) (29)	Installing, modifying, and maintaining DCF	System programmer	Give information on error isolation, program tailoring, and use of SCRIPT/VS.
(22)	Creating mathematical formulas with SMFF	Experienced end users	Describes the function and use of the SCRIPT Mathematical Formula Formatter (SMFF), including .EQ control word and messages.
(23)	Creating memos, transparencies, and schedules with GML applications	Novice to experienced end users	Describes the use of the memo, transparencies, and schedule applications, including messages.

<sup>4</sup> Central Programming Service support and maintenance are provided **only** on the unmodified GML applications shipped with DCF. If you modify any of these GML applications shipped with DCF, it is recommended that you also maintain an **unmodified** copy for diagnostic purposes.

<sup>5</sup> The document administrator is responsible for defining markup conventions and procedures for an organization. The text programmer implements application processing functions (APFs) that provide the processing specified by the document administrator.

## DCF Publications

Number	Titles and Order Numbers
(1)	<i>About DCF</i> , G520-6362.
(2)	<i>Document Composition Facility and Document Library Facility General Information</i> , GH20-9158.
(3)	<i>Document Composition Facility: Introduction to Generalized Markup Language</i> , G544-3192.
(4)	<i>Document Composition Facility: Generalized Markup Language Starter Set User's Guide</i> , SH20-9186.
(5)	<i>Document Composition Facility: Generalized Markup Language Starter Set Reference</i> , SH20-9187.
(6)	<i>Document Composition Facility: Bar Code User's Guide</i> , S544-3115.
(7)	<i>Document Composition Facility: Generalized Markup Language Starter Set Implementation Guide</i> , SH35-0050.
(9)	<i>Document Composition Facility: SCRIPT/VS User's Guide</i> , S544-3191.
(10)	<i>Document Composition Facility: SCRIPT/VS Text Programmer's Guide</i> , SH35-0069.
(11)	<i>Document Composition Facility: SCRIPT/VS Language Reference</i> , SH35-0070.
(12)	<i>Document Composition Facility: SCRIPT/VS Messages</i> , SH35-0048.
(13)	<i>Document Composition Facility: Diagnosis Guide and Reference</i> , LH40-0209.
(14)	<i>Document Composition Facility: Office Document Feature User's Guide</i> , G544-3129.
(15)	<i>Document Composition Facility: Office Document Feature Reference</i> , S544-3130.
(16)	<i>Using the Document Composition Facility</i> , SR21-0515 (Training Course 32291).
(17)	<i>Using DCF with the 4250 Printer</i> , SR20-8486 (Training Course 32908).
(18)	<i>Using DCF with Page Printers</i> , SR21-1211 (Training Course 32243).
(19)	<i>Document Composition Facility—Release 3 (SCRIPT/VS) for Document Administrators and Text Programmers</i> , SR20-7525 (Training Course).
(20)	<i>Document Composition Facility (SCRIPT/VS) Student Text</i> , SC20-1894 (Training Course).
(21)	<i>Document Composition Facility: TSO Enhancements Update Guide</i> , G544-3345.
(22)	<i>Document Composition Facility: SCRIPT Mathematical Formula Formatter User's Guide</i> , S544-3306.
(23)	<i>Document Composition Facility: Generalized Markup Language (GML) Applications User's Guide</i> , G544-3305.
(24)	<i>Document Composition Facility: MVS Program Directory</i> , G544-3669.
(25)	<i>Program Directory for use with DCF and SMFF for VM</i> , G544-3670.
(26)	<i>Document Composition Facility: VSE Program Directory</i> , G544-3671.
(27)	<i>Document Composition Facility: ODF Program Directory for MVS</i> , G544-3687.
(28)	<i>Document Composition Facility: ODF Program Directory for VM</i> , G544-3686.
(29)	<i>Document Composition Facility: SMFF Program Directory for MVS</i> , G544-3685.

The following are also available:

- *Document Composition Facility: GML Starter Set Quick Reference*, SX26-3723.
- *Document Composition Facility: SCRIPT/VS Text Programmer's Quick Reference*, SX26-3719.
- *Document Composition Facility Post-Processor Examples*, S544-3484.
- *Document Composition Facility*, SH35-0086 (binder).

**Note:** The DCF publications will be available at the end of the second quarter of 1991 as displayable BookManager built BOOKs and as source files on a CD-ROM, SK25-1980.

## Related Products

The following products are related to DCF. For information about ordering any of these products, contact your local IBM Branch Office.

- **MARKUP:** A PC-based editor that helps you create GML documents. Refer to *MARKUP User's Guide and Tutorial*, which you can order with the MARKUP product, 6476161.
- **Publishing Systems BookMaster:** A host-based application that runs under control of DCF and is designed for high-volume in-house publishing applications. This application contains a superset of the GML Starter Set.

- **Publishing Systems BookManager BUILD and BookManager READ:** Host-based IBM licensed programs that let you create and use online books and documents at your terminal in a VM/CMS system.
- **OfficeVision:** OfficeVision provides an integrated electronic office that delivers a broad range of office functions for business communications with the help of an integrated set of services.
- **Standard Generalized Mark-up Language (SGML) TextWrite OS/2 Edition:** TextWrite is a software program that writers can use to create and modify SGML-compliant documents.
- **TextTagger:** TextTagger is a software program that analyzes electronic documents and inserts tagging to comply with the Department of Defense's Computer-Aided Acquisition and Logistics Support (CALS) initiative.
- **SGML Translator DCF Edition:** A program that parses, validates, and translates SGML documents so they can be formatted and printed by DCF.
- **Print Services Facility (PSF):** A licensed program that combines print data with resources to manage and control data transmitted to IBM page printers.

---

## Optional Features

For information about ordering these optional features, contact your local IBM Branch Office:

- **SCRIPT Mathematical Formula Formatter (SMFF):** An optional feature of DCF that can be ordered separately. SMFF makes it possible to construct and display mathematical and scientific formulas on page printers.
- **Office Document Feature (ODF):** An optional feature of DCF that can be ordered separately. ODF allows documents that have been created with an office system to be printed on DCF-supported printers.

---

## Related Publications

You should use the following publications to evaluate the use of DCF in different operating environments:

- *IBM Virtual Machine/System Product: Introduction*, GC19-6200.

This publication contains an introduction to CMS (Conversational Monitor System), which is one of the interactive systems that SCRIPT/VS operates with.

- *IBM Virtual Machine/System Product: Terminal User's Guide*, GC19-6206.

This publication describes the various terminal types supported by VM/SP for those who plan to use VM/SP in their operations.

- *OS/VS2 TSO Terminal User's Guide*, GC28-0645.

This publication gives detailed user information about OS/VS2 TSO (Time Sharing Option), which is one of the interactive systems that SCRIPT/VS operates with. It describes the TSO EDIT facility and related facilities for text entry, text editing, and data set management.

- *A Guide to IBM's Advanced Function Printing*, G544-3095.

This publication describes the use of a licensed program (PSF, DCF, GML, OGL, GDDM, and PMF) and the use of a subset of a licensed program in conjunction with the IBM Advanced Function Printing (AFP) printers, including the IBM 3800 Printing Subsystem Models 3 and 6 and the IBM 3820 Page Printer.

- *Advanced Function Printing Software: General Information*, G544-3415.

This publication defines Advanced Function Printing (AFP), describes the features and functions of the AFP licensed programs, and shows how the programs work together. It is intended for the people in your organization who will plan for, install, use, and maintain IBM's AFP software products. It also contains a list of the AFP publications.

If you install DLF in the MVS environment, you need a copy of *OS/VS2 Access Method Services*, GC26-3841.

If you install DLF in the VSE environment, you need a copy of *VSE/VSAM Access Method Services: User's Guide and Reference*, SC24-5144.



## Index

### A

- abbreviations, list of 50
- ABEND keyword 10
- ABEND keyword procedure 10
- ABSOLUTE parameter ix
- APARs
  - documentation
  - requirements 21
  - submitting tapes or cartridges 21
  - supporting 20
  - initiating 20
  - preparing 20
- APF description 26

### B

- batch environments
  - running DCF in 30
  - running DLF in 30
- BookManager BUILD and READ 54
- BookMaster 53

### C

- changes with Release 4.0 ix
- CICS
  - as a programming requirement 30
- CICS with DCF 23
- CMS
  - as a programming requirement 30
- command usage errors, eliminating 6
- common entry and exit logic 49
- communication between modules 38
- component identification keyword 7
  - procedure 7
- component identification number 7
- control word
  - .IT 40
  - .MG 40
  - .SY 44
  - .ZZ 43
- control word description 25
- courses
  - self-study 53
  - training 53

### D

- DCF
  - Foreground Environment Feature of 30
  - hardware requirements
    - printers 31
    - processors 30

### DCF (continued)

- hardware requirements (continued)
  - storage devices 32
  - terminals 31
- in a batch environment 30
- in an interactive environment 30
- storage requirements and estimates
  - direct-access storage 32
  - real storage 32
- DCF load modules 34
- DCF program overview 23
  - general description 23
  - hardware requirements
    - printers 31
    - processors 30
    - storage devices 32
    - storage requirements and estimates 32
    - terminals 31
  - host system
    - LIB option description 25
    - PROFILE option description 25
    - SEARCH option description 25
  - input files 25
  - operating environment
    - programming systems 30
  - SCRIPT/VS program control flow 27
  - structural overview
    - communication between modules 38
    - DCF load modules 34
    - DCF source modules 35
    - program conventions 37
    - program structure 34
  - types of input 25
    - APF description 26
    - control words description 25
    - GML tag description 26
    - macro description 26
    - symbol description 26
  - with CICS 23
  - with FLIP 23
  - with GML 23
  - with SCRIPT/VS 23
- DCF publications 53
  - library guide 52
- DCF source modules 35
- developing the keyword string 6
- diagnosis and correction process personnel
  - diagnostician 3
  - IBM change team 4
  - IBM specialist 4
  - IBM support center representative 3

diagnosis information, gathering of 7  
 diagnostician, description of 3  
 direct-access storage 32  
 DLF  
   in a batch environment 30  
 DOC keyword 12  
 DOC keyword procedure 12  
 Document Composition Facility  
   storage requirements and estimates  
   virtual storage 33

## E

eliminating command usage errors 6  
 eliminating errors, command usage 6  
 errors, command usage, elimination of 6  
 extended architecture for VM and MVS ix

## F

failure conditions, types of 8  
 FE patch areas 42  
 features, optional  
   Office Document Feature (ODF) 54  
   SCRIPT Mathematical Formula Formatter  
     (SMFF) 54  
 file size limitation, removal of ix  
 FLIP enhancement x  
 FLIP with DCF 23  
 font character set size limitation, removal of ix  
 font library index program (FLIP) enhancement x  
 foreground environment feature 30  
 function keyword 14  
   procedure 14

## G

gathering diagnosis information 7  
 GML tag description 26  
 GML with DCF 23

## H

hardware requirements  
   printers 31  
   processors 30  
   storage devices 32  
   terminals 31  
 hyphenation ix

## I

IBM change team, description of 4  
 IBM LaserPrinter 4028 support ix  
 IBM specialist, description of 4  
 IBM support center representative, description of 3

image object content architecture (IOCA) ix  
 including overlays ix  
 INCORROUT keyword 11  
 INCORROUT keyword procedure 12  
 interactive environments  
   using DCF in 30  
 internal program service aids 42  
 internal programming errors 42

## K

keyword  
   ABEND 10  
   component identification 7  
   DOC 12  
   function 14  
   INCORROUT 11  
   LOOP 11  
   maintenance level 18  
   module 14  
   MSGx 8  
   PERFM 13  
   release level 17  
   subfunction 16  
   type-of-failure 8  
   WAIT 10

keyword procedures  
   ABEND 10  
   component identification 7  
   DOC 12  
   function 14  
   INCORROUT 12  
   LOOP 11  
   maintenance level 18  
   module 14  
   MSGx 9  
   PERFM 13  
   release level 17  
   subfunction 16  
   WAIT 10

keyword string  
   description of 6  
   developing the 6

keywords  
   description of 5  
   using 5

## L

language attribute on the :GDOC tag ix  
 LIB option description 25  
 library guide  
   for DCF publications 52  
 list of abbreviations 50  
 LOOP keyword 11



LOOP keyword procedure 11

## M

macro description 26  
maintenance level keyword 18  
    procedure 18  
MARKUP 53  
memo tags x  
MESSAGE (DELAY) 39  
messages and codes 41  
MESSAGE(ID) option 39  
MESSAGE(TRACE) option 39  
module call stack 48  
module identification 48  
module keyword 14  
    procedure 14  
module renaming x  
modules, communication between 38  
MSGx keyword 8  
MSGx keyword procedure 9  
MVS  
    as a programming requirement 30  
MVS and VM extended architecture ix

## N

NOSPIE option 39  
NUMBER option 39

## O

ODF 54  
Office Document Feature (ODF) 54  
OfficeVision 54  
online help ix  
online help for error messages ix  
operating environment 30  
operating environments 30  
    hardware requirements  
        printers 31  
        processors 30  
        storage devices 32  
        terminals 31  
optional features  
    Office Document Feature (ODF) 54  
    SCRIPT Mathematical Formula Formatter (SMFF) 54  
options  
    MESSAGE (DELAY) 39  
    MESSAGE(ID) 39  
    MESSAGE(TRACE) 39  
    NOSPIE 39  
    NUMBER 39  
    PAGE 39  
    PSOUT 39

options (*continued*)

    UNFORMAT 39  
    output comment control word x  
    overhead transparency tags x

## P

page number limit x  
PAGE option 39  
page segment enhancements ix  
PERFM keyword 13  
PERFM keyword procedure 13  
prerequisite knowledge 2  
PRINT option ix  
Print Services Facility (PSF) 54  
printers 31  
procedure, search argument 19  
processors 30  
products, related  
    *See related products*  
PROFILE option description 25  
program control flow for SCRIPT/VS 27  
program conventions 37  
    module naming conventions 37  
    register conventions 37  
program overview, DCF 23  
program structure 34  
programming systems 30  
PSF 54  
PSOUT option 39  
publications  
    for DCF 53  
publication, related (*see related publications*)

## R

real storage 32  
related products  
    BookManager BUILD and READ 54  
    BookMaster 53  
    MARKUP 53  
    OfficeVision 54  
    Print Services Facility (PSF) 54  
    SGML TextWrite 54  
    SGML Translator DCF edition 54  
    TextTagger ProcessMaster edition 54  
    TextTagger Workstation edition 54  
related publications 51  
    descriptions 55  
release level keyword 17  
    procedure 17  
Release 4.0 changes  
    font library index program (FLIP) enhancement x  
    hyphenation ix  
    IBM LaserPrinter 4028 support ix  
    including overlays ix

Release 4.0 changes (*continued*)

- language attribute on the :GDOC tag ix
- memo tags x
- module renaming x
- MVS and VM extended architecture ix
- online help ix
- online help for error messages ix
- output comment control word x
- overhead transparency tags x
- page number limit x
- page segment enhancements
  - ABSOLUTE parameter ix
  - image object content architecture (IOCA) ix
- PRINT option ix
- removal of file size limitation ix
- removal of font character set size limitation ix
- revision character alignment x
- revision code font ix
- schedule tags x
- separation masters ix
- shading ix
- table storage relief x
- TSO changes x
- removal of file size limitation ix
- removal of font character set size limitation ix
- revision character alignment x
- revision code font ix

## S

- schedule tags x
- SCRIPT command options 39
  - MESSAGE (DELAY) 39
  - MESSAGE(ID) 39
  - MESSAGE(TRACE) 39
  - NOSPIE 39
  - NUMBER 39
  - PAGE 39
  - PSOUT 39
  - UNFORMAT 39
- SCRIPT Mathematical Formula Formatter (SMFF)
  - program 54
- SCRIPT/VS
  - APF description 26
  - control word description 25
  - GML tag description 26
  - input files 25
  - LIB option description 25
  - macro description 26
  - PROFILE option description 25
  - program control flow 27
  - SEARCH option description 25
  - symbol description 26
- SCRIPT/VS with DCF 23
- search argument
  - procedure 19

search argument (*continued*)

- technique 19
- SEARCH option description 25
- self-study courses 53
- separation masters ix
- service aids 39
  - internal program service aids 42
    - common entry and exit logic 49
    - FE patch areas 42
    - internal programming errors 42
    - module call stack 48
    - module identification 48
    - system tools 49
    - .SY control word 44
    - .ZZ control word 43
- text input service aids 39
  - messages and codes 41
  - SCRIPT command options 39
  - terminal-oriented control words 40
  - .IT control word 40
  - .MG control word 40
- SGML TextWrite 54
- SGML Translator DCF edition 54
- shading ix
- SMFF 54
- storage devices 32
- storage requirements and estimates 32
  - direct-access storage 32
  - real storage 32
  - virtual storage 33
- storage, direct-access 32
- storage, real 32
- storage, virtual 33
- structural overview 34
- subfunction keyword 16
  - procedure 16
- Summary of Amendments ix
- symbol description 26
- system tools 49

## T

- table storage relief x
- tag set for memos x
- tag set for overhead transparencies x
- tag set for schedules x
- terminal-oriented control words 40
- terminals 31
- TextTagger ProcessMaster edition 54
- TextTagger Workstation edition 54
- training courses 53
- TSO
  - as a programming requirement 30
- TSO changes x
- type-of-failure conditions, description of 8

type-of-failure keyword

ABEND 10

DOC 12

INCORROUT 11

LOOP 11

MSGx 8

PERFM 13

procedure 9, 10, 11, 12, 13

ABEND 10

DOC 12

INCORROUT 12

LOOP 11

MSGx 9

PERFM 13

WAIT 10

WAIT 10

type-of-failure keyword. 8

types of input with DCF 25

## U

UNFORMAT option 39

using keywords 5

## V

virtual storage 33

working storage 33

VM and MVS extended architecture ix

VSAM

as a programming requirement 30

VSE/VSAM 30

VSE

as a programming requirement 30

VSE/Advanced Functions 30

VSE/VSAM 30

## W

WAIT keyword 10

WAIT keyword procedure 10

## Special Characters

.IT control word 40

.MG control word 40

.SY control word 44

.ZZ control word 43



## Readers' Comments

## Document Composition Facility: Diagnosis Guide and Reference

**Publication No. LH40-0209-00**

Use this form to provide comments about this publication, its organization, or subject matter. Understand that IBM may use the information any way it believes appropriate, without incurring any obligation to you. Your comments will be sent to the author's department for the appropriate action. Comments may be written in your language.

**Note:** IBM publications are not stocked at the location to which this form is addressed. Direct requests for publications or for assistance in using your IBM system, to your IBM representative or local IBM branch office.

- |   | Yes                      | No                       |
|---|--------------------------|--------------------------|
| • Does the publication meet your needs? | <input type="checkbox"/> | <input type="checkbox"/> |
| • Did you find the information:         |                          |                          |
| Accurate?                               | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to read and understand?            | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to retrieve?                       | <input type="checkbox"/> | <input type="checkbox"/> |
| Organized for convenient use?           | <input type="checkbox"/> | <input type="checkbox"/> |
| Legible?                                | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete?                               | <input type="checkbox"/> | <input type="checkbox"/> |
| Well illustrated?                       | <input type="checkbox"/> | <input type="checkbox"/> |
| Written for your technical level?       | <input type="checkbox"/> | <input type="checkbox"/> |
| • Do you use this publication:          |                          |                          |
| As an introduction to the subject?      | <input type="checkbox"/> | <input type="checkbox"/> |
| As a reference manual?                  | <input type="checkbox"/> | <input type="checkbox"/> |
| As an instructor in class?              | <input type="checkbox"/> | <input type="checkbox"/> |
| As a student in class?                  | <input type="checkbox"/> | <input type="checkbox"/> |
| • What is your occupation?              |                          |                          |

**Thank you for your input and cooperation.**

**Note:** If mailed in the U.S.A., no postage stamp is necessary. For residents outside the U.S.A., your local IBM office or representative will forward your comments.

**Comments:**This image shows a single sheet of white paper with horizontal black ruling lines. The lines are evenly spaced and run across the width of the page. There are approximately 20 lines visible. The paper appears to be a standard notebook or legal pad style. The edges of the paper are slightly irregular, suggesting it might be a scan of a physical document. There is no handwriting or other markings on the page.

Name \_\_\_\_\_

Address

Company or Organization

Phone No. \_\_\_\_\_

Readers' Comments  
LH40-0209-00



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES



## BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Information Development  
Department 588  
P.O. Box 1900  
Boulder, CO 80301-9191



Fold and Tape

Please do not staple

Fold and Tape

LH40-0209-00

## Readers' Comments

## Document Composition Facility: Diagnosis Guide and Reference

**Publication No. LH40-0209-00**

Use this form to provide comments about this publication, its organization, or subject matter. Understand that IBM may use the information any way it believes appropriate, without incurring any obligation to you. Your comments will be sent to the author's department for the appropriate action. Comments may be written in your language.

**Note:** IBM publications are not stocked at the location to which this form is addressed. Direct requests for publications or for assistance in using your IBM system, to your IBM representative or local IBM branch office.

- |   | Yes                      | No                       |
|---|--------------------------|--------------------------|
| • Does the publication meet your needs? | <input type="checkbox"/> | <input type="checkbox"/> |
| • Did you find the information:         |                          |                          |
| Accurate?                               | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to read and understand?            | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to retrieve?                       | <input type="checkbox"/> | <input type="checkbox"/> |
| Organized for convenient use?           | <input type="checkbox"/> | <input type="checkbox"/> |
| Legible?                                | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete?                               | <input type="checkbox"/> | <input type="checkbox"/> |
| Well illustrated?                       | <input type="checkbox"/> | <input type="checkbox"/> |
| Written for your technical level?       | <input type="checkbox"/> | <input type="checkbox"/> |
| • Do you use this publication:          |                          |                          |
| As an introduction to the subject?      | <input type="checkbox"/> | <input type="checkbox"/> |
| As a reference manual?                  | <input type="checkbox"/> | <input type="checkbox"/> |
| As an instructor in class?              | <input type="checkbox"/> | <input type="checkbox"/> |
| As a student in class?                  | <input type="checkbox"/> | <input type="checkbox"/> |
| • What is your occupation?              |                          |                          |

**Thank you for your input and cooperation.**

**Note:** If mailed in the U.S.A., no postage stamp is necessary. For residents outside the U.S.A., your local IBM office or representative will forward your comments.

**Comments:**This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.

Name \_\_\_\_\_

**Address**

**Company or Organization**

**Phone No.**

Readers' Comments  
LH40-0209-00



Fold and Tape

Please do not staple

Fold and Tape



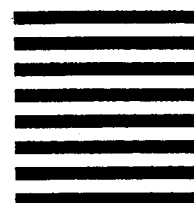
## BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Information Development  
Department 588  
P.O. Box 1900  
Boulder, CO 80301-9191

NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES



Fold and Tape

Please do not staple

Fold and Tape



## Readers' Comments

## Document Composition Facility: Diagnosis Guide and Reference

**Publication No. LH40-0209-00**

Use this form to provide comments about this publication, its organization, or subject matter. Understand that IBM may use the information any way it believes appropriate, without incurring any obligation to you. Your comments will be sent to the author's department for the appropriate action. Comments may be written in your language.

**Note:** IBM publications are not stocked at the location to which this form is addressed. Direct requests for publications or for assistance in using your IBM system, to your IBM representative or local IBM branch office.

- |   | Yes                      | No                       |
|---|--------------------------|--------------------------|
| • Does the publication meet your needs? | <input type="checkbox"/> | <input type="checkbox"/> |
| • Did you find the information:         |                          |                          |
| Accurate?                               | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to read and understand?            | <input type="checkbox"/> | <input type="checkbox"/> |
| Easy to retrieve?                       | <input type="checkbox"/> | <input type="checkbox"/> |
| Organized for convenient use?           | <input type="checkbox"/> | <input type="checkbox"/> |
| Legible?                                | <input type="checkbox"/> | <input type="checkbox"/> |
| Complete?                               | <input type="checkbox"/> | <input type="checkbox"/> |
| Well illustrated?                       | <input type="checkbox"/> | <input type="checkbox"/> |
| Written for your technical level?       | <input type="checkbox"/> | <input type="checkbox"/> |
| • Do you use this publication:          |                          |                          |
| As an introduction to the subject?      | <input type="checkbox"/> | <input type="checkbox"/> |
| As a reference manual?                  | <input type="checkbox"/> | <input type="checkbox"/> |
| As an instructor in class?              | <input type="checkbox"/> | <input type="checkbox"/> |
| As a student in class?                  | <input type="checkbox"/> | <input type="checkbox"/> |
| • What is your occupation?              |                          |                          |

**Thank you for your input and cooperation.**

**Note:** If mailed in the U.S.A., no postage stamp is necessary. For residents outside the U.S.A., your local IBM office or representative will forward your comments.

**Comments:**This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There is no handwriting or other markings on the paper.

Name \_\_\_\_\_

**Address**

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

## BUSINESS REPLY MAIL

FIRST CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Information Development  
Department 588  
P.O. Box 1900  
Boulder, CO 80301-9191



Fold and Tape

Please do not staple

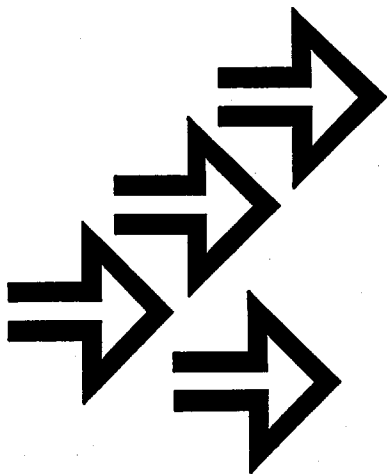
Fold and Tape





File Number: S370-37  
Program Number: 5748-XX9

Licensed Materials — Property of IBM



Printed in U.S.A.

LH40-0209-00

