IBM COBOL for VSE/ESA

Diagnosis Guide

Release 1

SC26-8528-00

IBM COBOL for VSE/ESA

Diagnosis Guide

Release 1

SC26-8528-00

Note!

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

First Edition (April 1995)

This edition applies to Version 1 Release 1 of IBM COBOL for VSE/ESA, Program Number 5686-068, and to all subsequent releases and modifications until otherwise indicated in new editions. Changes are made periodically to this publication; consult the latest *IBM System/390, 370, 30xx, 4300, and 9370 Processors Bibliography* for current information on this product.

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

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

IBM Corporation, Department J58
P.O. Box 49023
San Jose, California, 95161-9023
U.S.A.

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

Notices

References in this publication to IBM products or services do not imply that they will be available everywhere IBM operates, nor that only IBM products or services can be used. Functionally equivalent products or services that do not infringe legal rights held by IBM can be used instead. Operation with products or services other than those expressly designated by IBM is your responsibility.

IBM may have patents or pending patent applications covering subject matter described herein. This document neither grants nor implies any license or immunity under any IBM or third-party patents, patent applications, trademarks, copyrights, or other similar rights, or any right to refer to IBM in any marketing activities. Other than responsibilities assumed via the Agreement for Purchase of IBM Machines and the Agreement for IBM Licensed Programs, IBM assumes no responsibility for any infringement of third-party rights that may result from use of the subject matter disclosed in this publication or from the manufacture, use, lease, or sale of machines or programs described herein.

Licenses under utility patents held by IBM are available on reasonable and nondiscriminatory terms. IBM does not grant licenses under its appearance design patents. Direct licensing inquiries in writing to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594, U.S.A..

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. **This disclaimer does not apply in the United Kingdom or elsewhere if inconsistent with local law.**

Programming Interface

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

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

Trademarks

The following terms, denoted by an asterisk (*) on their first occurrences in this publication, are trademarks of the IBM Corporation in the United States and/or other countries:

BookManager	IBM	IBMLink
Language Environment	System/370	System/390
VSE/ESA		

About This Book

This book tells you how to diagnose failures in the COBOL/VSE compiler only. Because COBOL/VSE uses Language Environment* for VSE/ESA* as its run-time environment, consult the *LE/VSE Diagnosis Guide* to diagnose product failures you encounter in the run-time environment.

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 COBOL/VSE, or by an error in the logic of the application program. In the cases where a user error is the cause of the problem, consult the *COBOL/VSE Programming Guide* and *LE/VSE Debugging Guide and Run-Time Messages* for more information.

This book helps you 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 APAR" on page 20 explains how to prepare an Authorized Program Analysis Report (APAR).

This book uses the term "COBOL 85 Standard" to refer to the combination of the following standards:

- ISO 1989:1985, Programming languages - COBOL
- ISO 1989/Amendment 1, Programming Languages - COBOL - Amendment 1: Intrinsic function module
- X3.23-1985, American National Standard for Information Systems - Programming Language - COBOL
- X3.23a-1989, American National Standard for Information Systems - Programming Language - Intrinsic Function Module for COBOL

Note that the two ISO standards are identical to the American National standards.

Who Should Read This Book

This book is for system programmers, application programmers, and IBM* Support personnel who are involved in COBOL/VSE product diagnosis. Prerequisite knowledge for using this book is

- A general understanding of your operating system
- Some knowledge of the COBOL/VSE language and options

Publications Provided with COBOL/VSE

Publications provided with the COBOL/VSE product include the following:

Figure 1. IBM COBOL for VSE/ESA Publications

Task	Publication	Order number
Evaluation and Planning	*General Information* *Migration Guide* *Installation and Customization Guide*	GC26-8068 GC26-8070 SC26-8071
Programming	*Programming Guide* *Language Reference* *Reference Summary*	SC26-8072 SC26-8073 SX26-3834
Diagnosis	*Diagnosis Guide*	SC26-8528
Warranty	*Licensed Program Specifications*	GC26-8069

General Information

Contains high-level information designed to help you evaluate the COBOL/VSE product. This book describes new compiler and language features, application development with LE/VSE, and product support for industry standards.

Migration Guide

Contains detailed migration and compatibility information for current users of DOS/VS COBOL and VS COBOL II who wish to migrate to, or reuse existing applications on, COBOL/VSE. This book also describes several migration aids or tools to help you plan a migration path for your installation.

Installation and Customization Guide

Provides information you will need in order to install and customize the COBOL/VSE product. Detailed planning information includes:

- System and storage requirements for COBOL/VSE
- Information about changing compiler option defaults during installation
- Information for installing the product in shared storage

Programming Guide

Contains guidance information for writing and compiling application programs using COBOL/VSE, including information on the following topics:

- Programming using new product features, such as intrinsic functions
- Processing techniques for VSAM and SAM files
- Debugging techniques using compiler options and listings
- Nested programming techniques
- Subsystem considerations

Language Reference

Provides syntax and semantic information about the COBOL language as implemented by IBM, including rules for writing source programs, and descriptions of IBM language extensions. This book is meant to be used in conjunction with the *COBOL/VSE Programming Guide*, which provides programming task-oriented information.

Reference Summary
Contains a convenient summary of the COBOL/VSE language syntax—including new intrinsic functions—as well as syntax for compiler options, compiler-directing statements, and the COBOL/VSE reserved word list.

Diagnosis Guide
Provides instructions for diagnosing failures in the COBOL/VSE compiler product that are not caused by user error. This book will help you construct a keyword string that allows you or IBM Service to search the product failure database for previously documented problems and appropriate corrections.

Licensed Program Specifications
Contains a product description and product warranty information for the COBOL/VSE compiler.

Language Environment for VSE/ESA Publications

Other publications useful for developing applications with COBOL/VSE include the following publications provided with the LE/VSE product:

Figure 2. IBM Language Environment for VSE/ESA Publications

Task	Publication	Order number
Evaluation and Planning	*Fact Sheet* *Concepts Guide* *Installation and Customization Guide*	GC26-8062 GC26-8063 SC26-8064
Programming	*Programming Guide* *Debugging Guide and Run-Time Messages* *Reference Summary*	SC26-8065 SC26-8066 SX26-3835
Diagnosis	*Diagnosis Guide*	SC26-8060
Warranty	*Licensed Program Specifications*	GC26-8061

Fact Sheet
Provides a brief overview and description of LE/VSE.

Concepts Guide
Provides a detailed overview of program models and intended architecture for LE/VSE, the common run-time environment.

Installation and Customization Guide
Contains information needed to plan for installing and customizing the LE/VSE product.

Programming Guide.
Provides detailed information on the following topics:

- Directions for linking and running programs that use LE/VSE services
- Information on storage management, run-time message handling, and condition handling models

- Callable services and run-time options, and how to use them
- Instructions for writing programs that use interlanguage communication (ILC)

This book also contains language-specific run-time information.

Debugging Guide and Run-Time Messages
Provides detailed information on debugging techniques and services. Provides a listing of run-time messages and their explanations, as well as abend codes.

Reference Summary
Contains a convenient summary of the IBM Language Environment for VSE/ESA.

Diagnosis Guide
Provides instructions for diagnosing failures in the LE/VSE product that are not caused by user error. This book will help you construct a keyword string that allows you or IBM Service to search the product failure database for previously documented problems and appropriate corrections.

Licensed Program Specifications
Contains a product description and warranty information.

Chapter 1. Introduction

Failures in the IBM COBOL for VSE/ESA compiler can be described through the use of *keywords*. A keyword is a word or abbreviation used to describe one aspect of a product failure. A set of keywords, called a *keyword string*, can be used to describe the failure in detail. The procedures in this section will help you construct a keyword string that describes what you currently know about the compiler failure. For failures that occur in the IBM Language Environment for VSE/ESA run-time product, see *LE/VSE Diagnosis Guide*; for user compile-time problems, see *COBOL/VSE Programming Guide*; and for user run-time errors, see *LE/VSE Debugging Guide and Run-Time Messages*.

After it is constructed, the keyword string is used as a search argument against an IBM software support database, such as the Software Support Facility (SSF). The database contains keyword and text information describing all current problems reported through Authorized Program Analysis Reports (APARs). The database also contains Program Temporary Fixes (PTFs) associated with the APARs. 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 electronic link with IBM Service, 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 result in a fuller description of the problem and possibly identify a correction or circumvention. Such a search may produce several matches to previously reported problems. Thus, the diagnostician should review each error description carefully to determine if the problem description in the database matches the user's 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.

Keyword Use

In building a keyword string, the first keyword always identifies the failing component. The component identification for COBOL/VSE should be the compiler identifier. A search of the software support database with this single keyword would locate all problems reported for the compiler. Each additional keyword added to the keyword string narrows the scope of the search argument 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 circumstances make it difficult to follow the instructions for selecting a particular keyword, omit that keyword to avoid incorrectly identifying the problem. In general, if you contact IBM, you will be asked to identify your problem with a full set of keywords, as described here.

Depending upon the COBOL/VSE failure, a keyword string can contain some or all of these:

- Component identification
- Release level
- Type of failure
- Name of the module that failed
- Name of the system you were operating under at the time of failure
- One or more modifier keywords, depending on the type of failure

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

Figure 3 on page 3 shows the keyword string generation process for each type of failure.

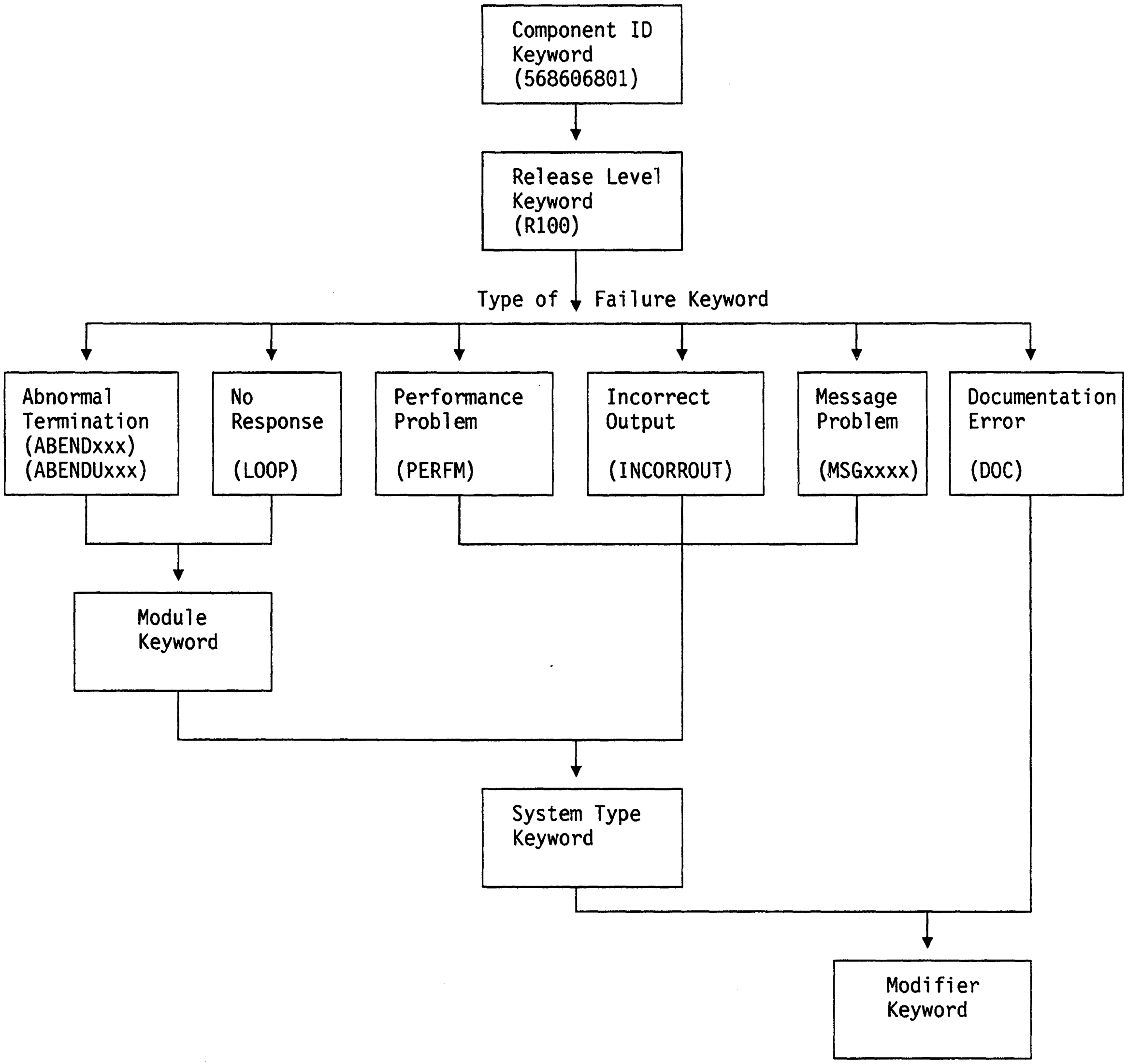
Component ID Keyword (568606801)
Release Level Keyword (R100)
Type of Failure Keyword
Abnormal Termination (ABENDxxx) (ABENDUxxx)
No Response (LOOP)
Performance Problem (PERFM)
Incorrect Output (INCORROUT)
Message Problem (MSGxxxx)
Documentation Error (DOC)
Module Keyword
System Type Keyword
Modifier Keyword

Figure 3. COBOL/VSE – Problem Identification Using Keywords

Using the Problem Identification Worksheet

You can use the Chapter 10, "Problem Identification Worksheet" on page 24 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.

Diagnosis Procedure

This procedure is designed to gather the diagnostic information required for developing a keyword string to search the software support database. It describes options that, if specified, 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 against the database fails to locate a fix for your problem.

Use the following procedure only if the problem is occurring at compile time. If the problem is in the LE/VSE product, refer to *LE/VSE Diagnosis Guide* for diagnostic information.

1. Determine if the program has been changed since it was last compiled successfully. If it has, examine the changes. If the error is occurring in the changed code, note the change that caused the error. If possible, retain copies of both the original and the changed programs, to submit with an APAR, if it is required.

   Determine if the operating system environment has been changed. If it has, examine the changes. If the error cannot be resolved by examination of the changes, have the information available when you call the IBM Support Center.

   Determine if COBOL/VSE maintenance has been applied. If it has, determine if the error was due to a particular program temporary fix (PTF) and call the IBM Support Center with this information.

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. Pay attention to warning messages (W-level messages). Here are a couple of points to remember:

   - COBOL/VSE compiler messages are prefixed by IGY.
   - Message prefixes identify the system/subsystem that issued the error. If the prefix is anything but IGY, consult the appropriate system/subsystem messages book.

3. 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:

   - Pinpoint the problem
   - Distinguish between an error in the application program and an error in COBOL/VSE
   - Choose keywords that best describe the error

4. Specify the following compiler options, in addition to the options originally specified, and recompile the program:

   COMPILE (only if requested by IBM support personnel)
   DUMP (only if requested by IBM support personnel)
   FLAG (I)
   LIST
   MAP
   XREF
   VBREF
   SOURCE

These options produce maximum diagnostic information, which will help you diagnose product errors. See *COBOL/VSE Programming Guide* for more information on how to use these options.

5. If the error symptoms change, return to step 2 on page 4.
6. Record the sequence of events that led to the error condition. This information may be useful in developing a keyword string, and will be needed if an APAR is required.
7. Begin developing the keyword string; start with the procedures in Chapter 2, "Component Identification Keyword" on page 6.

Chapter 2. Component Identification Keyword

This procedure shows what to specify in the component identification keyword. The component identification keyword is always the first keyword placed in the search argument string. It comes from the COBOL/VSE compiler program number and identifies the area within the software support database that contains APARs for COBOL/VSE.

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

1. Use **568606801** as the component identification keyword. This number is the COBOL/VSE Compiler product identifier, 5686-068, with 01 added.
2. If service tapes have been applied to the licensed program, the tape level of the last service tape applied should be noted. See your system programmer for the current service level of your COBOL/VSE compiler. Although the service tape level is not used in the keyword string, it will be useful when reviewing APARs selected during the keyword search.
3. Continue the diagnostic procedure with Chapter 3, "Release Level Keyword" on page 7.

Chapter 3. Release Level Keyword

Use the following procedure to identify the specific release level of COBOL/VSE under which you were operating when the failure occurred.

1. Locate the version, release, and modification level line at the top of the first page of your latest compiler output listing for the failing program. The version level line contains the current product identification data in the following format:

   ```
   PP 5686-068 IBM COBOL for VSE/ESA v.r.m mm/dd/yy ...
   ```

   where v specifies the current version and r.m specifies the current ***r***elease and ***m***odification number, and mm/dd/yy specifies the date of the current release and modification. The date and time of processing and the page number are also found on the current release level line. The version line for Version 1.1.0 of COBOL/VSE would be as follows:

   ```
   PP 5686-068 IBM COBOL for VSE/ESA 1.1.0...
   ```

2. Specify the release level keyword, using the format **Rrm0**:

R	Constant
r	Release level number
m	Modification level number
0	Constant

   If Version 1.1.0 appears at the top of your listing, the release level keyword would be R100.

   The following is an example of a partial keyword string, consisting of the component identification and release level keywords.

   ```
   568606801 R100
   ```

3. Continue the diagnostic procedure with Chapter 4, "Type-of-Failure Keyword" on page 8.

Chapter 4. Type-of-Failure Keyword

Various types of failure specific to the compiler may occur in the COBOL/VSE licensed program. For product failures that occur during run time, consult *LE/VSE Diagnosis Guide*; for user compile-time problems, see *COBOL/VSE Programming Guide* for information; and for user run-time errors, see *LE/VSE Debugging Guide and Run-Time Messages*.

Read the following table and select the type of failure that best describes the problem with the compiler. Then go to the associated keyword procedure listed in this table for instructions on how to complete the keyword string for that type of failure. If more than one of the keywords in the following table describes the problem you are experiencing, use the keyword that appears first in the table.

Figure 4. Types of COBOL/VSE Failures

Type-of-Failure	Symptom	Procedure
Abnormal Termination	The compiler has ended abnormally with a completion code that indicates that an abend has occurred.	"Abnormal Termination."
Message Problems	The compiler issues an inappropriate or invalid error message.	"Message Problems" on page 10.
No Response from the Compiler	Either an unexpected program suspension has occurred or the job has not completed in batch mode.	"No Response from the Compiler" on page 11.
COBOL/VSE Documentation Problems	Information in one of the COBOL/VSE publications or softcopy documents is incorrect or missing.	"COBOL/VSE Documentation Problems" on page 12.
Output Problems	The output from the compiler is missing or invalid.	"Output Problems" on page 13.
Performance Problems	The performance of a COBOL/VSE compilation is degraded.	"Performance Problems" on page 14.

Abnormal Termination

The compiler can end abnormally with either a system or user abend. In general, follow the procedure described in "ABENDxxx Procedure (System)" on page 9 for system abends and those described in "ABENDUxxxx Procedure (User)" for user abends.

Note: Do not use either of the abnormal termination procedures if termination was forced because too much time was spent in a wait state or in an endless loop. In this situation, refer to the procedure for "No Response from the Compiler" on page 11.

ABENDUxxxx Procedure (User)

Use this procedure if COBOL/VSE ends abnormally with a **user abend** code. A user abend will be accompanied by the message:

```
IGYRC5244 COBOL/VSE Compilation failed with completion code yyyy
```

where yyyy is a four digit user abend code.

1. Replace the xxxx of ABENDUxxxx with the user abend code (yyyy). For example, for a user abend of 1122, specify ABENDU1122 as your keyword.

2. If you received a message, you can use the message ID as a modifier. See Chapter 7, "Modifier Keywords" on page 18 for a description of modifier keywords.

   For example, if the compiler failed with a user abend of 1122, and you received message IGYRC5122-U, your keyword string would look something like this:

   ```
   568606801 R100 ABENDU1122 IGYRC5122-U
   ```

3. Continue the diagnostic procedure with Chapter 5, "Module Keyword" on page 15.

ABENDxxx Procedure (System)

Use this procedure if the compiler ends abnormally with a **system** abend code. A system abend is accompanied by the message:

```
IGYRC5108  COBOL compiler terminating:
Uncorrectable program interrupt condition.
```

To create the keyword, use the procedure described below.

The COBOL/VSE compiler STXIT PC exit has trapped the program check interrupt and the DUMP option of the compiler will produce a diagnostic message. Use relevant words from the text of the abend message as keywords. 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 keyword string for COBOL/VSE Release 1 when the system abend is accompanied by the above message is:

```
568606801 R100 PROTECTION EXCEPTION
```

Use the name of the current COBOL/VSE compiler phase as a modifier keyword. (See Chapter 7, "Modifier Keywords" on page 18 for more information.) Look near the end of the compiler listing for the line that appears in the same form as the following sample and use the name following the word "PHASE."

```
CURRENT COBOL COMPILER PHASE: IGYCPGEN (STORAGE LOC: 0002B000)
```

In this instance, your keyword string would appear as follows:

```
568606801 R100 PROTECTION EXCEPTION IGYCPGEN
```

Continue with Chapter 6, "System-Type Keyword" on page 17.

If you search the IBM support database and 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-abend keyword. Check the following list of common program checks for a match on the keywords used. If you find a match, you can use the equivalent table keyword. For example, if the program check is a Protection exception, specify ABEND0C4 as your type-of-failure keyword.

Figure 5. MVS System Abend Codes

Program Check	Abend Code	Keyword
Operation Exception	0C1	ABEND0C1
Privileged operation exception	0C2	ABEND0C2
Execute exception	0C3	ABEND0C3
Protection exception	0C4	ABEND0C4
Addressing exception	0C5	ABEND0C5
Specification exception	0C6	ABEND0C6
Data exception	0C7	ABEND0C7
Fixed-point overflow exception	0C8	ABEND0C8
Fixed-point divide exception	0C9	ABEND0C9

No Message from the Compiler

If the compiler has ended abnormally without a termination message, a keyword can be created using the VSE Cancel Code. The Cancel Code can be retrieved from the symptom record in the system dump. The Cancel Code is used in the format ABENDSxx00, where xx is the Cancel Code. For example, if the Cancel Code is 21, your keyword would be ABENDS2100.

Message Problems

The COBOL/VSE compiler issues messages prefixed with IGY. Messages with other prefixes are issued by the IBM Language Environment for VSE/ESA run-time environment, or by operating systems/subsystems and access methods, and should not be addressed as COBOL/VSE product problems. See the messages books for the appropriate components.

DO NOT USE THIS PROCEDURE FOR THE FOLLOWING SITUATIONS:

- You received an abnormal termination message.
- You received the following message:

  ```
  IGYRC5108  COBOL compiler terminating:
  Uncorrectable program interrupt condition.
  ```

For these problems, see “Abnormal Termination” on page 8.

This (Message Problems) procedure is used for any one of the following conditions:

- A message is issued under a set of conditions that should not have caused it to be issued.
- A message contains invalid data or is missing data.
- The message

  ```
  IGYLS5018 Error message number "?" was specified in
  E-TEXT, but was not found in the messages modules.
  ```

  or

  ```
  IGYDI5018 Error message number "?" was specified in
  E-TEXT, but was not found in the messages modules.
  ```

  was received while compiling a COBOL/VSE program.

Use the following procedure to construct the MSGx keyword:

1. If you received message IGYLS5018 or IGYDI5018, complete this step; otherwise, go to step 3.

   If you received message number IGYLS5018 or IGYDI5018, replace the x in MSGx with IGY5018, as in the following example:

   ```
   568606801 R100 MSGIGY5018
   ```

2. Proceed with Chapter 6, "System-Type Keyword" on page 17.
3. If the message prefix is IGY, complete this step. Replace the *x* of MSGx with the message identifier. For example, if the compiler message received is IGYDS1082-S, the MSGx keyword would be MSGIGYDS1082. The hyphen and severity code are not included in the MSGx keyword. Your set of keywords, so far, would look something like this:

   ```
   568606801 R100 MSGIGYDS1082
   ```

4. Proceed with Chapter 6, "System-Type Keyword" on page 17

No Response from the Compiler

Use this keyword procedure for any of the following conditions:

- The compiler seems to be doing nothing or is doing something repetitively.
- The compiler does not reach completion in batch mode.

If the problem looks like a WAIT state and the compiler is not waiting for input from the console, it is probably a system problem. In that case, follow your local procedures for resolution. Otherwise, perform the following LOOP procedure.

1. Your set of keywords, so far, would look something like this:

   ```
   568606801 R100 LOOP
   ```

2. Continue with Chapter 5, "Module Keyword" on page 15

COBOL/VSE Documentation Problems

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

1. Locate the page or pages in the document, or the online panel for a softcopy document, where the problem occurs, and prepare a description of the error and the problem it caused. 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, submit a Reader's Comment Form (RCF). You will find an RCF attached to the back of the publication in question. As an option, you can fax the RCF to IBM using the fax number printed on the RCF. If the RCF is missing, send a note to the address shown on the edition notice for 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. Should this be a new problem, you will be asked to submit a severity-3 or -4 documentation (DOC) APAR.

3. Use the order number on the cover of the document along with the DOC keyword as your type-of-failure keyword, but omit the hyphens. Leave a single space between DOC and the document number. If the number following the last hyphen has only one digit, it must be preceded by a zero. For example, if the order number is SC26-8073-0 (*COBOL/VSE Language Reference*), use SC26807300. Your keyword string would look something like this:

   ```
   568606801 R100 DOC SC26807300
   ```

4. Search the IBM software support database to determine if this documentation problem has already been reported. If, after searching the database, you do not find a matching description, **return here to continue.** To search the database, turn to Chapter 8, "Using the Keyword String as a Search Argument" on page 19.

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

   ```
   568606801 R100 DOC SC268073**
   ```

   In case several levels of the document exist, you can use two asterisks added to the end of 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" on page 19.

Output Problems

Use this procedure when the output appears to be incorrect or missing, but the compiler otherwise ended normally. If the data or records were repeated endlessly, follow the steps under "No Response from the Compiler" on page 11 instead of this "Output Problems" 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 section Chapter 7, "Modifier Keywords" on page 18.

Figure 6. Incorrect Output Modifier Keywords

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 in which the error occurred.

Figure 7. Output Error Location Keywords

Modifier Keyword	Portion of Output in Error
SOURCE	Source listing
OBJECT	Machine-language object program
XREF	Cross-reference listing
STAT	Statistics and error listing
MAP	Data Division map, global tables, literal pools
LIST	Assembler language expansion of source listing, global tables, literal pools
MESSAGE	Diagnostic Message
TERM	Progress and diagnostic messages on the console
OFFSET	Condensed Procedure Division listing
VBREF	Cross-reference of verbs used in the source program

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

```
568606801 R100 INCORROUT INVALID XREF
```

4. Continue the diagnostic procedure with Chapter 6, "System-Type Keyword" on page 17.

Performance Problems

Most performance problems can be related to system tuning and should be handled by system engineers and system programmers. You might want to contact your IBM system engineer who can use ASKQ to retrieve recommendations for improving product performance.

Use the following 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 problems might look like this:

   ```
   568606801 R100 PERFM
   ```

2. Continue the diagnostic procedure with Chapter 6, “System-Type Keyword” on page 17.

Chapter 5. Module Keyword

This procedure will help you determine if there are module and/or verb modifier keywords that you should use in your keyword string. Use this procedure to locate the point of compiler failure and/or the verb you were using when the failure occurred. For product failures that occur during run time, see *LE/VSE Diagnosis Guide*; for user compile-time program problems, see *COBOL/VSE Programming Guide* for information; and for user run-time problems, see *LE/VSE Debugging Guide and Run-Time Messages.*

If the failure is occurring on an ESA* system, the addresses referred to below can be either 24 or 31 bits, depending on the AMODE bit (bit 32) in the PSW.

1. Locate the PSW to help determine your module keyword.
2. Locate the address of the point of failure in the associated dump. Scan backward (high memory address to low) in the EBCDIC area to locate the copyright character string that starts within the first 8 bytes of the module, or scan forward (low to high) to locate the patch area character string near the end of the module. Both areas contain the module name. See Figure 8 for more details.

```
*                        .00..IGY*
*PCNTL  95.0605686.068 .C. COPYRI*
*GHT IBM CORP. 1988.1995LICENSED *
*MATERIAL . PROGRAM PROPERTY OF I*
*BM.................O...........D*
*.....H....K..D............ABRT50*
*17K.........K..S...O M..K..P..K.*

   :

*................................*
*.............O.N................*
*................................*
*..PATCH AREA . IGYPCNTL 95.060..*
*................................*
*................................*
*................................*
```

Figure 8. Example of Dump Area with Compiler Module Name (IGYPCNTL)

To locate the module to use in your keyword string:

a. Scan forward from the address of failure to locate a PATCH AREA character string. (If a copyright character string is located immediately after the address of failure, this means you are at the beginning of the failing module. Use the module name from the copyright statement and continue with step 3 on page 16.)

b. If, in your forward scan, you find a copyright statement at any location other than the one immediately following the point of failure and prior to locating the PATCH AREA, your scan has crossed a module boundary into a module beyond the one containing the address of failure. In this case, you should go to step 2d on page 16.

c. If you locate the PATCH AREA, use the module name following the words PATCH AREA as your modifier keyword.

If the compiler abended with a protection exception and you received a dump like the one in Figure 8, then at this point, your keyword string would look something like this:

```
568606801 R100 PROTECTION EXCEPTION IGYPCNTL
```

Continue the diagnosis procedure with step 3.

d. If you were unable to find the module name, use the address that was in register 14 at the time of failure as the address of failure and return to step 2a on page 15. (Register 14 should contain an address within the calling routine.) If you have tried using both the PSW and register 14 addresses to search the dump and have been unable to locate the module, continue this diagnosis procedure.

3. If the compiler failure is peculiar to a COBOL/VSE verb, use the verb as a modifier keyword. For example, if the source of failure was the COBOL/VSE verb ADD, your keyword string would look something like this:

   ```
   568606801 R100 PROTECTION EXCEPTION ADD
   ```

   See Chapter 7, "Modifier Keywords" on page 18 for information about using modifier keywords in your keyword search.

4. If the compiler failure appears to be correlated with any particular compiler option or options, such as XREF, use those options as additional modifier keywords, as in this example:

   ```
   568606801 R100 PROTECTION EXCEPTION XREF
   ```

   See Chapter 7, "Modifier Keywords" on page 18 for information about using modifier keywords in your keyword search.

5. Continue the diagnosis procedure with the Chapter 6, "System-Type Keyword" on page 17.

Chapter 6. System-Type Keyword

System-type keywords indicate which system you were operating under when the COBOL/VSE compiler failed.

1. Use VSE as your first system-type keyword.
2. Use ESA as your next system-type keyword. For instance, your keyword search might look like this:

   ```
   568606801 R100 PROTECTION EXCEPTION VSE ESA
   ```

3. Continue the diagnostic procedure with Chapter 7, "Modifier Keywords" on page 18.

Chapter 7. Modifier Keywords

One or more modifier keywords may be used in the same keyword string to define the compiler problem. Additional modifiers help to make the search argument more specific. Use the capitalized spelling of the modifier in the keyword string. The various types of modifier keywords include:

- COBOL/VSE Reserved Words
  - Verbs
  - Statement and statement options
  - Listing control statement

  See Appendix A, "COBOL/VSE Reserved Words" on page 25, for a list of the reserved words. The reserved word itself is the keyword.

- Compiler options

  Select from your compiler listing those compiler options that you consider significant to the type of failure. See Appendix B, "Compiler Options" on page 42, for a list of the compiler options. The option name itself is the keyword.

- Message ID

  If you receive a compile-time message, you can use the ID as an additional modifier. For example, assume that you received this message:

  ```
  IGYRC5122-U  A recursive entry in the COBOL abort module occurred.
  ```

  Then you could use IGYRC5122-U as an additional modifier.

Continue the diagnostic procedure with Chapter 8, "Using the Keyword String as a Search Argument" on page 19.

Chapter 8. Using the Keyword String as a Search Argument

The following explains how to use the keyword string as a search argument against a software support database. The search can be performed by calling an IBM Support Center or using your electronic link with IBM, if available.

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

- Use only the keywords given in this book.
- 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.

In order to search the support database, the diagnostician should perform the following steps:

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

   ```
   568606801 R100 PROTECTION EXCEPTION IGYPCNTL VSE ESA FUNCTION
   ```

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 APAR fixes those that have already been applied to your system.
4. Compare each of the remaining closed APAR descriptions with the current failure symptoms.
5. If a match is found, apply the program temporary fix (PTF) to your system and exit 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, R100) 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. The diagnostic procedures directed you to construct the keyword string with the most significant keywords listed first. By dropping a keyword from the right, you eliminate the least significant keyword, thereby broadening your search while maintaining the relevance of your search argument string. Perform the search against the software support database, using your shortened search argument string. Repeat this step as necessary.

7. If a match is not found using the preceding techniques, go to Chapter 9, "Preparing an APAR" on page 20.

Chapter 9. Preparing an APAR

Before proceeding to prepare an Authorized Program Analysis Report (APAR), make sure that:

- You have followed the diagnosis procedure
- You have eliminated user errors as a source for the problem
- The keyword search was unsuccessful

Then open a Problem Management Record (PMR).

Opening a PMR

If you have IBMLink or some other connection to IBM databases, you may open a PMR yourself; otherwise, you can call the IBM Support Center and request that they open the PMR for you.

The PMR is used to document your problem and to record the work done on the problem by members of the IBM Support Center or the IBM Change Team. After analyzing the problem, the Support Center may recommend that an APAR be initiated.

Initiating an APAR

In order to initiate an APAR, you need to complete the following steps.

1. Contact the IBM Support Center for assistance Be prepared to supply the following information:

   - Customer number and security code
   - PMR number
   - Operating system
   - Operating system release level
   - Current COBOL/VSE 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 COBOL/VSE environmental information with your APAR:

   - Job control statements
   - Any special EXECs
   - Compiler listings, including:
     - Source listing
     - Object listing
     - Storage map
     - Cross-reference listing

   Use LIST, MAP, SOURCE, XREF, and other options pertinent to the problem.

- Machine-readable copy of the program causing the problem, including all copy members required by the program
- A dump on tape, if available, or if the use of the DUMP option was requested by an IBM representative
- The compiler on tape if requested by an IBM representative
- Machine-readable reserved word list, if modifications were made
- Hard copy of the job control language (JCL) for unloading the submitted machine-readable tape
- Any other data that may help in recreating the problem

Any listings supplied must be from the COBOL/VSE compilation version that failed.

Figure 9 on page 22 describes how to produce documentation required for submission with the APAR. Additional requirements are explained after the figure. Many of these materials may already have been produced in their required format during the formulation of the keyword string. (See "Diagnosis Procedure" on page 4.)

Figure 9. Problem Resolution Documentation Requirements

Item	Materials Required	How to Obtain Materials
1	Machine-readable source program	See "Machine-Readable Source."
	Machine-readable reserved word table, if reserved words have been changed.	Librarian Backup of IGYCxxxx (Where 'xxxx' is the reserved word table identifier)
2	Compiler listings:	
	Source listing	SOURCE option
	Cross-reference listing	XREF option
	Data Division listing	MAP option
	Assembler-language expansion	LIST option
3	Compiler termination dump	DUMP option and VSE JCL OPTION DUMP or PARTDUMP (as directed by IBM support personnel)
4	Partition/virtual storage size	JCL or system programmer
5	List of applied PTFs for COBOL/VSE and others as requested	System programmer
6	Operating instructions or console log	Application programmer
7	VSE control statements with OPTION LOG	See "Batch Environment"

Machine-Readable Source

The source program must be supplied in machine-readable form, using a system utility program supplied by IBM. The source program should be reduced to the smallest, least complex form that still produces the error.

Batch Environment

- Supply the JCL listings used to run the program, including an expanded list of the cataloged procedures used.
- Run LSERV to list the System Standard labels and Partition Standard labels in effect.
- If there is a large amount of JCL, it must be supplied in machine-readable form, preferably on unlabeled magnetic tape.
- Issue a LISTIO for the partition to list the system logical unit assignments active.
- Issue a MAP command on the VSE console to show the partition size and GETVIS allocation.
- Run LIBR to list the contents of the SDL (System Directory List - phases that have been placed in the SVA).

Submitting the APAR Documentation

When submitting material for an APAR to IBM, be sure that the media containing source programs, job stream data, or interactive environment information are carefully packed and clearly identified.

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

1. The APAR number assigned by IBM
2. A list of files on the tape (for example, source program, JCL)
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 files
    c. The recording mode and density
    d. The name of the utility program that created each file
    e. The record format and block size used for each file

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.

Appendix A. COBOL/VSE Reserved Words

This list identifies all reserved words in the COBOL/VSE product.

- Words marked under **Compiler-Directing** are compiler-directing words and are flagged with an S-LEVEL message if used as user-defined names.
- Words marked under **Non-COBOL/VSE** are COBOL 85 Standard Reserved Words that are NOT implemented in COBOL/VSE Release 1 and are flagged with an S-LEVEL message if used as user-defined names.
- Words marked under **CODASYL** are reserved for future development and are flagged with an I-LEVEL message.

Note: The contents of the reserved word table can be changed by using the WORD compiler option. See *COBOL/VSE Programming Guide* for details on how to specify an alternate reserved word table.

Figure 10 (Page 1 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
ACCEPT			
ACCESS			
ACQUIRE			
ADD			
ADDRESS			
ADVANCING			
AFTER			
ALL			
ALLOWING			X
ALPHABET			
ALPHABETIC			
ALPHABETIC-LOWER			
ALPHABETIC-UPPER			
ALPHANUMERIC			
ALPHANUMERIC-EDITED			
ALSO			
ALTER			
ALTERNATE			
AND			
ANY			
APPLY			
ARE			
AREA			

Figure 10 (Page 2 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
AREA-VALUE			
AREAS			
ARITHMETIC			X
ASCENDING			
ASSIGN			
AT			
AUTHOR			
AUTO			
AUTO-SKIP			
AUTOMATIC			
B-AND			X
B-EXOR			X
B-LESS			X
B-NOT			X
B-OR			X
BACKGROUND-COLOR			
BACKGROUND-COLOUR			
BACKWARD			
BASIS	X		
BEEP			
BEFORE			
BEGINNING			
BELL			
BINARY			
BIT			X
BITS			X
BLANK			
BLINK			
BLOCK			
BOOLEAN			X
BOTTOM			
BY			
CALL			
CANCEL			
CBL	X		
CD		X	
CF		X	
CH		X	

Figure 10 (Page 3 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
CHAIN			
CHAINING			
CHANGED			
CHARACTER			
CHARACTERS			
CLASS			
CLOCK-UNITS		X	
CLOSE			
COBOL			
CODE			
CODE-SET			
COL			
COLLATING			
COLOR			
COLUMN		X	
COM-REG			
COMMA			
COMMAND-LINE			
COMMIT			X
COMMITMENT			
COMMON			
COMMUNICATION		X	
COMP			
COMP-X			
COMP-0			
COMP-1			
COMP-2			
COMP-3			
COMP-4			
COMP-5			X
COMP-6			X
COMP-7			X
COMP-8			X
COMP-9			X
COMPUTATIONAL			
COMPUTATIONAL-X			
COMPUTATIONAL-0			
COMPUTATIONAL-1			

Figure 10 (Page 4 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
COMPUTATIONAL-2			
COMPUTATIONAL-3			
COMPUTATIONAL-4			
COMPUTATIONAL-5			X
COMPUTATIONAL-6			X
COMPUTATIONAL-7			X
COMPUTATIONAL-8			X
COMPUTATIONAL-9			X
COMPUTE			
CONFIGURATION			
CONNECT			X
CONSOLE			
CONTAINED			X
CONTAINS			
CONTENT			
CONTINUE			
CONTROL	X	X	
CONTROL-AREA			
CONTROLS		X	
CONVERTING			
COPY	X		
CORR			
CORRESPONDING			
COUNT			
CRT			
CRT-UNDER			
CURRENCY			
CURRENT			X
CURSOR			
CYCLE			X
DATA			
DATE			
DATE-COMPILED			
DATE-WRITTEN			
DAY			
DAY-OF-WEEK			
DB			X
DB-ACCESS-CONTROL-KEY			X

Figure 10 (Page 5 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
DB-DATA-NAME			X
DB-EXCEPTION			X
DB-FORMAT-NAME			
DB-RECORD-NAME			X
DB-SET-NAME			X
DB-STATUS			X
DBCS			
DE		X	
DEBUG-CONTENTS			
DEBUG-ITEM			
DEBUG-LINE			
DEBUG-NAME			
DEBUG-SUB-1			
DEBUG-SUB-2			
DEBUG-SUB-3			
DEBUGGING			
DECIMAL-POINT			
DECLARATIVES			
DEFAULT			X
DELETE	X		
DELIMITED			
DELIMITER			
DEPENDING			
DESCENDING			
DESTINATION		X	
DETAIL		X	
DISABLE		X	
DISCONNECT			X
DISK			
DISPLAY			
DISPLAY-1			
DISPLAY-2			X
DISPLAY-3			X
DISPLAY-4			X
DISPLAY-5			X
DISPLAY-6			X
DISPLAY-7			X
DISPLAY-8			X

Figure 10 (Page 6 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
DISPLAY-9			X
DIVIDE			
DIVISION			
DOWN			
DROP			
DUPLICATE			X
DUPLICATES			
DYNAMIC			
EGCS			
EGI		X	
EJECT	X		
ELSE			
EMI		X	
EMPTY			X
EMPTY-CHECK			
ENABLE		X	
END			
END-ACCEPT			
END-ADD			
END-CALL			
END-COMPUTE			
END-DELETE			
END-DISABLE			X
END-DIVIDE			
END-ENABLE			X
END-EVALUATE			
END-IF			
END-MULTIPLY			
END-OF-PAGE			
END-PERFORM			
END-READ			
END-RECEIVE		X	
END-RETURN			
END-REWRITE			
END-SEARCH			
END-SEND			X
END-START			
END-STRING			

Figure 10 (Page 7 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
END-SUBTRACT			
END-TRANSCEIVE			X
END-UNSTRING			
END-WRITE			
ENDING			
ENTER			
ENTRY			
ENVIRONMENT			
EOP			
EQUAL			
EQUALS			X
ERASE			X
ERROR			
ESCAPE			
ESI		X	
EVALUATE			
EVERY			
EXACT			X
EXCEEDS			X
EXCEPTION			
EXCESS-3			
EXCLUSIVE			X
EXEC			
EXECUTE			
EXHIBIT			
EXIT			
EXTEND			
EXTERNAL			
EXTERNALLY-DESCRIBED-KEY			
FALSE			
FD			
FETCH			X
FILE			
FILE-CONTROL			
FILE-ID			
FILLER			
FINAL		X	
FIND			X

Figure 10 (Page 8 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
FINISH			X
FIRST			
FIXED			
FOOTING			
FOR			
FOREGROUND-COLOR			
FOREGROUND-COLOUR			
FORM			X
FORMAT			X
FREE			X
FROM			
FULL			
FUNCTION			
GENERATE		X	
GET			X
GIVING			
GLOBAL			
GO			
GOBACK			
GREATER			
GROUP		X	
HEADING		X	
HIGH-VALUE			
HIGH-VALUES			
HIGHLIGHT			
I-O			
I-O-CONTROL			
ID			
IDENTIFICATION			
IF			
IN			
INDEX			
INDEX-1			X
INDEX-2			X
INDEX-3			X
INDEX-4			X
INDEX-5			X
INDEX-6			X

Figure 10 (Page 9 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
INDEX-7			X
INDEX-8			X
INDEX-9			X
INDEXED			
INDIC			
INDICATE		X	
INDICATOR			
INDICATORS			
INITIAL			
INITIALIZE			
INITIATE		X	
INPUT			
INPUT-OUTPUT			
INSERT	X		
INSPECT			
INSTALLATION			
INTO			
INVALID			
IS			
JAPANESE			
JUST			
JUSTIFIED			
KANJI			
KEEP			X
KEPT			
KEY			
KEYBOARD			
LABEL	X		
LAST		X	
LD			X
LEADING			
LEFT			
LEFT-JUSTIFY			
LENGTH			
LENGTH-CHECK			
LESS			
LIKE			
LIMIT		X	

Figure 10 (Page 10 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
LIMITS		X	
LINAGE			
LINAGE-COUNTER			
LINE			
LINE-COUNTER		X	
LINES			
LINKAGE			
LOCALLY			X
LOCK			
LOW-VALUE			
LOW-VALUES			
MANUAL			
MEMBER			X
MEMORY			
MERGE			
MESSAGE		X	
MODE			
MODIFIED			
MODIFY			X
MODULES			
MORE-LABELS			
MOVE			
MULTIPLE			
MULTIPLY			
NAME			
NATIVE			
NEGATIVE			
NEXT			
NO-ECHO			
NO			
NORMAL			X
NOT			
NULL			
NULLS			
NUMBER		X	
NUMERIC			
NUMERIC-EDITED			
OBJECT-COMPUTER			

Figure 10 (Page 11 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
OCCURS			
OF			
OFF			
OMITTED			
ON			
ONLY			X
OPEN			
OPTIONAL			
OR			
ORDER			
ORGANIZATION			
OTHER			
OUTPUT			
OVERFLOW			
OWNER			X
PACKED-DECIMAL			
PADDING			
PAGE			
PAGE-COUNTER		X	
PALETTE			
PARAGRAPH			X
PASSWORD			
PERFORM			
PF		X	
PH		X	
PIC			
PICTURE			
PLUS		X	
POINTER			
POSITION			
POSITIVE			
PRESENT			X
PREVIOUS			
PRINT-SWITCH			
PRINTER			
PRINTER-1			
PRINTING		X	
PRIOR			X

Figure 10 (Page 12 of 17). Reserved Words

Reserved Word	**Compiler-Directing**	**Non-COBOL/VSE**	**CODASYL**
PROCEDURE			
PROCEDURE-POINTER			
PROCEDURES			
PROCEED			
PROCESS			
PROCESSING			
PROGRAM			
PROGRAM-ID			
PROMPT			
PROTECTED			X
PURGE		X	
QUEUE		X	
QUOTE			
QUOTES			
RANDOM			
RANGE			
RD		X	
READ			
READY	X		
REALM			X
RECEIVE		X	
RECONNECT			X
RECORD			
RECORD-NAME			X
RECORDING			
RECORDS			
REDEFINES			
REEL			
REFERENCE			
REFERENCES			
RELATION			X
RELATIVE			
RELEASE			
RELOAD	X		
REMAINDER			
REMOVAL			
RENAMES			

Figure 10 (Page 13 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
REPEATED			X
REPLACE	X		
REPLACING			
REPORT		X	
REPORTING		X	
REPORTS		X	
REQUIRED			
RERUN			
RESERVE			
RESET	X		
RETAINING			X
RETRIEVAL			X
RETURN			
RETURN-CODE			
REVERSE-VIDEO			
REVERSED			
REWIND			
REWRITE			
RF		X	
RH		X	
RIGHT			
RIGHT-JUSTIFY			
ROLLBACK			X
ROLLING			
ROUNDED			
RUN			
SAME			
SCREEN			
SD			
SEARCH			
SECTION			
SECURE			
SECURITY			
SEGMENT		X	
SEGMENT-LIMIT			
SELECT			
SEND		X	
SENTENCE			

Figure 10 (Page 14 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
SEPARATE			
SEQUENCE			
SEQUENTIAL			
SERVICE	X		
SESSION-ID			X
SET			
SHARED			X
SHIFT-IN			
SHIFT-OUT			
SIGN			
SIZE			
SKIP1	X		
SKIP2	X		
SKIP3	X		
SORT			
SORT-CONTROL			
SORT-CORE-SIZE			
SORT-FILE-SIZE			
SORT-MERGE			
SORT-MESSAGE			
SORT-MODE-SIZE			
SORT-RETURN			
SOURCE		X	
SOURCE-COMPUTER			
SPACE			
SPACE-FILL			
SPACES			
SPECIAL-NAMES			
STANDARD			
STANDARD-1			
STANDARD-2			
STANDARD-3			X
STANDARD-4			X
START			
STARTING			
STATUS			
STOP			
STORE			X

Figure 10 (Page 15 of 17). Reserved Words

Reserved Word	**Compiler-Directing**	**Non-COBOL/VSE**	**CODASYL**
STRING			
SUB-QUEUE-1		X	
SUB-QUEUE-2		X	
SUB-QUEUE-3		X	
SUB-SCHEMA			X
SUBFILE			
SUBPROGRAM			
SUBTRACT			
SUM			
SUPPRESS			
SWITCH			
SWITCH-1			
SWITCH-2			
SWITCH-3			
SWITCH-4			
SWITCH-5			
SWITCH-6			
SWITCH-7			
SWITCH-8			
SYMBOLIC			
SYNC			
SYNCHRONIZED			
TABLE		X	
TALLY			
TALLYING			
TAPE			
TENANT			X
TERMINAL		X	
TERMINATE		X	
TEST			
TEXT		X	
THAN			
THEN			
THROUGH			
THRU			
TIME			
TIMEOUT			X
TIMES			

Figure 10 (Page 16 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
TITLE	X		
TO			
TOP			
TRACE	X		
TRAILING			
TRAILING-SIGN			
TRANSACTION			
TRANSCEIVE			X
TRUE			
TYPE		X	
UNDERLINE			
UNEQUAL			X
UNIT			
UNLOCK			
UNSTRING			
UNTIL			
UP			
UPDATE			X
UPON			
USAGE			
USAGE-MODE			X
USE	X		
USER			
USING			
VALID			X
VALIDATE			X
VALUE			
VALUES			
VARIABLE			
VARYING			
WAIT			X
WHEN			
WHEN-COMPILED			
WITH			
WITHIN			X
WORDS			
WORKING-STORAGE			
WRITE			

Figure 10 (Page 17 of 17). Reserved Words

Reserved Word	Compiler-Directing	Non-COBOL/VSE	CODASYL
WRITE-ONLY			
ZERO			
ZERO-FILL			
ZEROES			
ZEROS			
<			
<=			
+			
*			
**			
-			
/			
>			
>=			
=			

Appendix B. Compiler Options

Note: See Chapter 7, “Modifier Keywords” on page 18 for information on how and why compiler options could be used as keywords.

COMPILER OPTION

ADATA
ADV
APOST
AWO
BUFSIZE

COMPILE
CMPR2
CURRENCY
DATA
DBCS

DECK
DUMP
DYNAM
EXIT
FASTSRT

FLAG
FLAGMIG
FLAGSAA
FLAGSTD
LANGUAGE

LIB
LINECOUNT
LIST
MAP
NAME

NOADATA
NOADV
NOAWO
NOCOMPILE
NOCMPR2

NOCURRENCY
NODECK
NODBCS
NODUMP
NODYNAM

NOEXIT
NOFASTSRT
NOFLAG
NOFLAGMIG
NOFLAGSAA

NOFLAGSTD
NOLIB
NOLIST
NOMAP
NONAME

NONUMBER
NOOBJECT
NOOFFSET
NOOPTIMIZE
NORENT

COMPILER OPTION

NOSEQUENCE
NOSOURCE
NOSSRANGE
NOTEST
NOTERMINAL

NOVBREF
NOWORD
NOXREF
NOZWB
NUMBER

NUMPROC(MIG)
NUMPROC(NOPFD)
NUMPROC(PFD)
OBJECT
OFFSET

OPTIMIZE
OUTDD
QUOTE
RENT
RMODE

SEQUENCE
SOURCE
SIZE
SPACE
SSRANGE

TEST
TERMINAL
TRUNC(BIN)
TRUNC(OPT)
TRUNC(STD)

VBREF
WORD
XREF(SHORT)
XREF(FULL)
ZWB

Bibliography

Language Environment Publications

IBM Language Environment for VSE/ESA

Fact Sheet, GC26-8062

Concepts Guide, GC26-8063

Installation and Customization Guide, SC26-8064

Programming Guide, SC26-8065

Debugging Guide and Run-Time Messages, SC26-8066

Diagnosis Guide, SC26-8060

Licensed Program Specifications, GC26-8061

Reference Summary, SX26-3835

LE/VSE-Conforming Language Product Publications

IBM COBOL for VSE/ESA

General Information, GC26-8068

Migration Guide, GC26-8070

Installation and Customization Guide, SC26-8071

Programming Guide, SC26-8072

Language Reference, SC26-8073

Reference Summary, SX26-3834

Diagnosis Guide, SC26-8528

Licensed Program Specifications, GC26-8069

IBM PL/I for VSE/ESA

Fact Sheet, GC26-8052

Programming Guide, SC26-8053

Language Reference, SC26-8054

Licensed Program Specifications, GC26-8055

Migration Guide, SC26-8056

Installation and Customization Guide, SC26-8057

Diagnosis Guide, SC26-8058

Reference Summary, SX26-3836

Compile-Time Messages and Codes, SC26-8059

Related Publications

VSE/ESA Version 1 Release 4

Messages and Codes Vol.1 & 2, SC33-6507

VSE/ESA Version 2 Release 1

Messages and Codes Vol.1 & 2, SC33-6607

Softcopy Publications

These collections contain the COBOL/VSE and LE/VSE-conforming language product publications:

VSE Collection, SK2T-0060

Application Development Collection, SK2T-1237

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

Index

We'd Like to Hear from You

IBM COBOL for VSE/ESA
Diagnosis Guide
Release 1

Publication No. SC26-8528-00

Please use one of the following ways to send us your comments about this book:

- Mail—Use the Readers' Comments form on the next page. If you are sending the form from a country other than the United States, give it to your local IBM branch office or IBM representative for mailing.
- Fax—Use the Readers' Comments form on the next page and fax it to this U.S. number: 800-426-7773.
- Electronic mail—Use one of the following network IDs:
  - IBMMail: USIB2VVG at IBMMAIL
  - IBMLink: COBPUBS at STLVM27
  - Internet: COMMENTS@VNET.IBM.COM

  Be sure to include the following with your comments:

  - Title and publication number of this book
  - Your name, address, and telephone number if you would like a reply

Your comments should pertain only to the information in this book and the way the information is presented. To request additional publications, or to comment on other IBM information or the function of IBM products, please give your comments to your IBM representative or to your IBM authorized remarketer.

IBM may use or distribute your comments without obligation.

Readers' Comments

IBM COBOL for VSE/ESA
Diagnosis Guide
Release 1

Publication No. SC26-8528-00

How satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Technically accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□
Grammatically correct and consistent	□	□	□	□	□
Graphically well designed	□	□	□	□	□
Overall satisfaction	□	□	□	□	□

Please tell us how we can improve this book:

May we contact you to discuss your comments? □ Yes □ No

Name

Address

Company or Organization

Phone No.

Readers' Comments
SC26-8528-00

Fold and Tape — **Please do not staple** — Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Department J58
International Business Machines Corporation
PO BOX 49023
SAN JOSE CA 95161-9945

Fold and Tape — **Please do not staple** — Fold and Tape

Readers' Comments

IBM COBOL for VSE/ESA
Diagnosis Guide
Release 1

Publication No. SC26-8528-00

How satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Technically accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□
Grammatically correct and consistent	□	□	□	□	□
Graphically well designed	□	□	□	□	□
Overall satisfaction	□	□	□	□	□

Please tell us how we can improve this book:

May we contact you to discuss your comments? □ Yes □ No

Name

Address

Company or Organization

Phone No.

Fold and Tape **Please do not staple** Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Department J58
International Business Machines Corporation
PO BOX 49023
SAN JOSE CA 95161-9945

Fold and Tape **Please do not staple** Fold and Tape

IBM®

Program Number: 5686-086

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber

IBM COBOL for VSE/ESA Publications

SC26-8528	Diagnosis Guide
GC26-8068	General Information
GC26-8069	Licensed Program Specification
SC26-8073	Language Reference
GC26-8070	Migration Guide
SC26-8072	Programming Guide
SC26-8071	Installation and Customization Guide
SX26-3834	Reference Summary