IBM

VS FORTRAN Compiler and Library Diagnosis Guide

Program Product

```
IRR(I,J) = 1
IRI(I,J) = 2
CONTINUE
PRINT  20, (
1  I = 1, 3)
FORMAT (3(1
STOP
END
```

Program Numbers:

5748-FO3 Compiler and Library
5748-LM3 Library Only

Release 4.0

IBM

VS FORTRAN Compiler and Library Diagnosis Guide

Program Numbers
5748-FO3 (Compiler and Library)
5748-LM3 (Library Only)
Release 4.0

SC26-3990-4

Fifth Edition (October 1984)

This is a major revision of, and makes obsolete, SC26-3990-3.

This edition applies to Release 4.0 of VS FORTRAN, Program Product 5748-F03 (Compiler and Library),5748-LM3 (Library Only), and to any subsequent releases until otherwise indicated in new editions or technical newsletters.

The changes for this edition are summarized under "Summary of Amendments" following the preface. Specific changes are indicated by a vertical bar to the left of the change. These bars will be deleted at any subsequent republication of the page affected. Editorial changes that have no technical significance are not noted.

Changes are made periodically to this publication; before using this publication in connection with the operation of IBM systems, consult the latest IBM System/370 and 4300 Processors Bibliography, GC20-0001, for the editions that are applicable and current.

PREFACE

This book presents a systematic way of selecting keywords to describe a suspected failure in the VS FORTRAN Compiler or Library. It assumes that you have:

1. Used all debug tools and options available to you, such as static debug statements, or the VS FORTRAN Interactive Debug product.

2. Examined all VS FORTRAN messages and error conditions produced by the program, and corrected them when possible.

3. Ensured, as far as possible, that the error is occurring in the VS FORTRAN Compiler or Library and not in the application program.

4. Noted the specific sequence of events preceding the error condition.

HOW TO USE THIS BOOK

If you suspect a product failure, follow the instructions for keyword selection in "Developing a Keyword String" on page 5.

Then use the keywords to search an indexed data base to determine whether your problem has been previously described in an APAR (authorized program analysis report). You may do the search yourself if you have access to a suitable data base such as Info Access or Early Warning System, or you can call the IBM Support Center to search the Software Support Facility. "Using the Keyword String as a Search Argument" on page 15 describes the search procedure.

If no APAR is identified for your product failure, you may be asked to submit one so that the product can be corrected if necessary. "Preparing an APAR" on page 17 describes this procedure.

VS FORTRAN PUBLICATIONS

The following VS FORTRAN publications also contain helpful diagnostic information.

- VS FORTRAN Programming Guide, SC26-4118

- VS FORTRAN Language and Library Reference, GC26-4119

- VS FORTRAN Interactive Debug Guide and Reference, SC26-4116

RELATED PUBLICATIONS

Publications that contain diagnostic information relevant to VS FORTRAN are:

- _VSE/Advanced Functions Messages_, SC33-6098
- _OS/VS Message Library: VS2 System Messages_, GC38-1002
- _OS/VS2 MVS Supervisor Services and Macro Instructions_, GC28-0683
- _VM/SP Messages and Codes_, SC19-6204
- _MVS/Extended Architecture Message Library: System Messages_, GC28-1128
- _Field Engineering Programming Systems General Information_, GC29-2228

SUMMARY OF AMENDMENTS

RELEASE 4.0, OCTOBER 1984

PROGRAMMING ENHANCEMENTS

- The following compiler options have been added:
  - AUTODBL
  - RENT or NORENT
  - SDUMP
  - SXM or NOSXM
- VS FORTRAN programs can now load and access VSAM KSDS files.
- The following compiler statements have been added:
  - REWRITE
  - DELETE
- The library has been restructured to allow more execution-time loading of library routines. You can choose to have all library routines (other than the mathematical routines) either link-edited into the load module with compiler-generated code, or loaded dynamically at execution time.

SERVICE CHANGES

- This manual contains only "Guide" information on selecting and using keyword strings.
- "VS FORTRAN Statements" has been moved from the Appendix to the section on procedure under "Type of Failure Keyword."
- "Service Aids" is now an appendix.

RELEASE 3.1, MARCH 1984

PROGRAMMING ENHANCEMENTS

- VS FORTRAN Interactive Debug is now supported. When a VS FORTRAN program is executed, the user has a choice of two different execution options:
  - DEBUG, which activates VS FORTRAN Interactive Debug immediately; and
  - NODEBUG, the IBM default, which does not invoke VS FORTRAN Interactive Debug.

Note: The TEST compiler option is not necessary for VS FORTRAN Interactive Debug.

RELEASE 3.0, MARCH 1983

PROGRAMMING ENHANCEMENTS

- The following compiler options have been added:
  - CHARLEN
  - CI
  - NOSDUMP
  - SRCFLG or NOSRCFLG
  - SYM or NOSYM
  - TRMFLG or NOTRMFLG
- The compiler option SC has been discontinued.
- The debugging aid Symbolic Dump has been added for user errors.

SERVICE CHANGES

- A new keyword, ABENDU, is added to the Abnormal Termination procedure for user abends.
- The compiler modules have been reorganized into a directory format.
- The discussion of the main data area and control blocks have been expanded.

CONTENTS

FIGURES

INTRODUCTION TO GUIDE INFORMATION

VS FORTRAN Compiler and Library product failures can be described through the use of keywords. A keyword is a word or abbreviation used to describe a single aspect of a product failure. A set of keywords for a given problem is called a keyword string. The keyword string is used as a search argument in an IBM software support data base, such as the Software Support Facility (SSF) or the Early Warning System (EWS). To describe a product failure so you can search a data base, you must develop a set of keywords.

When the problem is described in the software support data base with the same set of keywords, the search yields matching descriptions of the problem and a correction is usually known.

If the problem is not listed, use the keywords to help prepare an APAR. Keywords are intended to ensure that any two people will identically describe the same type of problem caused by the same program error. For additional information on keywords and APAR preparation, see _Field Engineering Programming System General Information_.

KEYWORD USAGE

The first keyword identifies the product by its **Component Identification Number**. A search of a software support data base with this keyword alone would detect all reported problems for the entire program product. Each keyword added makes the search argument more specific, thereby reducing the types of problem descriptions to those shown in Figure 1 on page 2.

The procedures contained in the rest of this Guide will help you develop a full keyword string.

- "Component Identification Keyword" on page 5 lists the program product number used to identify VS FORTRAN products.
- "Type of Failure Keyword" on page 5 tells how to specify different types of failures.
- "Release and Modification Levels" on page 14 describes how to identify the release levels of various VS FORTRAN products.
- "Using the Keyword String as a Search Argument" on page 15 explains how the keyword string is used to search a software support data base.
- "Preparing an APAR" on page 17 explains what materials are needed to prepare an authorized program analysis report (APAR).

Figure 1 on page 2 illustrates the process of creating a keyword string.

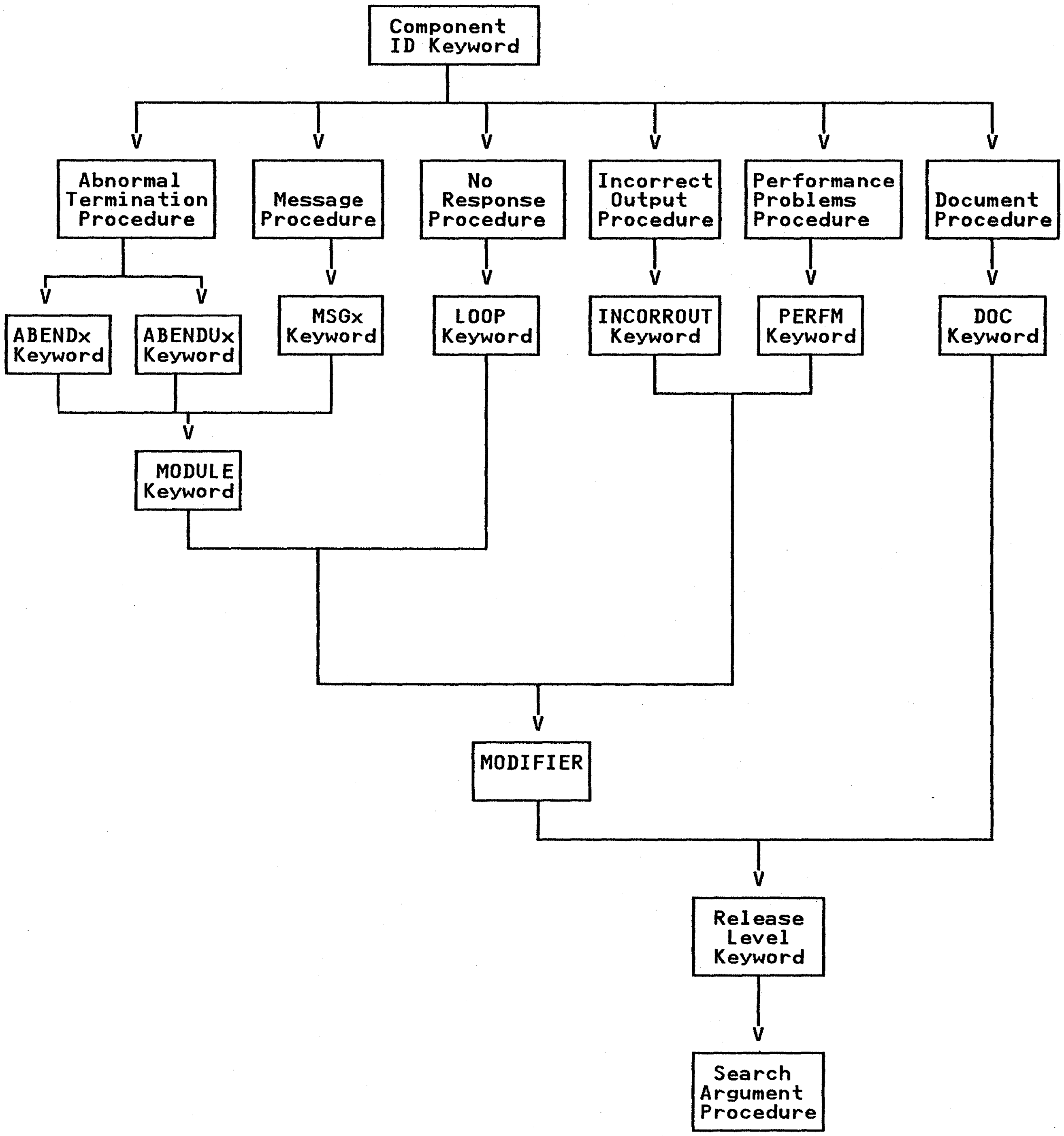

Figure 1. Flowchart of a VS FORTRAN Keyword String

DIAGNOSIS PROCEDURE

Use these procedures to gather the diagnostic information needed to develop keyword search arguments for a software support data base.

1. Correct all problems diagnosed by VS FORTRAN messages and ensure that any messages previously generated have nothing to do with the problem being worked on. VS FORTRAN Compiler messages have the prefix IFX; VS FORTRAN Library messages have the prefix IFY.

2. If your program has been compiled at optimization level 0 with the NORENT option, use VS FORTRAN Interactive Debug or FORTRAN static debug statements to identify a problem that occurs during run time. If you have the VS FORTRAN Interactive Debug product, you can use it for execution-time analysis of your program and its error. FORTRAN static debug can isolate the problem and determine whether it is a user error, a compiler-produced code error, or an execution-time library error.

   For more information on VS FORTRAN Interactive Debug, see "VS FORTRAN Interactive Debug" on page 18. An overview of the static debug statements and options is given in Appendix A, "Service Aids" on page 18.

3. If the compilation fails at an optimization level other than 0, try to recompile the program at a lower level. If you are successful, compile the program at that level to bypass the problem until a fix is found and applied. Remember to note the conditions and options in effect when the failure occurred.

4. Determine if the program has been changed since it last compiled or executed successfully. If it has, examine the changes. If the error is occurring in the changed code and cannot be corrected, note the change that caused the error. If possible, retain copies of both the original and the changed programs.

5. After the failure has been explored and identified, consider writing a small test case that re-creates the problem. This test case should help you to:

   - Distinguish between an error in the application program and an error in VS FORTRAN
   - Choose keywords that best describe the error
   - Pinpoint the problem

6. If the problem occurs during execution, specify the following options in the compilation, in addition to the options originally specified, to produce maximum diagnostic information.

   LIST

   GOSTMT

   MAP

   SDUMP

   SOURCE

   SRCFLG

   XREF

   If the compiler abends, see "Compiler Trace Option" on page 19.

7. Note the sequence of events that lead to the error condition. This information may be useful in developing a set of keywords, and will be needed if an APAR is required.

8. If the user program is abending, direct the object error unit (usually unit 6) to a file, not a terminal, to allow the post-abend dump to be printed.

9. Begin developing the set of keywords, using the procedure in "Component Identification Keyword" on page 5.

DEVELOPING A KEYWORD STRING

The following sections will help you select the correct keywords to describe a product failure and build a keyword string.

COMPONENT IDENTIFICATION KEYWORD

The component identification number is the first keyword in the keyword string. It indicates the library within the software support data base that contains known problem descriptions for the product.

5748F0300 is the component identification keyword for the VS FORTRAN Compiler and Library. **5748LM300** is the component identification keyword for the VS FORTRAN Library only.

Continue the diagnosis procedure with "Type of Failure Keyword."

TYPE OF FAILURE KEYWORD

From the list in Figure 2, select the symptom that best describes the problem. Then turn to the page referenced for that type of problem. If more than one keyword describes the problem, use the one that appears first in the list.

Symptom	**Description**	**Reference**
Abnormal Termination	A program exception has occurred, or the program has terminated abnormally.	"Abnormal Termination Problems" on page 6
Message Problems	A message indicates an internal program error, or seems itself to be an error.	"Message Problems" on page 10
No Response	The program seems not to be doing anything, or is doing something repetitively.	"No Response Problems" on page 11
Documentation Problems	The information in one of the VS FORTRAN publications is incorrect or missing.	"Documentation Problems" on page 11
Incorrect Output	The output from the program is missing or invalid.	"Incorrect Output Problems" on page 12
Slow Response	The performance of the program is degraded.	"Performance Problems" on page 13

Figure 2. VS FORTRAN Symptom Table

ABNORMAL TERMINATION PROBLEMS

Use the ABENDx keyword procedure when a program exception occurs:

- Within VS FORTRAN
- Within a VS FORTRAN-compiled program
- Whenever a VS FORTRAN or a VS FORTRAN-compiled program terminates without a message

Do not use this keyword if termination was forced because too much time was spent in a wait state or in an endless loop. For those situations, refer to the procedure for "No Response Problems" on page 11.

Procedure

1. Determine whether the problem occurs during compile time or during execution time.

   - If the problem occurs during the compilation of your program, go to step 2 on page 7.
   - If the problem occurs during execution of your program, there is a strong probability that it is a user error. Before continuing, verify that the problem is not the result of incorrect coding.

     Execution-time errors that cause the program to terminate abnormally produce message IFY240I and a traceback map.

     The IFY240I message has the following format:

     ```
     IFY240I VSTAE:  ABEND CODE IS:  SYSTEM SSSS, USER UUUU,
             SCB/SDWA=HHHHHHHH.
     ```

     a. In the ABENDx keyword, replace the x with the abend code, which is the SSSS coding shown inside the IFY240I message.

        VS FORTRAN does not issue user ABENDs during program execution. If you receive a message that indicates a user ABEND, contact your system programmer to determine the problem in your code.

     b. Use message IFY240I as a modifier in the keyword string. If you received an additional message besides IFY240I, also include that message as a modifier. For further details on the IFY240I message, see VS FORTRAN Language and Library Reference.

     c. Study the traceback map to discover which source statement is causing the problem. For a complete description of how to request and interpret a traceback map, see VS FORTRAN Programming Guide. If you are unable to determine which statement caused the failure, continue with "Release and Modification Levels" on page 14.

d. Use the following statement names as modifier keywords to describe the statement you think is creating the error.

ASSIGN	GENERIC[1]
AT	GOTO[3]
BACKSPACE	IF[4]
BLOCKDATA	IF THEN[5]
CALL	IMPLICIT
CHARACTER	INCLUDE
CLOSE	INQUIRE
COMMON	INTEGER
COMPLEX	INTRINSIC
CONTINUE	LOGICAL
DATA	NAMELIST
DEBUG	OPEN
DEFINE FILE[1]	PARAMETER
DELETE	PAUSE
DIMENSION	PRINT
DISPLAY	PROGRAM
DO	PUNCH[1]
DOUBLE PRECISION	READ[6]
EJECT	REAL
ELSE[2]	RETURN
ELSE IF[2]	REWIND
END	REWRITE
ENDFILE	SAVE
END IF[2]	STOP
ENTRY	SUBROUTINE
EQUIVALENCE	TRACE OFF
EXTERNAL	TRACE ON
FIND[1]	WAIT
FORMAT	WRITE[6]
FUNCTION	

[1] LANGLVL(66) only.

[2] Use IF THEN as keyword.

[3] Includes assigned GOTO, computed GOTO, and unconditional GOTO.

[4] Includes arithmetic IF and logical IF.

[5] Includes block IF, ELSE, ELSE IF, and END IF.

[6] Includes asynchronous, with list-directed I/O, with NAMELIST, with internal files, formatted or unformatted with direct or keyed access, or formatted or unformatted with sequential access.

A sample keyword string showing an abend in the library displaying two messages might look like this:

```
  Component Identification:  5748F0300
  Type of Failure:           ABEND0C1
  Modifiers:                 IFY240I
                             IFY210I
```

e. Determine at what levels of optimization the failure occurs.

   If it occurs at only some levels, indicate in the keyword string the levels of optimization at which it does occur. Select the appropriate modifier keyword from the list shown in Appendix B, "Compiler Options" on page 20.

f. Continue with "Release and Modification Levels" on page 14.

2. Use ABENDU0016 as your type of failure keyword, because all abnormal terminations in the compiler receive this user abend.

3. Recompile the program to produce a trace point 77 dump if the VS FORTRAN compiler terminates abnormally.

   For information on how to create a trace point 77 dump, refer to "Compiler Trace Option" on page 19.

4. Use the trace point 77 dump to determine the module that failed, the location within the module at which the problem occurred, and the type of error exception.

   Figure 3 on page 9 shows a sample trace point 77 dump. Areas discussed in the following procedure are in boldface and underscored.

5. Locate the program status word (PSW) in the first line under the heading of the trace point 77 dump. The last position of the first word in the PSW is an identifier which stands for the exception type.

   Select the identifier to use for the exception type in your keyword from the following table:

Identifier	**Exception Type**	**Code**
1	operation	0C1
4	protection	0C4
5	addressing	0C5
6	specification	0C6
7	data	0C7
9	fixed-point divide	0C9
C	exponent overflow	0CC
D	exponent underflow	0CD
F	floating-point divide	0CF

   Now replace the x of ABEND0Cx with the identifier to make a keyword.

   For example, if you suspect that the compiler caused the abend and have the trace point 77 shown in Figure 3, you would develop the following keyword string:

Component Identification:	5748F0300
Type of Failure:	ABENDU0016
Exception Code Modifier:	ABEND0C1

6. Use the trace point 77 dump to develop two modifier keywords from the module name and from the displacement in the module at which the error occurred. Both of these modifiers can be found in the dump, on the line following the general registers. In the example given in Figure 3, the module name is IFX2STAL and the displacement is 0058.

   Using the example shown in Figure 3, your set of keywords would look like this:

Component Identification:	5748F0300
Type of Failure:	ABENDU0016
Exception Code Modifier:	ABEND0C1
Module Modifier:	IFX2STAL
Displacement Modifier:	0058

7. Determine at what levels of optimization the failure occurs.

   If it occurs at only some levels, indicate in the keyword string the levels of optimization at which it does occur. Select the appropriate modifier keyword from the list shown in Appendix B, "Compiler Options" on page 20.

8. Continue with "Release and Modification Levels" on page 14.

```
PSW AND GENERAL REGISTERS.   PHASE 2.   TRACE POINT 77.   PROGRAM  CHECK.
PSW:    FFE40001  4208B53A

GR 0-7:000F1B99  0008F978  00000000  00000000  0011F9E0  00000000  00090463  0008F464
GR 8-F:0008E465  0008D466  0008C467  000F1C00  0008B468  000F2180  4208B538  00000000

AREA AROUND INTERRUPT.
(0008B510) B2BC5040 D3685840 B2C05040 D36C5840  B2C45040 D3709101 B07B4770 C0D458F0
(0008B530) 76D04110 751405EF 00000000 58F076D0  4110751C 05EF5840 B1BC5040 B4F89640
(0008B550) B4C65840 B1BC5040 D198D201 B358762E  1F445040 D18C4340 76FF4240 D1CA4140

THE PROGRAM CHECK OCCURED AT DISPLACEMENT (00D2) IN IFX2STAL.

SAVE AREA CONTENTS: UNUSED, BKPTR, FWDPTR, GPR(14-12), CALLER ID, WORKING STORAGE.
SAVE AREA FOR IFX2STAL (000F2180).  CALLED BY IFX2CNTL AT DISPLACEMENT (0496).
(00000000) 00000000 000F2120 000F24F8 0036C3EA  0202724C 00000004 003FC808 00000000
(00000020) 00000000 0011F9E0 00000000 00090463  0008F464 0008E465 0008D466 0008C467
(00000040) 000F1C00 0008B468 C3D5E3D3 0002C54F  000F2178 001075F0 0002B527 000F1C00
                                •
                                •
                                •
(00000340) 00000026 00000000 00000000 00000000  0011F680 00000000 00000000 00000000
(00000360) 0011F878 00000000 0011F908 00000000  0011F9E0 00000000
SAVE AREA FOR IFX2CNTL (000F2120).  CALLED BY IFX0CNTL AT DISPLACEMENT (0706).
(00000000) 00000000 000F1930 000F2180 4202C48E  0008B468 000F1B99 000F216C 000F217D
(00000020) 00000000 00000000 00000000 00000000  00000000 00000001 000F1930 00000000
(00000040) 000F1C00 0002BFF8 C3D5E3D3 0002C54F  000F2178 00000000 0000002C 00000000
```

Figure 3. Sample Trace Point 77 Output

MESSAGE PROBLEMS

Use the MSGx keyword procedure for any of these conditions:

- A message is issued indicating an internal program error.
- A message is issued under conditions that should not have caused it to be issued.
- A message contains invalid data or is missing data.

Do not use this procedure if you received a message as the result of an abnormal termination. In that case, see "Abnormal Termination Problems" on page 6.

Each VS FORTRAN compiler message is identified by a string of eight characters, followed by a string of four characters:

```
  IFXpnnnn mmmm
```

where:

IFX	is the message prefix identifying all VS FORTRAN compiler messages.
p	is the number of the compiler phase issuing the message.
nnnn	is the message number.
mmmm	represents the last four characters of the name of the module issuing the message.

Each VS FORTRAN library message is identified by a string of seven characters, followed by a string of up to five characters:

```
  IFYnnnX mmmmm
```

where:

IFY	is the message prefix identifying all VS FORTRAN library messages.
nnn	is the message number.
X	is either an I for informational messages, or an S for severe error conditions.
mmmmm	represents the rightmost 3-5 characters of the name of the library module issuing the message.

Procedure

1. Replace the x of MSGx with the complete message identifier. For example, if the message identifier is IFX1YYYY, the MSGx keyword is MSGIFX1YYYY.
2. Use the name of the module that caused the message to be issued as the module name keyword.
   a. If the message identifier begins with IFX, the compiler module name is prefixed with IFX and the phase number. For example

      ```
      IFX10060 IFAR
      ```

      results in the following keywords:

      ```
      Type of Failure:     MSGIFX10060
      Module Name:         IFX1IFAR
      ```

b. If the message identifier begins with IFY, it is a library message. In this case, the module name is prefixed with IFY. For example

```
IFY207I VFNTH
```

results in the following keywords:

Type of Failure:	MSGIFY207I
Module Name:	IFYVFNTH

3. Proceed with "Release and Modification Levels" on page 14.

NO RESPONSE PROBLEMS

Use the LOOP keyword procedure when a program seems not to be doing anything or is doing something repetitively. However, if the problem looks like a WAIT, it is probably a system problem and you should follow your installation's procedures for resolution.

Procedure

1. If the failure occurs during execution, there is a strong probability that this is a user error. Carefully check your VS FORTRAN source program to be sure it does not contain an endless loop. Adding a DEBUG or a PRINT statement and recompiling may help to detect such a program error. See Appendix A, "Service Aids" on page 18 for instructions on how to use static debug.

2. If you are running in batch mode and the error is a system abend indicating not enough time, it is possible that inadequate time was allotted to compile the program. Increase the time allotment and recompile. If the problem is still not resolved, a set of keywords describing it would look something like this:

Component Identification:	5748F0300
Type of Failure:	LOOP

3. Determine at what levels of optimization the failure occurs.

   If it occurs at only some levels, indicate in the keyword string the levels of optimization at which it does occur. Select the appropriate modifier keyword from the list shown in Appendix B, "Compiler Options" on page 20.

4. Continue with "Release and Modification Levels" on page 14.

DOCUMENTATION PROBLEMS

Use the DOC keyword procedure when a program problem appears to be caused by incorrect or missing information in one of the VS FORTRAN publications.

Procedure

1. Locate the page where the problem occurs in the document, and prepare a description of the error and the problem it caused.

2. Decide whether this documentation problem is severe enough to cause lost time for other users.

   If the problem is not severe, fill out the Reader's Comment Form attached to the back of that manual giving the problem description you developed. Be sure to include your name and return address so IBM can respond to your comments.

If the problem is likely to cause lost time for other users, continue creating your keyword string to determine whether IBM has a record of the problem. Should it be a new problem, you will be asked to submit a severity 3 or 4 (DOC) APAR.

3. Use the order number on the cover of the document, along with the DOC keyword as the type of failure keyword, but omit the hyphens. For example, instead of SC26-3990-4 (this book), enter SC2639904. Your set of keywords would look like this:

```
Component Identification:     5748F0300
Type of Failure:              DOC SC2639904
```

4. Continue the diagnostic procedure with "Release and Modification Levels" on page 14. If, after searching the IBM software support data base, you do not find a matching problem description, **return here to continue**.

5. Before assuming your search argument is not in the data base, you may want to try the search again, using the following format:

```
Component Identification:     5748F0300
Type of Failure:              DOC SC263990**
Release Level:                I75
```

This method allows you to search for all problems reported for that document number rather than a specific release number.

INCORRECT OUTPUT PROBLEMS

Use the INCORROUT keyword procedure when output appears to be incorrect or missing, but the program terminates normally otherwise.

Procedure

1. The failure occurred during compilation.

   a. If you suspect incorrect or missing output from a compilation that otherwise compiled successfully, select a modifier keyword from the following list to describe the type of error in the output.

Modifier	**Type of Incorrect Output**
MISSING	Some expected output was missing.
DUPLICATE	Some records or data were duplicated, but not repeated endlessly (in that case, see "No Response Problems" on page 11).
INVALID	The proper amount of output appeared, but it was not what was expected.

b. Select another modifier keyword from the following list to describe the portion of the output in which the error occurred.

Modifier	**Portion of Output in Error**
DEBUG	Static debug
DUMP	CDUMP, PDUMP, and SDUMP
MAP	Storage map
OBJECT	Machine-language object program
SOURCE	Source listing
SRCFLG	Source listing
STAT	Statistics and error listing
TRMFLG	Terminal screen error listing
XREF	Cross-reference listing

For example, if you think the compiler has produced an incorrect cross-reference listing, your keyword string would look like this so far:

```
Component Identification:      5748F0300
Type of Failure:                INCORROUT
Modifiers:                      INVALID XREF
```

Continue the diagnostic procedure with "Release and Modification Levels" on page 14.

2. The failure occurred during execution.

   a. There is a strong probability that this is a user error. Check your VS FORTRAN source program to be sure it is free of program errors. Adding a DEBUG or PRINT statement and recompiling can help you to detect such an error. See Appendix A, "Service Aids" on page 18 for instructions on how to use static debug statements.

   b. If you suspect incorrect or missing output from a program that otherwise executed successfully, select a modifier from the following list to describe the type of error in the output.

Modifier	**Type of Incorrect Output**
MISSING	Some expected output was missing.
DUPLICATE	Some data or records were duplicated, but not repeated endlessly (in that case, see "No Response Problems" on page 11).
INVALID	The proper amount of output appeared, but it was incorrect.

   c. Continue the diagnostic procedure with "Release and Modification Levels" on page 14.

PERFORMANCE PROBLEMS

Use the PERFM keyword procedure when a performance problem cannot be corrected by system tuning, and performance is below expectations documented in an IBM product publication.

Procedure

1. Record the actual performance and the expected performance measurements for your system configuration. Note the order number and page of the IBM document that is the source of your performance expectations. You will be asked for this information if you contact the IBM support center; if you prepare materials for an APAR, you should also include this information in the error description.

2. Continue the diagnostic procedure with "Release and Modification Levels," below.

RELEASE AND MODIFICATION LEVELS

Use the following procedure to identify the release of the VS FORTRAN Compiler or Library and the current level of modification.

PROCEDURE

1. Locate the 'Level n.n.n' line at the top of your latest printout, or at the top of your trace point 77 dump. This line begins as follows:

   ```
   *LEVEL 1.4.0 (Oct 84)                    VS FORTRAN
   ```

   and also contains the date and time of processing and a page number.

2. Locate the release level keyword in the following chart by release number and operating system.

VS FORTRAN Version Release and Mod. Level	O P E R A T I N G S Y S T E M			
	MVS	VM	VSE	
			Compiler	Library
1.4.0	D00	A40	I75	I76
1.3.1	C03	A31	H53	H54

3. You now have the information necessary for an effective search of known problems documented in a software support data base. Continue with "Using the Keyword String as a Search Argument" on page 15.

USING THE KEYWORD STRING AS A SEARCH ARGUMENT

The following procedure explains how to use the set of keywords you've developed.

Searches against a software support data base will be most successful if VS FORTRAN diagnosticians follow these rules:

- Use only the keywords given in this book.
- Spell keywords exactly as they are presented in this book.
- Include all appropriate keywords in any discussion with IBM support personnel or in any written description of your problem.

If you decide to contact IBM to search the Software Support Facility, be prepared to supply your:

- Customer number
- The set(s) of keywords used to search the software support data base

PROCEDURE

If you do the search yourself, follow the procedure below:

1. Search the software support data base, using the full set of keywords you've developed. Given the following list:

Component Identification:	5748F0300
Type of Failure:	ABENDOC1
Module Name:	IFX2CNTL
Displacement Modifier:	01676
Release Level:	A40

   your keyword string would be:

   5748F0300 ABENDOC1 IFX2CNTL 01676 A40

2. Eliminate those APAR fixes that have already been applied to your system from the list of possible matches.
3. Compare each remaining APAR closing description with the symptoms of your current problem.
4. If a match is found, the problem's solution can be applied to your system through several different methods. Some APAR fixes have "super-zaps" listed within the APAR, while other APAR fixes are available from the IBM Support Center. You may already have the program temporary fix (PTF) at your installation, and only need to install it from the correct program update (PUT) tape.

   Note: For information on applying a specific PTF, refer to the cover letter associated with it.

5. If a match is not found, broaden the search, using the following techniques.

   a. Use the release level keyword for another operating system with the same version, release, and modification level.

   b. Omit the release level keyword completely from the search argument, thereby broadening the search to include similar failures on all other release levels and operating systems.

c. One by one, drop keywords from the right of your keyword string. (The keyword string was developed in a specific sequence to make this technique possible.)

d. Substitute 5748LM303 for the component identification keyword if the error has occurred while running within a module of the VS FORTRAN library.

   The VS FORTRAN library supports programs compiled with the VS FORTRAN compiler, as well as any of the following compilers:

Product Name	**Component ID**
FORTRAN IV H-Extended Compiler	5734F0301
FORTRAN IV G1 Compiler	5734F0201
FORTRAN IV F Compiler	360NF0479

   If you are using one of these compilers in conjunction with the VS FORTRAN library, search the software support data base, using that compiler's component identification keyword along with the VS FORTRAN library's identification number.

e. Consider using a synonym as a replacement for a modifier keyword. The problem may have been entered into the data base using a slightly different expression than the now-recommended format.

6. If a match is not found using the preceding techniques, go to "Preparing an APAR" on page 17.

PREPARING AN APAR

Prepare to submit an APAR only after the preceding diagnostic procedures have been followed, and the keyword search proves unsuccessful.

PROCEDURE

1. If you think you need to submit an APAR, contact the IBM Support Center. Often you will already have been in contact with IBM while searching a data base and attempting to resolve the problem. In either case, be prepared to supply your:
   - Customer number
   - The keyword string(s) used to search the software support data base
2. The support personnel will review the problem with you, and give you an APAR number when you and they have agreed that an APAR is necessary.
3. You may be asked to supply various types of information to help prepare the APAR. Applicable items from the following list may be necessary:
   - Job control statements
   - Link-edit or loader map output
   - Compiler listings, including:
     - Source listing
     - Object listing
     - Storage map
     - Cross-reference listing
   - Any debugging output
   - Under VM/SP, any spooled CMS console output or special CMS EXECs
   - Machine-readable copy of the application program experiencing the problem
   - A description of the application program and the organization of its data sets

APPENDIX A. SERVICE AIDS

SOURCE STATEMENTS AND OPTIONS

For more comprehensive information on the following statements and options, refer to VS FORTRAN Language and Library Reference.

STATIC DEBUG STATEMENTS

AT Statement: specifies the beginning of a debugging packet.

DEBUG and END DEBUG Statements: delimit the debugging portions of a program.

Debug Options	Description
INIT	displays a variable or an array name in the debug output when the variable or array element is assigned a new value.
SUBCHK	checks the validity of the subscripts used with the named arrays.
SUBTRACE	specifies that the name of this subprogram is to be produced whenever it is entered, and RETURN is printed when the subprogram is exited.
TRACE	specifies portions of a program to be traced.
UNIT	specifies a particular data set for all debugging output.

DISPLAY Statement: displays data within a debugging packet.

TRACE/ON and OFF Statements: specify when to begin and end tracing.

Note: Any FORTRAN statement can be used in conjunction with static debug statements. Refer to 1d on page 7 for a list of VS FORTRAN statements. These statements can only be used at optimization level 0 and with the NORENT option.

VS FORTRAN INTERACTIVE DEBUG

If you have VS FORTRAN Interactive Debug (product number 5668-903) at your installation, you can use it to help determine the statement causing the error. With VS FORTRAN Interactive Debug, you can suspend program execution, continue execution, skip sections of code, correct errors, set up expected input, and display output. You can also use VS FORTRAN Interactive Debug to debug optimized code, with some restrictions. For more information, see VS FORTRAN Interactive Debug Guide and Reference.

DUMP AND PDUMP SUBROUTINES

DUMP is a service subroutine used to dump storage and terminate object program execution. PDUMP is a service subroutine used to dump storage and continue object program execution. The DUMP and PDUMP subroutines are invoked by the CALL statement.

CDUMP AND CPDUMP SUBROUTINES

CDUMP is a service subroutine used to dump storage in character format and terminate object program execution. CPDUMP is a service subroutine used to dump storage in character format and continue object program execution. The CDUMP and CPDUMP subroutines are invoked by the CALL statement.

SDUMP SUBROUTINE

The Symbolic Dump (SDUMP) subroutine displays a symbolic dump of all the variables in one or more program units. The variables are displayed in a format that is determined by the variable types and sizes declared, or defaulted, in the source program. The SDUMP subroutine is invoked by the CALL statement.

COMPILER TRACE OPTION

TRACE

The TRACE compiler option indicates that the compiler diagnostic trace information is to be produced. The TRACE option is only valid as part of an @PROCESS statement. The format of the @PROCESS statement using the TRACE option is: @PROCESS beginning in column 1 and followed by TRACE and any other options.

Multiple options are separated by one or more blanks. The @PROCESS statement must be the first statement in the source program. Further details on the @PROCESS statement are found in VS FORTRAN Programming Guide.

When a program interrupt occurs within the FORTRAN Compiler and the TRACE option is on, a special trace point is activated in the interrupt handler. This trace point is known as trace point 77; it provides a printout of the PSW, all general registers, save areas for all active modules, and formatted printouts of all available tables.

Other trace points are available but are used only at the request of the IBM product specialist.

See Figure 3 on page 9 for sample output from the trace point 77.

APPENDIX B. COMPILER OPTIONS

Use the modifier keywords shown below to identify compiler options in the keyword string.

Option	Modifier Keywords			
AUTODBL (value)	AUTODBL			
CHARLEN (number)	CHARLEN			
CI (number)	CI			
DC (name)	DC			
DECK or NODECK	DECK	NODECK		
FIPS (F|S) or NOFIPS	FIPSF	FIPSS	NOFIPS	
FLAG (I|W|E|S)	FLAGI	FLAGW	FLAGE	FLAGS
FREE or FIXED	FREE	FIXED		
GOSTMT or NOGOSTMT	GOSTMT	NOGOSTMT		
LANGLVL (66|77)	LANGLVL66	LANGLVL77		
LINECOUNT (number)	LINECOUNT			
LIST or NOLIST	LIST	NOLIST		
MAP or NOMAP	MAP	NOMAP		
NAME (name)	NAME			
OBJECT or NOOBJECT	OBJECT	NOOBJECT		
OPTIMIZE (0|1|2|3) or NOOPTIMIZE[1]	OPT0	OPT1	OPT2	OPT3
RENT or NORENT	RENT	NORENT		
SDUMP or NOSDUMP	SDUMP	NOSDUMP		
SOURCE or NOSOURCE	SOURCE	NOSOURCE		
SRCFLG or NOSRCFLG	SRCFLG	NOSRCFLG		
SXM or NOSXM	SXM	NOSXM		
SYM or NOSYM	SYM	NOSYM		
TERMINAL or NOTERMINAL	TERM	NOTERM		
TEST or NOTEST	TEST	NOTEST		
TRMFLG or NOTRMFLG	TRMFLG	NOTRMFLG		
XREF or NOXREF	XREF	NOXREF		

Note:

[1] The NOOPTIMIZE option is the same as OPTIMIZE 0.

VS FORTRAN
Compiler and Library:
Diagnosis Guide
SC26-3990-4

Reader's Comment Form

This manual is part of a library that serves as a reference source for systems analysts, programmers, and operators of IBM systems. You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Your comments will be sent to the author's department for whatever review and action, if any, are deemed appropriate.

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

List TNLs here:

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

Last TNL ____________________

Previous TNL ____________________

Previous TNL ____________________

Fold on two lines, tape, and mail. No postage stamp necessary if mailed in the U.S.A. (Elsewhere, an IBM office or representative will be happy to forward your comments or you may mail directly to the address in the Edition Notice on the back of the title page.) Thank you for your cooperation.

Please use pressure sensitive or other gummed tape to seal this form.

SC26-3990-4

Reader's Comment Form

Fold and tape   Please do not staple   Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

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

POSTAGE WILL BE PAID BY ADDRESSEE

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

Fold and tape   Please do not staple   Fold and tape

IBM®

Compiler and Library: Diagnosis Guide (File No. S370-37) Printed in U.S.A. SC26-3990-4

VS FORTRAN
Compiler and Library
Diagnosis Guide

File No. S370-37

SC26-3990-4

Printed in U.S.A.

SC26-3990-4

IBM ®

10272 1978