IBM Library Server Print Preview

DOCNUM = GC09-2426-00
DATETIME = 10/01/96 05:12:12
BLDVERS = 1.2
TITLE = C/VSE V1R1 Diagnosis Guide
AUTHOR =

PATH = /home/webapps/epubs/htdocs/book

COVER Book Cover

IBM C for VSE/ESA

Diagnosis Guide

Release 1

Document Number GC09-2426-00

Program Number
5686-A01

NOTICES Notices

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" in topic FRONT_1.

EDITION Edition Notice

First Edition (December 1996)

This edition applies to Version 1, Release 1, Modification Level 0, of IBM C for VSE/ESA (Program 5686-A01); Version 1, Release 4, Modification Level 0, of IBM Language Environment for VSE/ESA (Program 5686-094); the VSE C Language Run-Time Support feature of VSE/ESA Version 2 Release 2 (Program 5690-VSE); and to all subsequent releases and modifications until otherwise indicated in new editions. Make sure you are using 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 readers' comments is provided at the back of this publication. If the form has been removed, address your comments to:

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 Eglinton Avenue East
North York, Ontario, Canada M3C 1H7

You can also send your comments by facsimile (attention: RCF Coordinator), or you can send your comments electronically to IBM. See "Communicating Your Comments to IBM" for a description of the methods. This page immediately precedes the Readers' Comment Form at the back of this publication.

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.

CONTENTS Table of Contents

FIGURES Figures

TABLES Tables

FRONT_1 Notices

purpose of enabling: (i) the exchange of information between independent created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM Canada Ltd., Department 071, 1150 Eglinton Avenue East, North York, Ontario M3C 1H7, Canada. Such information may be available, subject to appropriate terms and conditions, including in some cases payment of a fee.

Subtopics:

FRONT_1.1 Programming Interface Information

This book is intended to help you to do diagnosis of the IBM C for VSE/ESA compiler. This book documents information which is Diagnosis, Modification, and Tuning Information provided by IBM C for VSE/ESA.

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

FRONT_1.2 Trademarks

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

AIX/6000	Language Environment	OS/400
C/370	MVS	RETAIN
IBM	OS/2	SAA
IBMLink	OS/390	VSE/ESA

The following terms are trademarks of other companies:

ANSI	American National Standards Institute
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronic Engineers
ISO	International Organization for Standardization
POSIX	Institute of Electrical and Electronic Engineers

PREFACE About This Book

This book tells you how to diagnose failures in the IBM C for VSE/ESA compiler (C/VSE). C/VSE uses IBM Language Environment for VSE/ESA (LE/VSE) as a run-time environment, so to diagnose product failures that you encounter in that run-time environment, see the *LE/VSE Debugging Guide and Run-Time Messages*.

Note: References to LE/VSE also apply to the VSE C Language Run-Time Support feature of VSE/ESA Version 2 Release 2.

The book assumes that you have already determined that the suspected failure is not a user error; that is, it was not caused by incorrect use of C/VSE or by an error in the logic of the application program. If a user error is the cause of the problem, and the "Initial Checklist" in topic 1.1.1 does not address your case, then see the following books for

more information:

- *LE/VSE C Run-Time Programming Guide*, SC33-6688
- *LE/VSE Debugging Guide and Run-Time Messages*, SC33-6681
- *Debug Tool for VSE/ESA User's Guide and Reference*, SC26-8797

This book helps you to determine if a correction for a product failure similar to yours has been previously documented. If the problem has not been previously reported, Chapter 9, "Preparing an Authorized Program Analysis Report (APAR)" in topic 9.0 explains how to open a Problem Management Report (PMR) to report the problem to IBM, and if the problem is with an IBM product, how to prepare an Authorized Program Analysis Report (APAR).

This book is for system programmers, application programmers, and IBM Support personnel who are involved in C/VSE product diagnosis. You should have:

- A general understanding of the VSE operating system.
- Some knowledge of the C/VSE language and options.
- A basic knowledge of how C/VSE and LE/VSE work together. This is summarized in the introductory material to this book.
- Some knowledge of the IBM database, such as RETAIN, that you intend to search for known similar problems (this applies only if you have an electronic link to this type of IBM database).

Subtopics:

PREFACE.1 The C Language

The C language is a general purpose, function-oriented programming language that allows a programmer to create applications quickly and easily. C provides high-level control statements and data types as do other structured programming languages, and it also provides many of the benefits of a low-level language. Using the C/VSE language, you can write portable code conforming to the ANSI standard.

IBM offers the C language on other platforms, such as the OS/2, AIX/6000, OS/400, OS/390, and VM operating systems.

The elements of the C/VSE implementation include:

- All elements of the joint ISO and IEC standard: ISO/IEC 9899:1990 (E)
- ANSI/ISO 9899:1990[1992] (formerly ANSI X3.159-1989 C)
- Locale based internationalization support as defined in: ISO/IEC DIS 9945-2:1992/IEEE POSIX 1003.2-1992 Draft 12 (There are some limitations to fully-compliant behavior as noted in the *LE/VSE C Run-Time Programming Guide*.)
- Extended multibyte and wide character utilities as defined by a subset of the Programming Language C Amendment 1, which will be ISO/IEC 9899:1990/Amendment 1:1994(E)

PREFACE.2 IBM Language Environment for VSE/ESA

C/VSE exploits the C run-time environment and library of run-time callable services provided by IBM Language Environment for VSE/ESA (LE/VSE).

LE/VSE establishes a common run-time environment and common run-time callable services for language products, user programs, and other products.

The common execution environment is made up of data items and services performed by library routines available to a particular application running in the environment. The services that LE/VSE provides to your application may include:

- Services that satisfy basic requirements common to most applications. These include support for the initialization and termination of applications, allocation of storage, support for interlanguage communication (ILC), and condition handling.
- Extended services often needed by applications. These functions are contained within a library of callable routines, and include interfaces to operating system functions and a variety of other commonly used functions.
- Run-time options that help the execution, performance tuning, performance, and diagnosis of your application.
- Access to language-specific library routines, such as the C functions.

PREFACE.3 Using Your Documentation

The publications in the C/VSE and LE/VSE libraries are designed to help you develop C/VSE applications that run with LE/VSE. Each publication helps you perform a different task. For a complete list of publications you might need, see "Bibliography" in topic BIBLIOGRAPHY. Table 1 lists the publications in the C/VSE library.

Table 1. How to Use C/VSE Publications

To...	Use...	
Plan for, install, customize, and maintain C/VSE	Installation and Customization Guide	GC09-2422
Migrate VSE applications from C/370 to C/VSE	Migration Guide	SC09-2423
Get details on C/VSE syntax and specifications of language elements	Language Reference	SC09-2425
Find syntax for compile-time options; compile your C/VSE applications; get details on compile-time messages	User's Guide	SC09-2424
Diagnose compiler problems and report them to IBM	*Diagnosis Guide*	GC09-2426
Understand warranty information	*Licensed Program*	GC09-2421

	Specifications	

Table 2 lists the publications in the LE/VSE library. These include publications designed to help you develop and debug your C/VSE applications, diagnose run-time problems that occur in your C/VSE applications, and use C/VSE-related utilities.

Table 2. How to Use LE/VSE Publications

To...	**Use...**	
Evaluate LE/VSE	*Fact Sheet*	GC33-6679
	Concepts Guide	GC33-6680
Plan for, install, customize, and maintain LE/VSE	*Installation and Customization Guide*	SC33-6682
Understand the LE/VSE program models and concepts	*Concepts Guide*	GC33-6680
	Programming Guide	SC33-6684
Find syntax for LE/VSE run-time options and callable services	*Programming Reference*	SC33-6685
Develop your C/VSE applications	*Programming Guide*	SC33-6684
	C Run-Time Programming Guide	SC33-6688
	C Run-Time Library Reference	SC33-6689
Develop interlanguage communication (ILC) applications	*Writing Interlanguage Communication Applications*	SC33-6686
Debug your C/VSE applications and get details on run-time messages	*Debugging Guide and Run-Time Messages*	SC33-6681
Migrate applications to LE/VSE	*Run-Time Migration Guide*	SC33-6687
Diagnose run-time problems that occur in your C/VSE applications	*Debugging Guide and Run-Time Messages*	SC33-6681
Use C/VSE-related utilities	*C Run-Time Programming Guide*	SC33-6688
Understand warranty information	*Licensed Program Specifications*	GC33-6683

1.0 Chapter 1. Isolating Reportable Problems

Failures in C/VSE can be described through the use of *keywords*. A keyword is a word or abbreviation assigned to describe one aspect of a product failure. A set of keywords, called a *keyword string*, describes the failure in detail. The procedures in this section help you construct a keyword string that describes what you currently know about the compiler failure. This book is designed to help you specifically with failures in C/VSE, but also contains general information on reporting problems. For more information about failures that occur in the LE/VSE C run-time library, see the *LE/VSE Debugging Guide and Run-Time Messages*. For more information about user compile-time problems, see the *C/VSE User's Guide*; and for more information about user run-time errors, see the *LE/VSE Debugging Guide and Run-Time Messages*.

After you construct the keyword string, it is used as a search argument against an IBM software support database, such as RETAIN. The database contains keyword and text information describing all current problems reported through Authorized Program Analysis Reports (APARs) and associated Program Temporary Fixes (PTFs). IBM Support Center personnel have access to the software support database and are responsible for storing and retrieving the information. Using the keyword string, they will search the database to retrieve records that describe similar known problems.

If you have IBMLink or some other connection to the IBM databases you can do your own search for previously recorded product failures before calling the IBM Support Center.

If the keyword string produces a match in the software support database, the search may yield a fuller description of the problem and possibly identify a correction or circumvention. Such a search may yield several matches to previously reported problems. Review each error description carefully to determine if the problem description in the database matches the failure.

If a match is not found, use the keyword string you have constructed to describe the failure when contacting the IBM Support Center for assistance and when submitting an APAR. Keywords are intended to ensure that identical program errors will be described with identical keyword strings. Spelling the keywords exactly as they are presented in this book is especially important for a successful match. For additional information on keywords and APAR preparation, see *Programming Systems General Information*, G229-2228.

Subtopics:

1.1 Diagnosis Procedure

Step through each of the items in "Initial Checklist" in topic 1.1.1 to see if they apply to your problem. These suggestions are designed to help you gather the diagnostic information required for determining the source of the error and to develop a keyword string to search the software support database. It describes options which, if specified when searching the database, will supply you with all available diagnostic information. You will need this information to discuss the problem with your IBM support representative if your search fails to locate a fix for your problem.

If a problem occurs while you are using C/VSE, its cause may not be obvious; it might be an error in your application or in C/VSE itself. After you identify the failure, you may want to write a small test case that re-creates the problem. It will help you to:

- Pinpoint the problem
- Determine whether the error is in the application program or in C/VSE
- Choose keywords that best describe the error if it is in C/VSE

You can read more about creating a test case in "Creating a Test Case" in topic 1.2.1.

If the problem appears to be the result of an error in the application program, change your source code accordingly and then compile and link the new code. If the problem appears to be the result of an error in C/VSE, you should develop a set of keywords using the procedure in "Keyword Usage" in topic 1.3, then contact your Service Representative.

Use the following procedures to diagnose the problem. As you go through the procedures, always note the sequence of events that leads to the error.

Subtopics:

1.1.1 Initial Checklist

The following list contains suggestions to help you diagnose the problem and determine whether there is an error in C/VSE.

1. Check that the program has not been changed since you last compiled or executed it successfully. If it has, examine the changes. If the error occurs 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.

2. Be sure you have corrected all problems diagnosed by error messages and ensured that messages previously generated have no effect on the current problem. Be sure to pay attention to warning messages.

3. The message prefix and number can identify the system or subsystem that issued the message. This can help you determine the cause of the problem. Following are some of the prefixes and numbers, and their respective origins.

Messages	**Issued by**
EDC0001-EDC1999	C/VSE compiler
EDC4000-EDC4999	Prelinker and C utilities
EDC5000-EDC5599	C language run-time component of LE/VSE
CEE*nnnn*	Common execution environment (CEE) component of LE/VSE

   For information about messages issued by the C/VSE compiler, see the *C/VSE User's Guide*. For information about other EDC prefix messages, and CEE prefix messages, see the *LE/VSE Debugging Guide and Run-Time Messages*.

   Other prefixes indicate that the message was issued by another system or subsystem. See the appropriate messages manual for the system or subsystem.

   For VSE operating system messages, see *VSE/ESA Messages and Codes*, SC33-6507 (VSE/ESA Version 1) or SC33-6607 (VSE/ESA Version 2).

4. Ensure that you are compiling the correct version of the source code. It is possible that you have incorrectly indicated the location of your source file. For example, if you have reassigned SYSIPT to a disk file, check that the disk file contains the correct source code. Also, ensure that the LIBDEF SOURCE search chain specifies the correct location of any header files specified by #include statements in your source code.

5. In any program failure, keep a record of the conditions and options in effect at the time the problem occurred. The options are listed in the listing file. To get the listing, compile with the SOURCE option. The listing only contains options appearing after the command line is

processed, hence #pragma options do not appear.

Information about some of the options appears as a comment at the bottom of the text deck. There is always a comment about the status of the OPTIMIZE option whether you specify it or not. Information about the GONUMBER, INLINE, RENT, or UPCONV options is included only if you specify the option when compiling. Note any change from the previous compilation.

6. Your installation may have received an IBM Program Temporary Fix (PTF) for the problem. Make sure that you have all PTFs and that all PTFs have been installed, so that your installation is at the latest maintenance level.

7. The preventive service planning (PSP) bucket, an online database available to IBM customers through IBM service channels, gives information about product installation problems and other problems. See the *C/VSE Installation and Customization Guide* for more details.

8. Use Debug Tool for VSE/ESA, an optional feature of C/VSE, or some other debugging aid to determine the statement where the program fails and possible causes of failure. You can also use the GONUMBER compile-time option to print statement numbers in the traceback listing.

9. If a failing application is communicating with other IBM products, make sure that the correct interface procedure is used as documented in the *LE/VSE C Run-Time Programming Guide*. In many cases, you can localize the failing condition by taking out the function calls or making them no-ops.

10. If your application was developed on a different platform (such as a microcomputer or workstation) and then you try to compile and run using C/VSE, problems may occur if the source code:

    - Does not conform to the *American National Standards Institute (ANSI) C Standard*, (X3.159-1989) or SAA standards
    - Includes dependencies on the ASCII character set or the IEEE floating-point format
    - Is system dependent

    For more information on using code developed on other systems, see the *Portability Guide for IBM C*, SC09-1405.

11. If your application was prelinked, make sure that the prelinking was successful as indicated in the *C/VSE User's Guide*.

1.1.2 When Does the Error Occur?

Determine when the problem is occurring (at compile time, prelink time, link time, or run time) and use the procedures in the appropriate list on the following pages. If the problem occurs when using LE/VSE, for link-time and run-time diagnosis and debugging errors you should be using the *LE/VSE Debugging Guide and Run-Time Messages*.

After you identify the failure, you can write a small test case that re-creates the problem. See "Creating a Test Case" in topic 1.2.1, for details on constructing a test case from a failing program.

If the error occurs at compile time:

1. If your program uses any of the library routines, insert an #include directive for the appropriate header files. Also insert an #include directive for any of your own header files. Function prototypes, when present, are used by the compiler to help detect type mismatches on function calls. You can use the CHECKOUT option to find missing prototyping.

2. Compile your program using the CHECKOUT option. The CHECKOUT option specifies that the compiler is to give informational messages indicating possible programming errors. This option will give messages about such things as variables that are never used, and the tracing of include files.

3. Compile your program using the PPONLY option to see the results of all #define and #include statements. This option also expands all macros; a macro may have a different effect than you intended.

4. If your program was originally compiled using the OPTIMIZE option, recompile it using the NOOPTIMIZE option and then run it again. If you can successfully compile and run the program using NOOPTIMIZE, you have bypassed the problem, but not solved it. This also does not exclude the possibility of an error in your program. Even if you still have a problem, you can run the program until you find a permanent solution.

5. If you compiled your program with either the SEQUENCE or the MARGINS option, the error may be the result of a loss of code. Similarly, source code with sequence numbers compiled with NOSEQUENCE will try to parse the sequence numbers as C code, often with surprising results. This can happen in a source file that was meant to be compiled with margins but was actually compiled with no margins or different margins.

   This can cause either syntax errors or unexpected results when your program runs. Try recompiling the program using either the NOSEQUENCE or the NOMARGINS option.

6. Your source file may contain characters that are not supported by your terminal. Replace any characters that cannot be displayed in literals by the corresponding trigraph representation or the corresponding escape sequence. Verify that the error was not the result of one of these having been used incorrectly.

7. A compile-time abend can indicate an error in the compiler. An unsuccessful compilation, due to an error in the source code or an error from the operating system, should result in error messages, not an abend. However, the cause of the compiler failure may be a syntax error or an error from the operating system.

If the error occurs at prelink time:

1. Do not prelink the object modules separately.

2. Use the prelinker option MAP to obtain a full map of input members and symbols.

3. If you have unresolved symbols, make sure that the definition of an object and all references to that object are used consistently in both the code area and the writable static area. Also, make sure that

symbol references appear consistently in the same case.

4. Only naturally reentrant code can be linked with the output of the prelinker. For more information, see the *C/VSE User's Guide*.

If the error occurs at link time:

1. Linkage editor messages are described in *VSE/ESA Messages and Codes*, SC33-6507 (VSE/ESA Version 1) or SC33-6607 (VSE/ESA Version 2).

If the error occurs at run time:

1. Use the _ _librel() function to determine which LE/VSE library release level your program is using. See the *LE/VSE C Run-Time Library Reference* for more information on _ _librel().

2. If the problem occurs during execution, specify one or more of the following options in the compilation, in addition to the options originally specified, to produce maximum diagnostic information:

Option	**Information produced**
AGGREGATE	Aggregate layout.
CHECKOUT	Indication of possible programming errors.
EXPMAC	Macro expansions with the original source.
INLINE	Inline Summary and Detailed Call Structure Reports. (Specify with the REPORT suboption.)
LIST	Listing of the pseudoassembler listing produced by the compiler.
OFFSET	Offset addresses of functions in the listing.
PPONLY	Completely expanded C source code, by activating the preprocessor (PP) only. The output shows, for example, all the #include and #define directives.
SHOWINC	All included text in the listing.
SOURCE	Listing of the source file.
XREF	Cross reference listing.

3. If the failure is in a statement that can be isolated, for example, an if, switch, for, while, or do...while statement, try placing the failing statement in the mainline code. If the problem is occurring as a result of a switch statement, make sure that you have "breaks" on all appropriate statements.

4. If C/370 is installed on your system, ensure that the LE/VSE and C/VSE sublibraries appear before C/370 in the LIBDEF PHASE and LIBDEF SOURCE search chains. Ensure that C/VSE has not been installed in the same sublibrary as C/370.

5. Diagnostics and debugging information for run-time errors are found in the *LE/VSE Debugging Guide and Run-Time Messages*.

1.2 Installation Problems

You can avoid or solve most installation problems if you follow these steps:

1. Consult the PSP bucket as described under "Initial Checklist" in item

7.

2. Review the step-by-step installation procedure documented in the *C/VSE Installation and Customization Guide*.

If you still cannot solve the problem, develop a keyword string and contact your IBM Support Center.

The C/VSE product may have to be reinstalled using the procedure documented in the *C/VSE Installation and Customization Guide*. This procedure is tested for each product release and successfully installs the product.

Subtopics:

1.2.1 Creating a Test Case

You can create a test case to help isolate the problem and to use when you report the problem to IBM. To create a small test case from a large program that appears to be failing, try the suggestions listed below, after you have either backed up or made a copy of your original source code.

Begin with the suggestion that seems most appropriate for the problem that you are having. If the problem persists after you have tried one of the steps below, try another in the list.

Continue to break your program down until you obtain the smallest possible segment of code that still contains the error. Compile with the PPONLY option and send the expanded file as your source code. This is to ensure that all embedded header files are included. Save this last failing test case because you might need it if you have to contact an IBM Support Center.

Use of #include directives
If your program uses #include directives, put all the relevant code from the #include file directly in the main file. Use the PPONLY option and then use the expanded file as your source for submission to your IBM representative.

Unrelated code
Remove unrelated code if it is not necessary for syntactic or semantic validity.

Unreferenced variables
Remove unreferenced variables. You can find them by using the XREF or CHECKOUT compile-time option.

Compile-time failure
Remove any code that has not been processed at the time of failure (except for code necessary to ensure the syntactic and semantic validity of the program).

Compile-time failure in a function
Remove all code and declarations from the body of any other functions.

Use of structure variables
If your program uses structure variables, try replacing them with scalar variables.

1.3 Keyword Usage

The first keyword of a keyword string is called the component identification, or ID. The component identification for C/VSE is the compiler identifier, 5686A0100. A search of the software support database with this single keyword would locate all problems reported for the compiler. For a list of component identifiers you might need, see Chapter 2, "Component Identification Keyword" in topic 2.0.

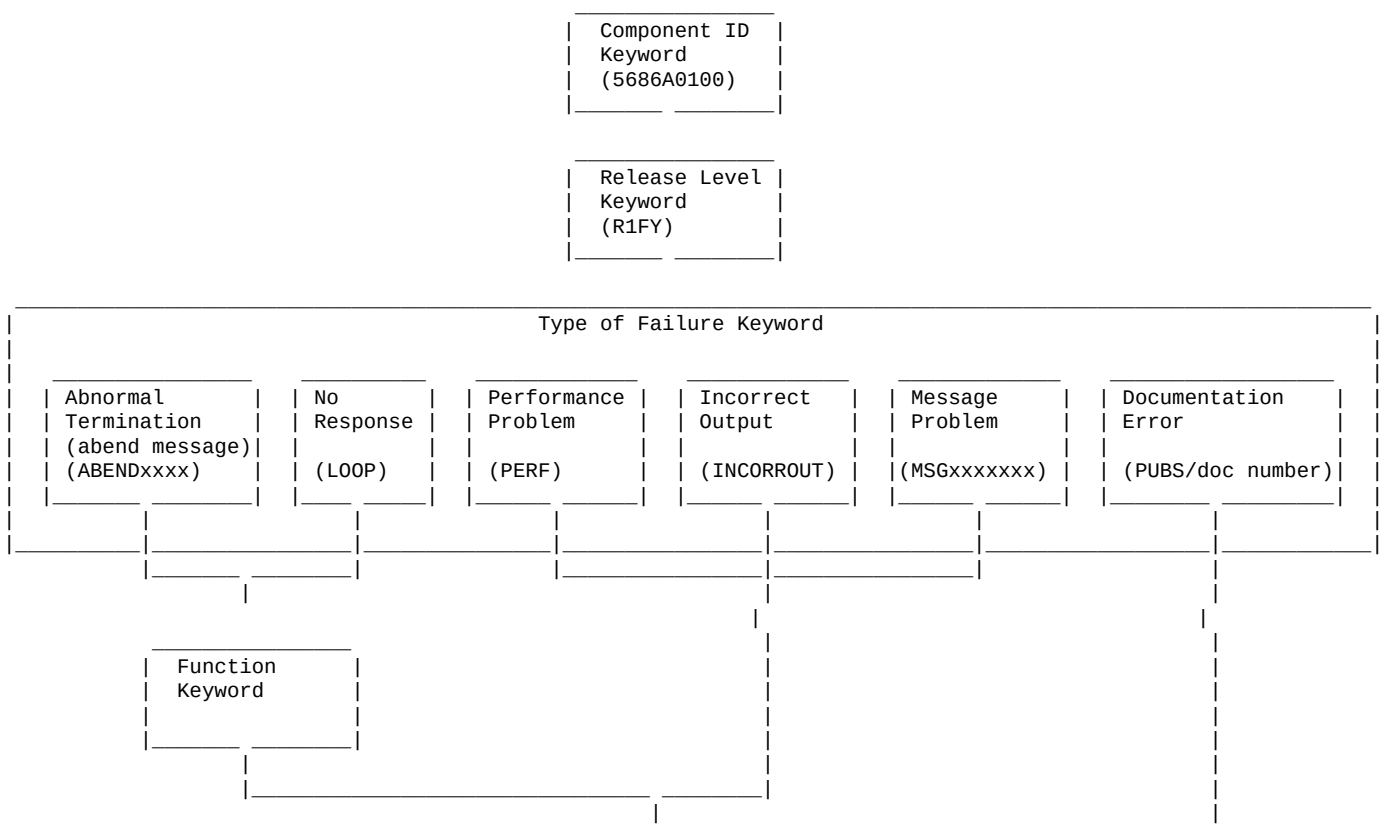
Each additional keyword added to the keyword string narrows the scope of the search and helps to eliminate unnecessary examination of problem descriptions that have similar, but not matching, characteristics. In some cases, a correction for a product failure might be located with less than a full string of keywords. If it is unclear how to select a particular keyword to describe your problem, omit that keyword to avoid incorrectly identifying the problem.

A full set of keywords for C/VSE contains:

- The component identification
- The release level
- The type of failure
- The name of the module that failed, if applicable and available
- The system type
- One or more modifier keywords, depending on the type of failure, if applicable

Follow the steps in the keyword procedures until you are directed to use the keyword string in a search argument.

Figure 1 illustrates the process of creating a keyword string.



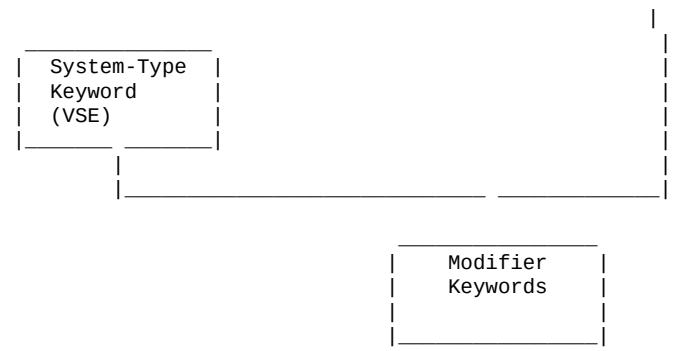


Figure 1. C/VSE-Problem Identification Using Keywords

1.4 Using the Problem Identification Worksheet

You can use Chapter 10, "Problem Identification Worksheet" in topic 10.0 to help you construct and record a keyword string. As you identify the keywords associated with your software problem, just record them in the spaces provided.

2.0 Chapter 2. Component Identification Keyword

This procedure shows you what to specify in the component identification keyword. This is always the first keyword placed in the search argument string. It comes from the C/VSE program number and limits the search to the area within the software support database that contains items for C/VSE.

Component identifiers you might need are:

5686A0100 C/VSE compiler

5686A0101 Japanese national language support (NLS) component of C/VSE

568609401 Common execution environment (CEE) library component of LE/VSE

568609408 C language run-time component of LE/VSE

The component identification keyword should be used with at least a type-of-failure keyword to search the software support database. If it is used without additional keywords, a full listing of all items affecting C/VSE will be produced.

1. Use 5686A0100 as the component identification keyword for the C/VSE compiler or 5686A0101 if the problem is in the Japanese NLS component of C/VSE. (The component identification keyword for the C language run-time component of LE/VSE is 568609408. Diagnostics for LE/VSE are explained in the *LE/VSE Debugging Guide and Run-Time Messages*.)

2. If service tapes have been applied to the licensed program, note the tape level of the last service tape applied. See your system programmer for the current service level of your C/VSE compiler. Although the service tape level is not used as a keyword search

argument, it will be useful when reviewing APARs selected during the keyword search.

3. Continue the diagnostic procedure with Chapter 3, "Release Level Keyword" in topic 3.0.

3.0 Chapter 3. Release Level Keyword

The release level keyword identifies the specific release level of C/VSE you were using when the failure occurred.

1. The release level keyword for the C/VSE Version 1 Release 1 compiler is R1FY. The release level keyword for the corresponding Japanese NLS component is R1G4 (use this if you specified the Japanese NLS component in the component identifier keyword).

   If you do not know the release level code for your product level, you can ask your IBM Support Center.

   The release level keyword is the release level code for your system prefixed with an R. For example, if the release code for you product is 1FY, the release level keyword is R1FY.

2. Continue the diagnostic procedure with Chapter 4, "Type-of-Failure Keyword" in topic 4.0.

4.0 Chapter 4. Type-of-Failure Keyword

A specific type of failure may occur in C/VSE. For product failures that occur in your C/VSE applications during LE/VSE run-time, consult the *LE/VSE Debugging Guide and Run-Time Messages*.

Read the following table and select the type of failure that best describes your problem. Then go to the associated keyword procedure for instructions on how to complete the keywords for that type of failure. If more than one keyword describes your problem, use the one that appears first in the table.

Table 3. Types of C/VSE Failures

Type-of-Failure	Symptom	Procedure
Abnormal Termination	C/VSE terminated abnormally with a completion code that indicates that an abend has occurred.	See "Abnormal Termination" in topic 4.1.
Message Problems	C/VSE issued a message that is inappropriate or not valid.	See "Message Problems" in topic 4.2.
No Response from the Compiler	Either an unexpected program suspension has occurred or the batch job has not completed.	See "No Response Problems" in topic 4.3.
Documentation Problems	Information in one of the C/VSE publications is incorrect or missing.	See "C/VSE Documentation Problems" in topic 4.4.
Output Problems	The output from the compiler	See "Output

	is missing or not valid.	Problems" in topic 4.5.
Performance Problems	The performance of a C/VSE compilation or program is degraded.	See "Performance Problems" in topic 4.6.

Subtopics:

4.1 Abnormal Termination

If C/VSE terminates abnormally you will get one or more of the following:

- A message from LE/VSE
- A traceback
- A system abend, such as an addressing exception

If you get an LE/VSE message but no traceback after an abnormal compiler termination, use the *LE/VSE Debugging Guide and Run-Time Messages* to diagnose the problem.

If you get a message from the operating system indicating a system abend but no traceback, follow the procedure described in "System Abend Procedure" in topic 4.1.1.

If you get a traceback, use the procedure described in Chapter 5, "Function Keyword" in topic 5.0. Record any other messages you get. You will need the other messages in order to do your keyword search or to report the problem.

Subtopics:

4.1.1 System Abend Procedure

Use this procedure if the C/VSE compiler terminates abnormally with a VSE system abend message.

1. Replace ABEND*xxxx* with the relevant words from the text of the abend message. For example, if you receive the message

   ```
   0S03I PROGRAM CHECK INTERRUPTION - HEX LOCATION 0006D2D0 -
   CONDITION CODE 0 - PROTECTION EXCEPTION
   ```

   you should use PROTECTION and EXCEPTION as keywords. The following is a example of a keyword string for C/VSE Version 1 Release 1 when the

system abend is accompanied by the above message:

```
5686A0100 R1FY PROTECTION EXCEPTION
```

If you use the above method, but when you search the IBM support database you do not find a closed APAR with a description that matches your current failure symptoms, you can use the equivalent MVS abend code as the type-of-failure keyword. Check the following list of common program checks for a match on the keywords you used. If you find a match, you can use the equivalent MVS system abend code to replace the *xxxx* of ABEND*xxxx*. Add a zero to the beginning of the code to complete the *xxxx* field. For example, if the code is 0C4, specify ABEND00C4 as your type-of-failure keyword.

Program Check	**MVS System Abend Code**
Operation exception	0C1
Privileged operation exception	0C2
Execute exception	0C3
Protection exception	0C4
Addressing exception	0C5
Specification exception	0C6
Data exception	0C7
Fixed-point overflow exception	0C8
Fixed-point divide exception	0C9

2. If you did not get a listing continue with Chapter 6, "System-Type Keyword" in topic 6.0.

3. Depending on the type of abend, you may receive a listing. The compiler phase is shown in the listing. Use the name of the current C/VSE compiler phase as a modifier keyword. If the compiler phase was EDCDP, your keyword string would appear as follows:

```
5686A0100 R1FY PROTECTION EXCEPTION EDCDP
```

or (if you found no match for PROTECTION EXCEPTION)

```
5686A0100 R1FY ABEND00C4 EDCDP
```

4. Continue with Chapter 6, "System-Type Keyword" in topic 6.0.

4.2 Message Problems

The appearance of a C/VSE message normally indicates a correctable problem in the source code that is being compiled. Do not use this procedure to diagnose source code problems. Use this diagnosis procedure when one of the following occurs during compile time:

- A message is issued when it should not have been issued.
- A message contains data that is not valid or data is missing.
- A message is issued and the description of the message indicates that the problem described by the message is an LE/VSE problem or a C/VSE problem.

For information about where to find message descriptions, see item 3 under "Initial Checklist" in topic 1.1.1.

Do not use this procedure when an abnormal termination message is received. In that case, refer to "Abnormal Termination" in topic 4.1.

To construct the MSG*xxxxxxx* keyword:

1. Replace the *xxxxxxx* of MSG*xxxxxxx* with the message identifier. For example, if the compiler message received is EDC0041 the MSG*xxxxxxx* keyword would be MSGEDC0041. Do not include a message severity code in the keyword. Your set of keywords, so far, would look like this:

   ```
   5686A0100 R1FY MSGEDC0041
   ```

2. Proceed with Chapter 6, "System-Type Keyword" in topic 6.0.

4.3 No Response Problems

Use the LOOP keyword procedure if C/VSE seems to be doing nothing or appears to be in a loop, or the compiler batch job does not reach completion.

If the problem looks like a WAIT state, it is probably a system problem. In that case, follow your local procedures for resolution. If the problem is still unresolved, your set of keywords, so far, would look like this:

```
5686A0100 R1FY LOOP
```

Continue with Chapter 5, "Function Keyword" in topic 5.0.

4.4 C/VSE Documentation Problems

Follow this PUBS keyword procedure when you notice a problem caused by incorrect or missing information in one of the C/VSE published or softcopy documents.

1. If the existing information is wrong, locate the page or pages in the document or the online panel where the problem occurs, and prepare a description of the error and the problem it caused. If it is missing, locate the place where you think it should appear. This information will be required for APAR preparation if no similar problem is found in the software support database.

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

   If the problem is not severe, fill out and mail the Readers' Comments Form (RCF) attached to the back of the publication in question. If the RCF is missing, mail a note to the address shown on the edition notice at the front of this book. You can also send a fax or electronic mail (for details, see "Communicating Your Comments to IBM" at the back of this book). Include the problem description you have developed, along with your name and return address, so that IBM can respond to your comments.

If the problem is severe enough to cause lost time for other users, continue creating your keyword string to determine whether IBM has a record of the problem. If this is a new problem, a severity 3 or 4 documentation (PUBS) APAR will be created.

3. In the case of manuals or softcopy books, use the order number on the cover of the document along with the PUBS keyword as your type-of-failure keyword, but omit the hyphens. For example, if the number on the cover is SC09-2425-00 (*C/VSE Language Reference*), then use SC09242500. Place a forward slash (/) between PUBS and the document number. Your set of keywords would now consist of:

   5686A0100 R1FY PUBS/SC09242500

4. To determine if this documentation problem has already been reported, turn to Chapter 8, "Using the Keyword String as a Search Argument" in topic 8.0.

   If, after searching the IBM software support database, you do not find a matching description, return here to continue.

5. Before discontinuing your database search, you may want to search again, using the following format:

   5686A0100 R1FY PUBS/SC092425**

   In case several levels of the document exist, you can use two asterisks appended to the document number to search for all problems reported for the document rather than only those for a specific release of the document.

6. Go to Chapter 8, "Using the Keyword String as a Search Argument" in topic 8.0.

4.5 Output Problems

Use this procedure when the output appears to be incorrect or missing, but the C/VSE compiler otherwise terminated normally. If the data or records were repeated endlessly, follow the steps under "No Response Problems" in topic 4.3 instead of the INCORROUT procedure to create your keyword string.

1. Use INCORROUT as your type-of-failure keyword.

2. Select a modifier keyword from the following table to describe the type of error in the output. For more information on the use of modifier keywords, see the section Chapter 6, "System-Type Keyword" in topic 6.0.

Modifier Keyword	**Type of Incorrect Output**
MISSING	Some expected output was missing.
DUPLICATE	Some data or records were duplicated, but were not repeated endlessly.
INVALID	The output that appeared was not as expected; that is, the output was bad or incorrect.

3. Select another modifier keyword from the following table to describe the portion of the output where the error occurred.

Modifier Keyword	Portion of Output in Error
AGGREGATE	Structure Maps
INLINE	Inline Report
LIST	Assembler language expansion of source listing
MESSAGE	Diagnostic messages
OBJECT	Machine-language object program
OFFSET	Storage Offset Listing
PPONLY	Preprocessor output (written to SYSPCH)
SOURCE	Source listing
TERM	Progress and diagnostic messages (written to SYSLOG)
XREF	Cross-reference listing

For example, if you think that the compiler has given an incorrect cross-reference listing, your keyword string so far would contain the following:

```
5686A0100 R1FY INCORROUT INVALID XREF
```

4. Continue the diagnostic procedure with Chapter 6, "System-Type Keyword" in topic 6.0.

4.6 Performance Problems

Most performance problems can be related to system tuning and should be handled by system engineers and system programmers. Use this keyword procedure when the performance problem could not be corrected by system tuning and performance is significantly below explicitly stated expectations.

1. Use PERFM as your type-of-failure keyword. For example, your keyword string for performance C/VSE problems might look like this:

```
5686A0100 R1FY PERFM
```

2. Continue the diagnostic procedure with Chapter 6, "System-Type Keyword" in topic 6.0.

5.0 Chapter 5. Function Keyword

For product failures or user problems that occur at LE/VSE run time, see the *LE/VSE Debugging Guide and Run-Time Messages*; for user compile-time program problems, see the *C/VSE User's Guide* for information.

Should the compiler terminate abnormally, the following procedure helps you determine if there are function or language element modifier keywords that you should use in your keyword string. Use this procedure to locate the point of compiler failure and the language element you were using when the failure occurred.

The C/VSE compiler is itself a C/VSE program, and runs with LE/VSE. If an error occurs in the compiler, it is handled by LE/VSE which produces a *traceback* with information about the compiler error. This output goes to the DLBL or TLBL name CEEDUMP (if present) or system logical unit SYSLST (otherwise).

Subtopics:

5.1 How to Find the Function Name in the LE/VSE Traceback

```
CEE5DMP V1 R4.0: Condition processing resulted in the unhandled condition.         05/14/96 1:23:40 PM

Information for enclave main

  Information for thread 8000000000000000

  Traceback:
    DSA Addr  Program Unit  PU Addr   PU Offset  Entry         E Addr    E  Offset   Statement  Status
    00550018  CEEHDSP       00516D30  +00002386  CEEHDSP       00516D30  +00002386              Call
    00552760                01A698B8  +00000000  compdir       01A698B8  +00000000              Exception
    005525B0                01A4FC30  +00003534  scan          01A4FC30  +00003534              Call
    00552520                01A595D8  +00000060  gettoken      01A595D8  +00000060              Call
    00552428                01A4BE80  +0000083C  edcdp         01A4BE80  +0000083C              Call
    00552378                01C8DCA8  +00000000                01C8DCA8  +00000000              Call
    005522C0                01CDD1D8  +0000001A  @@GETFN       01CDD130  +000000C2              Call
    005521E8                005000F8  +000001C0  main          005000F8  +000001C0              Call
    005520D8                01DA7E66  +000000B4  @@MNINV       01DA7E66  +000000B4              Call
    00552018  CEEBBEXT      005046B8  +00000132  CEEBBEXT      005046B8  +00000132              Call


  Condition Information for Active Routines
    Condition Information for  (DSA address 00552760)
      CIB Address: 005504D0
      Current Condition:
        CEE3201S The system detected an operation exception.
      Location:
        Program Unit:  Entry: compdir Statement:  Offset: +00000000
      Machine State:
        ILC..... 0002    Interruption Code..... 0001
        PSW..... 071D1000 80000002
        GPR0..... 005529C8  GPR1..... 00552878  GPR2..... 00552760  GPR3..... 01A6A8F4
        GPR4..... 01C8DE68  GPR5..... 01C965D0  GPR6..... 01C96655  GPR7..... 00000868
  .
  .
  .
```

Figure 2. Sample traceback from LE/VSE

1. Look in the text that follows the word Traceback:.

2. Find the row that contains the word Exception in the Status column.

3. Use the function name found in the Entry column in that row. (Enter it in RETAIN in all capital letters).

If the compiler failed and you received a traceback like the one in Figure 2, you would use COMPDIR as your function keyword. At this point your keyword string would look like this:

```
5686A0100 R1FY COMPDIR
```

5.2 Additional keywords

- If the failure is peculiar to a compiler statement or library function, use the statement or function name as a modifier keyword. For example, if the source of failure was the C/VSE library function ctime(), your keyword string would look like this:

  ```
  5686A0100 R1FY CTIME
  ```

  See Chapter 7, "Modifier Keywords" in topic 7.0 for information about using modifier keywords in your keyword search.

- If the failure appears to be correlated with any particular compile-time option or options, such as SHOWINC, use those options as additional modifier keywords, as in this example:

  ```
  5686A0100 R1FY SHOWINC
  ```

  See Chapter 7, "Modifier Keywords" in topic 7.0 for information about using modifier keywords in your keyword search.

- Continue the diagnosis procedure with Chapter 6, "System-Type Keyword" in topic 6.0.

6.0 Chapter 6. System-Type Keyword

Use VSE as the system-type keyword for C/VSE problems.

The following is an example of a keyword string for C/VSE Version 1 Release 1 with a no-response problem:

```
5686A0100 R1FY LOOP VSE
```

Continue the diagnosis procedure with Chapter 7, "Modifier Keywords" in topic 7.0.

Note: The C/VSE component identifier 5686A0100 already implicitly defines the system type as VSE (since C/VSE only runs under VSE), so the system-type keyword is redundant in the example above. However, when specifying keywords for creating an APAR, you should include VSE as a keyword, because other users searching for the problem might specify VSE without a component ID.

7.0 Chapter 7. Modifier Keywords

You can use one or more modifier keywords in the same keyword string to define the compiler problem. They help to make the search argument more specific. Capitalize all of the letters of the modifier in the keyword string. The various types of modifier keywords include:

- C/VSE
  - Language Elements
  - Listing Control Statements
  - Compile-Time Options

    Select from your compiler listing those compile-time options that you consider significant to the type of failure. The option name itself is the keyword.
  - Message IDs

- LE/VSE library
  - Library functions
  - Link-edit options
  - Run-time options

If the failure appears to be associated with any particular compile-time option or options, such as SHOWINC, use these options as modifier keywords, as in this example:

```
5686A0100 R1FY VSE SHOWINC
```

Continue the diagnostic procedure with Chapter 8, "Using the Keyword String as a Search Argument" in topic 8.0.

8.0 Chapter 8. Using the Keyword String as a Search Argument

Searches against a software support database will be most successful if you follow these rules:

- Spell keywords the way they are spelled in this book. Any variation in spelling may result in an unsuccessful search.
- Include all the appropriate keywords in any discussion with IBM support personnel or in an APAR.

The following explains how to use the keyword string as a search argument against a software support database.

1. Search the software support database, using the full set of keywords you have developed. For example:

   ```
   5686A0100 R1FY ABEND00C4 EDCDP FUNCTION
   ```

   It is usually better not to search on more than three keywords in a single statement. Using the asterisk (*) in a search string can make the steps easier. Using the * in place of the first search parameter means search in the results of the previous search. The actual commands to search on the above string would be:

   ```
   P: 5686A0100 R1FY ABEND00C4
   P: * EDCDP FUNCTION
   ```

   After each search is entered, a message tells you how many matches were made. The message is in the form:

   ```
   THE ABOVE SEARCH ARGUMENT RESULTED IN    19 MATCHES
   ```

2. If the search produces a list of APARs, continue with step 3; otherwise, go to step 6.

3. When your search is complete, eliminate from the list of possible PTFs those that have already been applied to your system.

4. Compare each of the remaining closed APAR descriptions and PTF descriptions with the current failure symptoms.

5. If a match is found, obtain the program temporary fix (PTF), apply it to your system, and exit from this procedure.

6. If the search did not produce a list of APARs, or an APAR description matching the current failure is not found, broaden the search, using the following techniques:

   a. Omit the release level keyword (for example, R1FY) from the search argument, thereby broadening the search to include similar failures on other release levels.

   b. Drop one keyword from the right end of the search argument string.

      By dropping a keyword from the right, you broaden your search while maintaining the relevancy of your search argument string. Perform the search against the software support database, using your shortened search argument string. Repeat this step as necessary, dropping keywords from the right of the string until only the component identifier is left.

7. If a match is not found using the preceding techniques, go to Chapter 9, "Preparing an Authorized Program Analysis Report (APAR)" in

topic 9.0.

9.0 Chapter 9. Preparing an Authorized Program Analysis Report (APAR)

Subtopics:

9.1 APARS and Problem Management Reports (PMRs)

A Problem Management Record (PMR) can be opened after you have done the following:

1. Eliminated user errors as a possible cause of the problem.

2. Followed the diagnostic procedures.

3. You or your local IBM Support Center has been unsuccessful with the keyword search.

If you have IBMLink or some other connection to IBM databases, you can open a PMR yourself. Otherwise, the IBM Support Center may open the PMR after consulting with you on the phone. You may be asked for any of the documentation in the list below. The PMR is used to document your problem and to record the work that the Support Center does on the problem. If IBM concludes that the problem described in the PMR is a problem with the C/VSE product, they will work with you to open an Authorized Program Analysis Report (APAR), so the problem can be fixed. After the APAR is opened and the fix is produced, the description of the problem and the fix will be in the software support database in RETAIN.

9.2 Before Initiating an APAR

1. Contact the IBM Support Center for assistance. You will have been in contact with the IBM Support Center while they were working on the PMR. Be prepared to supply the following information:

    - Customer number and security code
    - PMR number
    - Operating system
    - Operating system release level
    - Current C/VSE compiler maintenance level (PTF list and list of APAR fixes applied)

- The various keyword strings used to search the software support database
- Processor number (model and serial)

2. From the following list, you may be asked to include the applicable C/VSE environmental information with your APAR:

   - Job control statements.
   - Compiler listings, including:
     - Source listing
     - Object listing
     - Storage map
     - Traceback
     - Cross-reference listing

     Use LIST, MAP, SOURCE, XREF, and other options pertinent to the problem.
   - A machine-readable copy of the program causing the problem, including all include files required by the program.
   - A partition or standalone dump, if available.
   - A copy of the compiler, if requested by an IBM representative.
   - A copy of the job control language for unloading the submitted machine-readable tape.
   - Any other data that may help in re-creating the problem.

In addition, a description of the application, the VSE Librarian organization, and the operating instructions or console log may be helpful in reproducing the error. Any listings supplied must be from the C/VSE compilation version that failed.

Table 4 describes how to produce documentation required for submission with the APAR. Additional requirements are explained after the table. Many of these materials may already have been produced in their required format during the formulation of the keyword string. (See "Diagnosis Procedure" in topic 1.1.)

Table 4. Problem Resolution Documentation Requirements

Item	Materials Required	How to Obtain Materials
1	Machine-readable source program	Generated by an IBM-supplied system utility program. The source program should be reduced to the smallest, least complex form that still produces the error.
2	Compiler listings:	
	Source listing	SOURCE option
	Cross-reference listing	XREF option

	Assembler-language expansion	LIST option
3	Compiler termination dump	SYSLST (as directed by IBM support personnel)
4	Partition size, virtual storage map, amount of GETVIS available	JCL or system programmer
5	List of applied PTFs	System programmer
6	Operating instructions	Application programmer
7	Console log	Application programmer
8	Machine-readable JCL for the compile job	JCL or system programmer

---

9.3 Preparation of Material

When submitting material for an APAR to IBM, be sure that the media containing source programs and JCL are carefully packed and clearly identified.

Any magnetic tape submitted must have the following information attached and visible:

1. The PMR number assigned by IBM.

2. A list of data sets on the tape (such as source program, JCL, or interactive environment information).

3. A description of how the tape was made, including the following information:

    a. A full listing of JCL used to produce the machine-readable source. Include the block size, LRECL, and format of each file.

    b. Labeling information used for the volume and its data sets.

    c. The recording mode and density.

    d. The name of the utility program that created each data set.

    e. The record format and block size used for each data set.

10.0 Chapter 10. Problem Identification Worksheet

Component Identification: _____

Release Level: _____

Type of Failure: _____

Module: _____

System Type: _____

Modifiers: _____

_____

_____

Note: Some keywords may not be applicable to all problems.

11.0 Chapter 11. Maintenance

You might need to apply maintenance or service updates to C/VSE periodically. There are two types of formally supported software fixes. One is the program temporary fix (PTF) applied as corrective service or as preventive maintenance. The other is the authorized program analysis report (APAR) fix applied as a code replacement in a corrective maintenance mode.

If you report a problem with C/VSE to your IBM Support Center, you will receive a tape containing one or more APARs or PTFs which have been created to solve your problem.

You may also receive a list of prerequisite APARs or PTFs which should have been applied to your system before applying the current service. These prerequisite APARs or PTFs may relate to C/VSE or any other licensed product you have installed, including VSE/ESA.

You apply service to C/VSE using either the VSE/ESA Interactive Interface or a batch job.

For detailed information about maintaining C/VSE, see the *C/VSE Installation and Customization Guide*.

BIBLIOGRAPHY Bibliography

Subtopics:

BIBLIOGRAPHY.1 IBM C for VSE/ESA Publications

Licensed Program Specifications, GC09-2421

Installation and Customization Guide, GC09-2422

Migration Guide, SC09-2423

User's Guide, SC09-2424

Language Reference, SC09-2425

Diagnosis Guide, GC09-2426

BIBLIOGRAPHY.2 IBM Language Environment for VSE/ESA Publications

Fact Sheet, GC33-6679

Concepts Guide, GC33-6680

Debugging Guide and Run-Time Messages, SC33-6681

Installation and Customization Guide, SC33-6682

Licensed Program Specifications, GC33-6683

Programming Guide, SC33-6684

Programming Reference, SC33-6685

Run-Time Migration Guide, SC33-6687

Writing Interlanguage Communication Applications, SC33-6686

C Run-Time Programming Guide, SC33-6688

C Run-Time Library Reference, SC33-6689

BIBLIOGRAPHY.3 Related Publications

VSE/ESA Version 1 Release 4

Messages and Codes, SC33-6507

VSE/ESA Version 2

Messages and Codes, SC33-6607

Debug Tool for VSE/ESA

User's Guide and Reference, SC26-8797

BIBLIOGRAPHY.4 Softcopy Publications

The following collection kit contains the C/VSE, LE/VSE, and other LE/VSE-conforming language product publications:

VSE Collection, SK2T-0060

You can order these publications from Mechanicsburg through your IBM representative.

INDEX Index

A

B

C

D

E

F

G

I

J

K

L

M

N

O

P

R

S

T

U

V

X

BACK_1 Communicating Your Comments to IBM

IBM C for VSE/ESA
Diagnosis Guide
Release 1

Publication No. GC09-2426-00

If there is something you like--or dislike--about this book, please let us know. You can use one of the methods listed below to send your comments to IBM. If you want a reply, include your name, address, and telephone number. If you are communicating electronically, include the book title, publication number, page number, or topic you are commenting on.

The comments you send should only pertain to the information in this book and its presentation. To request additional publications or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

If you are mailing a readers' comment form (RCF) from a country other than the United States, you can give it to the local IBM branch office or IBM representative for postage-paid mailing.

- If you prefer to send comments by mail, use the RCF at the back of this book.

- If you prefer to send comments by FAX, use this number:

  - United States and Canada: 416-448-6161

  - Other countries: (+1)-416-448-6161

- If you prefer to send comments electronically, use the network ID listed below. Be sure to include your entire network address if you

wish a reply.

- Internet: torrcf@vnet.ibm.com
- IBMLink: toribm(torrcf)
- IBM/PROFS: torolab4(torrcf)
- IBMMAIL: ibmmail(caibmwt9)

COMMENTS Readers' Comments -- We'd Like to Hear from You

IBM C for VSE/ESA
Diagnosis Guide
Release 1

Publication No. GC09-2426-00

Overall, how satisfied are you with the information in this book?

Legend:

1 Very satisfied
2 Satisfied
3 Neutral
4 Dissatisfied
5 Very dissatisfied

	1	2	3	4	5
Overall satisfaction					

How satisfied are you that the information in this book is:

	1	2	3	4	5
Accurate					
Complete					
Easy to find					
Easy to understand					
Well organized					
Applicable to your tasks					

Please tell us how we can improve this book:

IBM Canada Ltd. Laboratory
Information Development
2G/345/1150/TOR
1150 EGLINTON AVENUE EAST
NORTH YORK ONTARIO CANADA M3C 1H7

Name . . . . . _____
Company or Organization _____
Address . . . . . _____
_____
_____
Phone No. . . . . _____

IBM Library Server Print Preview

DOCNUM = GC09-2426-00
DATETIME = 10/01/96 05:12:12
BLDVERS = 1.2
TITLE = C/VSE V1R1 Diagnosis Guide
AUTHOR =

PATH = /home/webapps/epubs/htdocs/book
