

IBM Compiler and Library for REXX on zSeries



Diagnosis Guide

Version 1 Release 4

IBM Compiler and Library for REXX on zSeries



Diagnosis Guide

Version 1 Release 4

Note!

Before using this information and the product it supports, be sure to read the general information under Appendix B, "Notices", on page 31.

Third Edition, August 2003

This edition applies to version 1 release 4 of IBM Compiler for REXX on zSeries (product number 5695-013) and the IBM Library for REXX on zSeries (product number 5695-014), and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SH19-8179-01.

© **Copyright International Business Machines Corporation 1989, 2003. All rights reserved.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About This Book	v
How This Book Is Organized	v
How to Read the Syntax Notation	vi
How to Find Service Help	vi
How to Send Your Comments	vi

Chapter 1. Problem Identification Worksheet	1
--	----------

Chapter 2. Introduction	3
Problem Description	3
Keyword Searching	3
Common Errors	4
Using the Problem Identification Worksheet	5
The Diagnosis Procedures	5

Chapter 3. Building the Search Argument	7
Include the Component Identifier	7
Identify the Release Level	7
Identify the Release Level for Compilation Problems	8
Identify the Release Level for Library Problems	8
Identify the Service Level	9
Identify the Type of Failure	11
ABEND Procedure	12
DOC Procedure	13
INCORROUT Procedure	14
MSG Procedure	15
PERFM Procedure	16
WAIT/LOOP Procedure	17
Identify the Module	18

Chapter 4. Searching for a Solution	21
Search for a Solution	21
Narrowing the Search	21
Identifying REXX Language Elements	22
Identifying Compiler Options	22
Identifying Commands	23
Identifying Cataloged Procedures (z/OS)	23

Identifying Cataloged Procedures (VSE/ESA)	23
Identifying ISPF Panels (z/OS)	23

Chapter 5. Preparing an APAR	25
Initiating an APAR	25
Gathering Supporting Documentation	25
Submitting the APAR Documentation	26
Submitting Hardcopy	26
Submitting Tapes or Diskettes	26

Appendix A. Obtaining Interphase Compiler Dumps	29
--	-----------

Appendix B. Notices	31
Programming Interface Information	33
Trademarks	33

List of Abbreviations	35
--	-----------

Related Publications	37
IBM Compiler and Library for REXX on zSeries Publications	37
Other IBM Publications	37
ISPF Publications	37
Learning REXX	37
REXX Reference	37
TSO/E and MVS/ESA™ Publications	38
OpenEdition® Publication	38
VM/SP Publications	38
VM/XA SP Publications	38
VM/ESA Publications	38
VSE/ESA Publication	38
C Publication	38
CMS Publications	38
z/VM Publications	38
z/OS Publications	38
OS/390 Publications	39

Index	41
------------------------	-----------

About This Book

This book is intended to help system programmers and IBM service personnel to diagnose problems encountered when using the:

- IBM Compiler for REXX on zSeries® (referred to as the Compiler)
- IBM Library for REXX on zSeries (referred to as the Library)
- IBM Library for REXX in REXX/VSE (also referred to as the Library)

You should be familiar with system diagnostic techniques. Before you use this book, make sure that the suspected failure was *not* caused by incorrect usage of the IBM Compiler for REXX on zSeries, the IBM Library for REXX on zSeries, the IBM Library for REXX in REXX/VSE, or by an error in the logic of the REXX program. See the *IBM Compiler and Library for REXX on zSeries: User's Guide and Reference* for information on using Compiler options.

It is assumed that you are familiar with the REXX language and with the operating system under which you compile or run your programs:

- z/OS® or OS/390® with Time Sharing Option Extensions (TSO/E)
- CMS on Virtual Machine/Extended Architecture (VM/XA), Virtual Machine/Enterprise System Architecture (VM/ESA®), or z/VM®

Notes:

1. Machine mode 370 is no longer supported.
 2. Only CMS Release 6 and subsequent releases are supported.
- Virtual Storage Extended/Enterprise System Architecture (VSE/ESA™) with REXX/VSE

For information on tracing your program under:

- z/OS, see the *TSO/E REXX/MVS™: Reference* or the corresponding z/OS documentation.
- z/VM, see the corresponding z/VM CMS documentation.
- VSE/ESA, see the corresponding VSE/ESA documentation.

How This Book Is Organized

This book contains a sequence of step-by-step procedures that help you build a string of keywords that describe a problem, to use that string to search a database for a solution to or a bypass for the problem, and to supply supporting documentation for an authorized program analysis report (APAR) if the problem has not previously been reported.

A worksheet is provided at the front of this book to help you collect the appropriate information that describes the problem.

The appendix provides further details about dumps.

How to Read the Syntax Notation

The notation used to define the command syntax in this book is as follows:

- A symbol (word) in boldface, such as **CEXEC**, denotes a keyword.
- Words in italics, such as *options-list*, denote variables or collections of variables.
- The brackets [and] delimit optional parts of the commands.
- The logical OR character | separates choices within brackets.

How to Find Service Help

For the latest information on how to get service help and how to exchange dumps via NetView[®] FTP, refer to APAR II11865. It provides:

- Additional service information and mailing addresses
- Information on how to send documentation and data sets via NV/FTP to the German REXX Compiler & Library Change Team in Boeblingen (BB) SWSD Lab. The RAP security feature of NetView FTP is used to get access to the Boeblingen Laboratory.

How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other REXX documentation:

- Visit our home page at:
<http://www.ibm.com/software/awdtools/rexx/>
There you can access the Internet Online Form where you can enter comments and send them.
- Send your comments by e-mail to swsdid@de.ibm.com. Be sure to include the name of the book, the part number of the book, the version and release of REXX, and, if applicable, the specific location of the text you are commenting on (for example, a page number or table number).
- Fill out one of the forms at the back of this book and return it by mail, by fax, or by giving it to an IBM representative. The mailing address is on the back of the Readers' Comments form. The fax number is +49-(0)7031-16-4892.
- If you have any questions or comments, send an e-mail to:
REXXHELP@VNET.IBM.COM
- z/OS or z/VM customers can also use the IBM Service Center to raise a PMR against:
RETAIN queue: WTR EXX,245

Specify the program ID of your IBM Compiler and Library for REXX on zSeries installation:

569501303 (Compiler under z/OS)
569501403 (Library under z/OS)
569501304 (Compiler under z/VM)
569501404 (Library under z/VM)

Chapter 1. Problem Identification Worksheet

Complete the blanks and circle the appropriate selections below. Note that some keywords are not applicable to all problems.

Component Identifier:

(for details refer to "Include the Component Identifier" on page 7)

569501303

Compiler under z/OS

569501403

Library under z/OS

569501304

Compiler under z/VM

569501404

Library under z/VM

568605802

IBM Library for REXX in REXX/VSE Version 1

568606612

Library under VSE/ESA with versions of REXX/VSE after Version 1

Release Level:

(for details refer to "Identify the Release Level" on page 7)

R130 z/OS and z/VM

R140 z/OS and z/VM

RDC7 VSE/ESA with REXX/VSE Version 1

R55I VSE/ESA with versions of REXX/VSE after Version 1

Service Level:

(The service level is not part of the keyword string, for details refer to "Identify the Service Level" on page 9.)

Type of Failure:

(for details refer to "Identify the Type of Failure" on page 11)

- ABEND (for details refer to "ABEND Procedure" on page 12)
- DOC (for details refer to "DOC Procedure" on page 13)
- INCORROUT (for details refer to "INCORROUT Procedure" on page 14)
- MSG (for details refer to "MSG Procedure" on page 15)
- PERFM (for details refer to "PERFM Procedure" on page 16)
- WAIT/LOOP (for details refer to "WAIT/LOOP Procedure" on page 17)

Module:

(for details refer to "Identify the Module" on page 18)

Modifiers:

(for details refer to “Narrowing the Search” on page 21)

Chapter 2. Introduction

The procedures in this book help you to systematically diagnose a problem encountered when using the:

- IBM Compiler for REXX on zSeries (the Compiler)
- IBM Library for REXX on zSeries (the Library)
- IBM Library for REXX in REXX/VSE (also referred to as the Library)

It also shows how to describe the problem using a string of keywords.

However, before you try these procedures, you should try to run your program with the interpreter. If the problem does not occur with the interpreter, continue with these instructions.

Problem Description

While you build the keyword string, you collect information that helps you to describe a failure in either the Compiler or the Library. A keyword is a word or abbreviation that describes one aspect of a problem.

The keyword string for a particular problem identifies:

- The failing product and component
- The type of failure
- Additional information, depending on the type of failure, such as:
 - Diagnostic information contained in a message
 - The name of the module that was in control at the time of the failure

The use of keywords ensures that identical program errors are described in the same way.

Keyword Searching

If you have installed Information/Access, a feature of the IBM licensed program Information/System, Program Number 5735-OZS, you can access an abstract of the information in this software support database. You can, then, search for previously recorded product problems before calling the IBM Support Center.

The IBM Support Center has access to this software support database, which contains records of known problems and corrections. Your IBM Support Center representative uses your keyword string as the search argument when searching the database. For further details, see "Search for a Solution" on page 21. The next step depends on the result of the search:

- If a similar problem description is found in the database, you are directed to a solution or to a bypass for the problem. This is usually a program temporary fix (PTF).
- If the problem has not been previously reported, the IBM Support Center might ask you to investigate the problem further and to gather more diagnostic information. Using this information, the Support Center prepares an authorized program analysis report (APAR).

Common Errors

Make sure that the problem is not caused by any of the following errors:

- Under z/OS:
 - If the IBM Library for REXX on zSeries is not in the link pack area (LPA), the LINKLIST concatenation, or the STEPLIB DD statement, the following failure occurs:

```
CSV003I REQUESTED MODULE EAGRTPRC NOT FOUND
CSV003I REQUESTED MODULE EAGRTXLD NOT FOUND
CSV003I REQUESTED MODULE EAGRTXVH NOT FOUND
+IRX0158E The run time processor EAGRTPRC could not be found.
```
 - If you do not use the correct stubs, a program that runs with REXX compiled and link-edited code may end abnormally. Check in which environment the program is supposed to run. You can use the multi-purpose stub to link-edit and package programs compiled under z/OS. For more information refer to the *IBM Compiler and Library for REXX on zSeries: User's Guide and Reference*.
 - Check the different behavior when running under IRXJCL and IKJEFT01. For more information refer to the *TSO/E REXX/MVS Reference*.
- Under z/VM:
 - Module A overlaid by module B

If your program is in the form of a module (for example, module A) and it calls another module (for example, module B) module B might overlay your program in storage. This occurs if, for example, both modules are loaded at the default starting address. The failure occurs when module B tries to return control to your program.

To determine whether an overlay caused the failure, recompile the program, creating a compiled EXEC, and recreate the circumstances under which the failure occurred. If the problem disappears, the failure was almost certainly caused by a module overlay. In this case, either continue to run the program as a compiled EXEC or explicitly specify a different starting address when loading your module. If the problem persists, the failure has a different cause, and you should follow the diagnosis procedures in this book.
 - Return code -3 at runtime

If you get a return code of -3 when you invoke your program, it usually means that the program was not found. However, it can alternatively mean that the IBM Library for REXX on zSeries product was not found. So, if you get this return code when the program is available, make the Library available—either in a discontinuous saved segment (DCSS) or on disk.
 - Object modules generated with stub code terminate abnormally when they are run under z/VM.
- Under VSE/ESA:
 - If the IBM Library for REXX in REXX/VSE is not loaded in the shared virtual area (SVA), the following failure occurs:

```
1U5nt PROGRAM NOT FOUND
```

The following error messages are described in the *IBM Compiler and Library for REXX on zSeries: User's Guide and Reference*, they occur when using the:

- Alternate Library:
 1. **EAGALT0300E Error 3 running compiled ATESTALT EXEC, line 0: Program is unreadable**

Under z/OS and VSE/ESA, this message is always followed by a secondary message.

2. EAGALT0304I The program cannot run using the Alternate Library
 - REXX Library
1. EAGREX0300E Error 3 running compiled LIBLVLOW EXEC, line 0:
Program is unreadable
Under z/OS and VSE/ESA, this message is always followed by a secondary message.
2. EAGREX0303I Level of IBM REXX Library too low

Using the Problem Identification Worksheet

You can use the Chapter 1, “Problem Identification Worksheet”, on page 1 to help you construct and record a keyword string. When you identify a keyword associated with the software problem, circle the appropriate one or record it in the space provided.

The Diagnosis Procedures

The procedures in the rest of this book explain how to build and use a keyword string.

You can most accurately diagnose the problem if you follow these rules:

- Develop a full set of keywords.
- Spell keywords exactly as they appear in this book.
- Give all the available information, and include all the appropriate keywords when you discuss the problem with IBM support personnel.

z/OS users

When an abend occurs, a symptom dump is displayed automatically. Before starting the diagnosis, get the following:

- A symptom dump
- A console log
- A dump (SYSUDUMP or SYSMDUMP), if it is available

Modify the SYSTERM DD statement and allocate a data set to it.

Optionally, use the session manager and save the data to a data set.

z/VM users

Enter the **cp spool** command to start spooling your console. This ensures that you can get a hardcopy of any information that appears on your screen during the diagnosis procedures.

For a description of the **spool** command, see the corresponding z/VM CMS documentation.

VSE/ESA users

Before starting the diagnosis, get the following:

- A console log
- A dump

If // STDOPT DUMP=YES or PART is specified in the permanent job control options during system initialization, a dump is taken automatically when an abend occurs.

If no dump was taken automatically, add a // OPTION DUMP statement to the failing job. If // OPTION SYSDUMP is specified and the dump sublibrary is large enough, the dump output is routed to the dump sublibrary, otherwise it is written to the data set specified for SYSLST.

Chapter 3. Building the Search Argument

To build a keyword string, you will need the Chapter 1, “Problem Identification Worksheet”, on page 1 to record pertinent information. Work through all the procedures in the specified order, starting with “Include the Component Identifier”. If you contact the IBM Support Center for assistance, you will be asked to provide the full set of keywords to identify the problem.

Include the Component Identifier

Use the component identifier as the first keyword to identify the Library that contains APARs for the IBM Compiler for REXX on zSeries, the IBM Library for REXX on zSeries, or the IBM Library for REXX in REXX/VSE. This Library is in the software support database.

If you experience a problem during compilation, use the Compiler component identifier:

- 569501303 under z/OS
- 569501304 under z/VM

If you experience a problem at runtime, use the Library component identifier:

- 569501403 under z/OS
- 569501404 under z/VM
- 568606612 under VSE/ESA

Your z/OS keyword string could now look like this

For the Compiler:	569501303
For the Library:	569501403

Your z/VM keyword string could now look like this

For the Compiler:	569501304
For the Library:	569501404

Your VSE/ESA keyword string could now look like this

For the Library:	568606612
------------------	-----------

Continue with “Identify the Release Level”.

Identify the Release Level

Use the release-level keyword to identify the release of the IBM Compiler for REXX on zSeries, the IBM Library for REXX on zSeries, or the IBM Library for REXX in REXX/VSE that was running when the failure occurred. To determine the release and modification level of the Compiler or Library follow the appropriate procedure in this section, depending on whether the problem occurred during compilation or at runtime.

Identify the Release Level for Compilation Problems

1. You can obtain the release level of the Compiler from the latest compiler listing for the failing program or from the compiled program file. Use whichever method is most convenient for you:

- **In the compiler listing**, the product identification data is at the top of every page, in the following format, where *r* is the current release number, *m* is the modification level of the Compiler, and *xxxxxxx* is the APAR level:

```
IBM Compiler for REXX on zSeries r.m LVL xxxxxxx
```

For example, if no PTFs have been installed, the product identification data for Release 4 Modification Level 0 of the IBM Compiler for REXX on zSeries is as follows:

```
IBM Compiler for REXX on zSeries 4.0 LVL -NONE--
```

- **In the compiled program file**, the product identification data is in a character string near the beginning of the file, in the following format, where *r* is the current release number, *m* is the modification level of the Compiler, and *xxxxxxx* is the APAR level:

```
IBM Compiler for REXX on zSeries r.m LVL xxxxxxx Time: 16:38:23
```

For example, the product identification data for Release 4 Modification Level 0 of the IBM Compiler for REXX on zSeries is as follows:

```
IBM Compiler for REXX on zSeries 4.0 LVL -NONE-- Time: 16:38:23
```

2. Specify the release-level keyword in the following format, where *r* is the current release number and *m* is the modification level:

R1*m*

For example, if you have Release 4 Modification Level 0, the release-level keyword is R140.

Your keyword string could now look like this

```
For z/OS: 569501303 R140  
For z/VM: 569501304 R140
```

Go to “Identify the Service Level” on page 9.

Identify the Release Level for Library Problems

Note: In this section, always replace the string *xxx* with:

- LIB, if you are dealing with the Library
- ALT, if you are dealing with the Alternate Library
- **Under z/OS**, browse the EAGRT*xxx* load module and search for the product identification data by entering on the command line:

```
f ' EAGRTxxx'
```

The product identification data has the following format, where *r* is the current release number and *m* is the modification level of the Library:

```
EAGRTxxx r.m
```

For example, the product identification data for Release 4 Modification Level 0 of the IBM Library for REXX on zSeries is as follows:

EAGRTLIB 4.0

Specify the release-level keyword in the following format, where *r* is the current release number and *m* is the modification level:

R1*r*m

For example, if you have Release 4 Modification Level 0, the release-level keyword is R140.

- **Under z/VM**, you can either:
 - Browse the EAGRTxxx MODULE, if it is not NUCXLOADed.
 - Follow this procedure:
 1. Issue the command NUCXMAP EAGRTPRC
 2. Note the entry address shown, for example: 3B5000
 3. Display the beginning of the module, for example: DISPLAY t3B5000.50

In both cases, the product identification data is shown in the following format, where *r* is the release number and *m* is the modification level of the Library:

EAGRTxxx *r.m*

For example, the product identification data for Release 4 Modification Level 0 of the IBM Library for REXX on zSeries is as follows:

EAGRTLIB 4.0

Specify the release-level keyword in the following format, where *r* is the current release number and *m* is the modification level:

R1*r*m

For example, if you have Release 4 Modification Level 0, the release-level keyword is R140.

- **Under VSE/ESA**, the release-level keyword is:
 - RDC7** For the IBM Library for REXX in REXX/VSE Version 1
 - R55I** For the Library under VSE/ESA with versions of REXX/VSE after Version 1

For more information, see the VSE/ESA documentation.

Your keyword string could now look like this

For z/OS:	569501403 R140
For z/VM:	569501404 R140
For VSE/ESA:	568606612 R55I

Continue with “Identify the Service Level”.

Identify the Service Level

The service level is not part of the keyword string:

1. If service tapes have been applied to the licensed programs, identify the tape level of the last service tape applied.
2. Find the service level as follows:

- Under z/OS, record the level of the program update tape (PUT) that you have installed.
- Under z/VM, inspect the VMSESV output listing for the current service level of your IBM Compiler for REXX on zSeries or IBM Library for REXX on zSeries.
- Under VSE/ESA, record the level of the component using the Maintain System History Program (MSHP) RETRACE COMP statement.

Note: The service level of the Compiler may be identified from the compiler listing or from the compiled program file. See “Identify the Release Level for Compilation Problems” on page 8.

The service level of the Library can be found in the EAGRTxxx MODULE (where xxx can be either LIB or ALT) next to the product identification data. For example:

```
EAGRTLIB 4.0 LVL -NONE--
```

3. Record the current service level information. Use the EAGQRLIB EXEC to display the service level. This exec lists the library symptom string, such as:

```
EAGRTLIB 4.0 LVL -NONE-- LL(6) Generated mm/dd/yy hh.mm
EAGRTALT 4.0 LVL -NONE-- LL(14) Generated mm/dd/yy hh.mm
Name      Rel Level APAR  Library Level ...
```

The source of the EAGQRLIB EXEC utility contains the following definition:

```
/* REXX -----*/
/*   Diagnose to query the REXX Runtime Library Sympton string.   */
/*                                                                 */
/*   Licensed Materials - Property of IBM                         */
/*   5695-014 IBM REXX Library                                    */
/*   (C) Copyright IBM Corp. 1989, 2003                          */
/*                                                                 */
/* Change Activity:                                              */
/*   03-05-28          Release 4.0                               */
/*                                                                 */
/*-----*/
```

```
Trace '0';
Parse source src;
Say 'Query the REXX Runtime Library sympton string';
Say 'Source:' src;
If word(src,1)='CMS' then eagname='EAGRTPRC';
Else          eagname='EAGRTPRQ';
Say ' Calling query entry' eagname;
If word(src,1)='CMS' then do;
  'NUCXDROP' eagname;
  ADDRESS COMMAND eagname;
End;
Else ADDRESS LINKMVS eagname;
If rc>0 then do;
  Say ' Address: ' right(d2x(rc),8,0);
  ids=c2d(storage(d2x(rc+16),4));
  lvl=storage(d2x(ids+2),c2d(storage(d2x(ids),2)));
  Say ' Sympton:' lvl;
  Say ' Descrpt: Name      Rel    APAR    LibLevel';
  If word(lvl,1)='EAGRTALT' then,
    Say ' The REXX Alternate Library is in effect.';
  Else Say ' The REXX Runtime Library is in effect.';
End;
Else Do;
```

```

Say ' 'eagname' returned RC='rc;
Say ' A REXX Runtime System was not found.';
End;
Exit;

```

Notes:

- a. Under z/OS, the EAGQLIB EXEC is located in the data set *prefix.SEAGCMD*.
- b. Under z/VM, you can find the EAGQLIB EXEC on the installation minidisk of the IBM Library for REXX on zSeries.
- c. Under z/VM, when the REXX Library is defined for a segment via the loader EAGRTPRC (Segment=EAGRTSEG), the EXEC cannot display the symptom string. You must then define Segment=None for EAGRTPRC.

Continue with “Identify the Type of Failure”.

Identify the Type of Failure

Use the following table to identify the keyword that best describes the type of failure that occurred. If more than one keyword describes the problem, use the one that appears first in the table. Then go to the procedure for the chosen keyword.

Type of Failure	Symptom	Procedure
ABEND	Abnormal termination of the Compiler or of a compiled REXX program. Use the WAIT/LOOP keyword if the abnormal termination was forced by the system operator because too much time was spent in a wait state or an unintended loop.	“ABEND Procedure” on page 12
DOC	Incorrect, missing, or unclear information in one of the manuals for the IBM Compiler and Library for REXX on zSeries, the help text, or the help panels.	“DOC Procedure” on page 13
INCORROUT	Incorrect, unexpected, or missing output.	“INCORROUT Procedure” on page 14
MSG	A compilation or Library message: <ul style="list-style-type: none"> • Indicates an unrecoverable error • Appears when it should not appear • Does not appear when it should appear • Indicates an endlessly repeated message 	“MSG Procedure” on page 15
PERFM	Performance degradation in the Compiler or in a compiled REXX program. Most performance problems can be related to system tuning and should be handled by system engineers and system programmers. Use this keyword only when the problem first occurs after the installation of either a PTF or a new release of the Compiler.	“PERFM Procedure” on page 16
WAIT/LOOP	Unexpected suspension, or uncontrolled looping, of the Compiler or of a compiled REXX program; either of these may be indicated by a batch job that never returns or by the lack of an expected response in interactive mode. Use the MSG keyword for an endlessly repeated message.	“WAIT/LOOP Procedure” on page 17

ABEND Procedure

Follow the appropriate procedure in this section, depending on whether the failure occurred during compilation or at runtime.

For Compilation Failures

1. Gather diagnostic information, which your IBM Support Center representative might ask for, as follows:
 - a. Record the messages that indicate an abnormal termination. Such messages are issued by the component that trapped the error.
 - b. Gather Compiler information.
 - Under z/OS:
 - 1) Get a symptom dump and record the contents of the program status word (PSW) and the abnormal termination code.
 - 2) Get a dump (SYSUDUMP or SYSMDUMP) of the problem.
 - Under z/VM:
 - 1) If the PSW is not shown in any message, display it by entering:
`#cp d psw`
 - 2) Record the contents of the PSW and get the abnormal termination code.
 - 3) To get the abnormal termination code, see the chapter on CMS abend codes in the corresponding z/VM CMS documentation.
2. Append the abnormal termination code *xxx* to the keyword ABEND. For example, if the abnormal termination code is 0C4 (addressing exception), the type-of-failure keyword is ABEND0C4.

Your keyword string could now look like this

For z/OS:	569501303 R140 ABEND0C4
For z/VM:	569501304 R140 ABEND0C4

Go to “Identify the Module” on page 18.

For Library Failures

1. Gather diagnostic information for later use, as follows:
 - a. Record the messages that indicate an abnormal termination. Such messages are issued by the component that trapped the error.
 - b. Gather Library information.
 - Under z/OS:
 - 1) Get a symptom dump and record the contents of the PSW and the abnormal termination code.
 - 2) Get a dump (SYSUDUMP or SYSMDUMP) of the problem.
 - Under z/VM:
 - 1) If the PSW is not shown in any message, display it by entering:
`#cp d psw`
 - 2) Record the contents of the PSW and get the abnormal termination code.
 - 3) To get the abnormal termination code, see the chapter on CMS abend codes in the corresponding z/VM CMS documentation.
 - Under VSE/ESA:
 - 1) Get a symptom dump and record the contents of the PSW and the abnormal termination code.

- 2) Get a dump of the problem as described in “VSE/ESA Users” on page 6.
2. Under z/OS and z/VM, append the abnormal termination code *xxx* to the keyword ABEND. For example, if the abnormal termination code is 0C4 (addressing exception), the type-of-failure keyword is ABEND0C4. Under VSE/ESA, the type-of-failure keyword is PROGCK.

Your keyword string could now look like this

```
For z/OS:      569501403 R140 ABEND0C4
For z/VM:      569501404 R140 ABEND0C4
For VSE/ESA:   568606612 R55I  PROGCK
```

Go to “Identify the Module” on page 18.

DOC Procedure

1. First decide whether the documentation problem is severe enough to cause lost time for other users. If the problem will not significantly disrupt productive work, use the readers’ comment form or the address or Fax number provided in the document to suggest improvements. If the problem is severe, continue with the following instructions.
2. Find the page number or panel ID as follows:
 - Note the page number in the document where the error or omission occurs, and prepare a description of the problem. To the keyword string, add the DOC keyword followed by a space and the order number of the document. Omit hyphens. For example, if the order number is SH19-8179-02, the type-of-failure keyword is DOC SH19817902.

Your keyword string could now look like this

```
For z/OS:      569501303 R140 DOC SH19817902
For z/VM:      569501304 R140 DOC SH19817902
For VSE/ESA:   568606612 R55I  DOC SH19817902
```

- Note the panel ID in the help information where the error or omission occurs, and prepare a description of the problem.

Find the panel ID as follows:

- Under z/OS, enter **panelid on** on the command line to display the panel ID at the top left corner of the ISPF panel. Enter **panelid off** to return to normal.
- Under z/VM, the name of the help panel is the first token in the upper left corner if the second token is MENU; otherwise, it is the second token.

To the keyword string, add the DOC keyword followed by a space and the panel ID. For example, if the panel ID is FAN4E000, the type-of-failure keyword is DOC FAN4E000. If the panel ID is REXCO1, the type-of-failure keyword is DOC REXCO1.

Your keyword string could now look like this

```
For z/OS:      569501303 R140 DOC FAN4E000
For z/VM:      569501304 R140 DOC REXCO1
```

Go to Chapter 4, “Searching for a Solution”, on page 21.

INCORROUT Procedure

1. Add the keyword INCORROUT to the keyword string.
2. Select and add to the keyword string a modifier keyword from the following table to further describe the type of failure:

Modifier Keyword	Type of Incorrect Output
DUPLICATE	Some data or records were duplicated, but were not repeated endlessly.
INVALID	The output was not as expected.
MISSING	Some expected output was missing.

3. Add the selected modifier to the keyword string.

Your keyword string could now look like this

```
For z/OS:      569501303 R140 INCORROUT INVALID
For z/VM:      569501304 R140 INCORROUT INVALID
For VSE/ESA:   568606612 R55I INCORROUT INVALID
```

For Compilation Failures

4. If the error occurred during compilation, select another modifier keyword from the following table to identify the part of the output in which the error occurred:

Modifier Keyword	Part of Output in Error
CEEXEC	Compiled program—compiled EXEC
HELP	Compiler-specific online help
MESSAGE	Compiler message
OBJECT	Compiled program—TEXT file (z/VM) Compiled program—object module (z/OS)
SOURCE	Source listing
STAT	Error statistics
XREF	Cross-reference listing

5. Add the selected modifier to the keyword string.

Your keyword string could now look like this

```
For z/OS:      569501303 R140 INCORROUT INVALID XREF
For z/VM:      569501304 R140 INCORROUT INVALID XREF
```

For Library Failures

4. If the error occurred at runtime, select another modifier keyword from the following table to identify the part of the output in which the error occurred:

Modifier Keyword	Part of Output in Error
RESULTS	Output of the running program
MESSAGE	Library message

5. Add the selected modifier to the keyword string.

Your keyword string could now look like this

```
For z/OS:      569501403 R140 INCORROUT INVALID RESULTS
For z/VM:     569501404 R140 INCORROUT INVALID RESULTS
For VSE/ESA:  568606612 R55I INCORROUT INVALID RESULTS
```

Go to Chapter 4, “Searching for a Solution”, on page 21.

MSG Procedure

1. Find and record the message identifier. All messages for the IBM Compiler for REXX on zSeries and IBM Library for REXX on zSeries consist of the following message format:

ppp Compiler (FAN) or Library (EAG) product prefix

xxx REX (runtime), ALT (Alternate Library, refer to the message with the REX identifier), SIO (REXX Stream I/O), or the Compiler phase identifier. The following list shows the identifiers of the Compiler phases:

COD Coder

CON Controller

ENV Environment interface

FLA Flattener

FMU Final make-up

GAO Global analyzer and optimizer

LIS Lister

PAR Parser

POP Post-optimizer

TOK Tokenizer

nnnn Message number

For example, EAGREX3300E is the main message for an error 33. EAGREX3301I is a secondary message providing more information about error 33.

s Severity code, it can be:

I Informational

W Warning

E Error

S Severe error

T Terminating error

The messages issued by the Compiler always include the identifier.

The following items refer to the messages issued by the Library:

- Under z/OS and TSO/E, messages include the identifier only if the TSO/E command **profile msgid on** has been issued. If you had not issued the **profile msgid** command, when the problem occurred, issue the command and try to reproduce the problem. If you cannot reliably reproduce the problem, scan for the message in the *IBM Compiler and Library for REXX on zSeries: User's Guide and Reference*.
- Under z/VM, messages include the identifier only if the **cp** command **set emsg on** has been issued. If you had not issued the **emsg** command, when the problem occurred, issue the command and try to reproduce the problem. If you cannot reliably reproduce the problem scan for the message in the *IBM Compiler and Library for REXX on zSeries: User's Guide and Reference*.
- Under VSE/ESA, if the Interactive Computing and Control Facility (ICCF) is available, you can select operations to either retrieve the message or to browse the system console to find the message. Alternatively, you can use the problem handling dialog under ICCF for problem determination. If ICCF is not available, or you cannot reliably reproduce the problem, scan for the message in the *IBM Compiler and Library for REXX on zSeries: User's Guide and Reference*.

Use the PRINTLOG utility to print the hardcopy file, which resides on SYSREC (refer to *IBM VSE/ESA System Utilities*) and use the LISTLOG utility to print the job information.

2. Append the message identifier to the keyword MSG.

For example, if the Compiler message identifier is FANCON0055T, the type-of-failure keyword is MSGFANCON0055T.

Your keyword string for the Compiler could now look like this

```
For z/OS:      569501303 R140 MSGFANCON0055T
For z/VM:      569501304 R140 MSGFANCON0055T
```

Similarly, if the Library message identifier is EAGREX4000E, the type-of-failure keyword is MSGEAGREX4000E.

Your keyword string for the Library could now look like this

```
For z/OS:      569501403 R140 MSGEAGREX4000E
For z/VM:      569501404 R140 MSGEAGREX4000E
For VSE/ESA:   568606612 R55I MSGEAGREX4000E
```

Go to Chapter 4, "Searching for a Solution", on page 21.

PERFM Procedure

Use this procedure only if you have installed a new release or modification level or applied a program temporary fix (PTF).

1. Record the expected performance and the actual performance. The performance could be affected by storing data sets further apart or by any system-related actions or problems, and by the compiler options used. Record the type of maintenance that caused the problem.
2. Add the keyword PERFM to the keyword string.

Your keyword string for the Compiler could now look like this

```
For z/OS:      569501303 R140 PERFM
For z/VM:      569501304 R140 PERFM
```

Your keyword string for the Library could now look like this

```
For z/OS:      569501403 R140 PERFM
For z/VM:      569501404 R140 PERFM
For VSE/ESA:   568606612 R55I PERFM
```

Go to Chapter 4, “Searching for a Solution”, on page 21.

WAIT/LOOP Procedure

If your program is in a wait state, no action is required. If the problem looks like a wait state and your program is not waiting for input from the terminal, it is probably a system problem. In this case, follow your local procedures for resolution. Otherwise, follow the appropriate procedure in this section, depending on whether the problem occurred during compilation or at runtime.

For Compilation Failures

1. Gather diagnostic information as follows:
 - Under z/OS, if you note a wait or loop condition, refer to the corresponding z/OS documentation to diagnose the problem.
 - Under z/VM:
 - a. While the program is looping, display the PSW by entering:
`#cp d psw`
 - b. Record the contents of the PSW. Your IBM Support Center representative might ask for this information.
2. Add the keyword LOOP to the keyword string.

Your keyword string could now look like this

```
For z/OS:      569501303 R140 LOOP
For z/VM:      569501304 R140 LOOP
```

Go to Chapter 4, “Searching for a Solution”, on page 21.

For Library Failures

1. Check whether your REXX source program contains an endless program loop. Debugging the program with the TRACE facilities might help to detect a loop. To be able to use the TRACE facilities, you must compile your program with the TRACE Compiler option.
2. If the problem is still unresolved, gather diagnostic information for later use, as follows:
 - Under z/OS, if you note a wait or loop condition, refer to the corresponding z/OS documentation to diagnose the problem.
 - Under z/VM:
 - a. While the program is still looping, display the PSW by entering:

```
#cp d psw
```

- b. Record the contents of the PSW. Your IBM Support Center representative might ask for this information.
 - Under VSE/ESA, if you encounter a wait or loop condition, refer to *VSE/ESA: Guide for Solving Problems* to diagnose the problem.
3. Add the keyword LOOP to the keyword string.

Your keyword string could now look like this

```
For z/OS:      569501403 R140 LOOP
For z/VM:      569501404 R140 LOOP
For VSE/ESA:   568606612 R55I LOOP
```

Go to “Identify the Module”.

Identify the Module

The module keyword identifies the failing module. This procedure applies both to Compiler and to Library problems.

Gather diagnostic information as follows:

- Under z/OS:
 1. Get the name of the failing module or CSECT from a dump (SYSUDUMP or SYSMDUMP). If you have the offset into the load module, use the AMBLIST utility to get the cross-reference (XREF) listing and determine where the module failed.
 2. Add the full name of the module to the keyword string.

Your keyword string could now look like this

```
For the Compiler: 569501303 R140 ABEND0C4 FANVPARS
For the Library:  569501403 R140 ABEND0C4 EAGJAC9B
```

- Under z/VM:
 1. Determine, from the PSW, the address at which the failure occurred. (You obtained the PSW in the ABEND or LOOP procedure.)

Bit 32 of the PSW controls the size of the effective addresses. When the bit is zero, the instruction address is held in the low-order 24 bits of the PSW. When the bit is one, the instruction address is held in the low-order 31 bits of the PSW.
 2. Enter the **cp display** command to display X'1000' bytes before and up to the failure address, with EBCDIC translation. For example, if the address in the PSW is 020000, enter:

```
#cp d t1f000-20000
```
 3. Observing the displayed EBCDIC data, scan forward until you see the name of an IBM Compiler for REXX on zSeries module or an IBM Library for REXX on zSeries module, followed by a date and time. The Compiler module names begin with FAN. The Library module names begin with EAG or IXX.

Write down the module name and continue scanning. When you see another module name followed by a date and time, write it down and continue scanning. The last module name found is the name of the module with the failure address.

Usually a module is not longer than X'1000' bytes. If it is, you can find the module name by following the same procedure, but starting at X'2000' bytes before the failure address.

4. If the first three characters of the module name are IXX, run the source program with the interpreter. In this case, VM/SP Release 6 is the minimum level of CMS required for the interpreter.

If the failure occurs only at the time the program is compiled, continue with step 5. If the failure also occurs at the time the program is interpreted, the problem lies in an interpreter routine. In that case, exit from this procedure and refer to the corresponding z/VM documentation.

5. Add the full name of the module to the keyword string.

Your keyword string could now look like this

```
For the Compiler: 569501304 R140 ABEND0C4 FANVPARS
For the Library:  569501404 R140 ABEND0C4 EAGJAC9B
```

- Under VSE/ESA:

1. Get the name of the failing module or CSECT from a dump. If you have the offset into the load module, you can use the problem handling dialog under ICCF to search for the name in the dump.
2. Add the full name of the module to the keyword string.

Your keyword string could now look like this

```
For the Library: 568606612 R55I PROGCK EAGJAC9B
```

Continue with Chapter 4, "Searching for a Solution", on page 21.

Chapter 4. Searching for a Solution

This section explains how to use your keyword string to search the software support database.

Besides having built a keyword string, you should also know the service level of your IBM Compiler for REXX on zSeries or your IBM Library for REXX on zSeries. Although the service level is not used as a keyword search argument, it will be useful when you review APARs selected during the keyword search.

Search for a Solution

1. If you have installed Information/Access, search it using the entire keyword string as a search argument. Otherwise, call your IBM Support Center representative.
2. If the search produced matches, continue with step 3.
If the search produced no matches, broaden the search, using the following techniques:
 - a. Try using the other three component identifier keywords described in “Include the Component Identifier” on page 7, one at a time.
 - b. Omit one keyword from the end of the keyword string and search, again. Repeat this step as often as necessary.
 - c. Omit the release-level keyword (for example, R140) from the search argument. This broadens the search to include similar failures on other releases.

If there are any matches, continue with step 3.

If there are no matches, go to Chapter 5, “Preparing an APAR”, on page 25.

3. If there are too many matches to review, narrow the search as described under “Narrowing the Search”.
4. For each match, review the problem description and the description of the APAR that closed the problem.
5. Use the service level (PTFs and APARs that have been applied) to eliminate those problems for which the corrections have already been applied to your system.
6. Compare each remaining APAR-closing description with the failure symptoms of your program.
7. If you find an exact match for the problem, apply the suggested correction and exit from this procedure.
8. If you do not find an exact match:
 - a. If the program was compiled on an earlier release, delete the release-level keyword and repeat the search.
 - b. Otherwise, go to Chapter 5, “Preparing an APAR”, on page 25.

Narrowing the Search

To narrow the search, add one or more modifier keywords. A modifier keyword names a REXX language element or Compiler option associated with the failure. Use a modifier keyword only if you think it is significant for the type of failure.

Identifying REXX Language Elements

The REXX language elements are the REXX instructions, keywords, sub-keywords, operators, built-in functions, and special variables. Determine the REXX language element being processed at the time of failure.

Examples of modifier keywords for these elements are:

```
ADDRESS
WHILE
>=
LENGTH
SIGL
```

Identify modifier keywords as follows:

- Under z/OS, spell these modifier keywords as described in the corresponding z/OS documentation.
- Under z/VM, spell these modifier keywords as described in the corresponding z/VM CMS documentation.
- Under VSE/ESA, spell these modifier keywords as described in the corresponding VSE/ESA documentation.

You can also use the modifier keyword %PAGE to refer to the %PAGE listing control directive.

For Compilation Failures

The message marker (|) points to the source code being processed, as shown in Figure 1.

```
IBM Compiler for REXX on zSeries 4.0 LVL -NONE-- Time: 15:14:58 Date: 2003-06-11 Page: 1
If Do Sel Line C -----1-----2-----3-----4-----5-----6-----7-----8-----9-----0
      1 /* REXX */
      2 rr,r = rr.r +1
      |
+++FANPAR0566S Unexpected ", " in expression
```

Figure 1. Example of a Compiler Message

If there is more than one REXX language element near the message marker, choose the one that you consider most relevant to the problem. For example, if the marker points to an expression within an instruction, the name of the instruction may be more relevant than an operator in the expression.

Identifying Compiler Options

Record, from your compiler listing, those Compiler options that you consider significant for the type of failure. The modifier keyword is the name of the Compiler option, shortened to eight characters if necessary. Examples are:

```
SLINE
NOSLINE
TESTHALT
NOTH
```

Identifying Commands

You can use command names as modifier keywords to identify the command you think caused the problem. Examples are:

EAGQRLIB (z/OS, z/VM)
REXXC (z/OS, z/VM)
REXXD (z/VM)
REXXF (z/OS, z/VM)
REXXL (z/OS, VSE/ESA)
REXXV (z/OS, z/VM, VSE/ESA)

Identifying Cataloged Procedures (z/OS)

Under z/OS, you can use cataloged procedures as modifier keywords. Examples are:

REXXC
REXXCG
REXXCL
REXXCLG
REXXOEC
REXXL

Identifying Cataloged Procedures (VSE/ESA)

Under VSE/ESA, you can use cataloged procedures as modifier keywords. Examples are:

REXXPLNK
REXXLINK
REXXL

Identifying ISPF Panels (z/OS)

Under z/OS, you can use compiler invocation panels as modifier keywords. Examples are:

FANFP14
FANJP14

These are ISPF panels, therefore, you can use the **panelid on** command to display the panel name in the upper left corner.

Chapter 5. Preparing an APAR

Contact your IBM Support Center representative if your keyword search is unsuccessful. The representative prepares an APAR and asks you to supply information about the problem and to provide supporting documentation.

Initiating an APAR

Be prepared to supply your IBM Support Center representative with the following information:

- Your customer number
- The release level of your operating system
- The current service level (the latest PUT level, plus any additional APARs or PTFs)
- The keyword string used to search the software support database
- Your processor serial number

The IBM representative will provide you with an APAR number that you will need when submitting APAR documentation.

Gathering Supporting Documentation

You will be asked to supply information that describes the functions that failed and the environment in which they failed. The exact information required depends on the nature and severity of the failure, but you should be prepared to supply the following documentation:

- The source program, reduced to the smallest, least complex form that still produces the error. Discuss with your IBM Support Center representative which of the following is the best way to send the documentation, such as:
 - Tape (unlabeled or standard labeled)
 - Diskette
 - VNET
- Any special operating instructions or test data for the program.
- An example of the failing output and the expected output.
- The compiler listing, including:
 - A source listing
 - A cross-reference listing
 - The messages for all severity codes

To get such a listing, use the FLAG(I), PRINT, SOURCE, and XREF Compiler options. See *IBM Compiler and Library for REXX on zSeries: User's Guide and Reference* for more information on how to use these options.

- Environment-specific information:
 - Under z/OS, supply information about whether the address space is integrated into TSO/E and supply information about the region size. If the problem appears in a batch job, supply the job control statements and job listing. If the failure occurred in a foreground session (TSO/ISPF), supply a description of the input and output that preceded the error.

- Under z/VM, supply a spooled console log for an interactive session during which the problem occurred. This should include details of the interactive environment immediately before you invoked either the Compiler or your REXX program. To get these details, issue the command:

QUERY SET

Then, if necessary, issue:

QUERY DISK

QUERY FILEDEF

QUERY INPUT

QUERY LANG

QUERY LIBRARY

QUERY OUTPUT

QUERY TERMINAL

QUERY VIRTUAL

- Under VSE/ESA, supply information about the partition size and the GETVIS area size. If the problem appears in a batch job, supply the job control statements and job listing.

Use the MAP command to map the storage areas (refer to *VSE/ESA System Control Statements*). Use the PRINTLOG utility to print the hardcopy file, which resides on SYSREC (refer to *VSE/ESA System Utilities*) and use the LISTLOG utility to print the job information.

You may be asked to supply:

- The source program and the compiled program (compiled EXEC or TEXT. file).
- Output from the DUMP. Compiler option; for details, see Appendix A, “Obtaining Interphase Compiler Dumps”, on page 29.
- Environment-specific information:
 - Under z/OS, a dump (SYSUDUMP or SYSMDUMP).
 - Under VSE/ESA, a system dump (SYSDUMP) or an SDAID SVA dump.

With this information, IBM can try to reproduce the failure and observe the symptoms of the failure.

Submitting the APAR Documentation

Identify and pack all material that you send to IBM for an APAR.

Submitting Hardcopy

Put the assigned APAR number on the top right-hand corner of each item of printed documentation.

Submitting Tapes or Diskettes

Before you ship your tapes or diskettes to IBM as part of the APAR documentation, be sure to include:

Name Your name

Region
IBM region number

Branch office

IBM branch office number

Customer number

Tape owner's IBM customer number

APAR serial

APAR serial number assigned by IBM

Mode and density

7 or 9 track, 18 or 36 track, IDRC

Label Write one of these:

STD Standard

non-STD

Nonstandard

NO No label

File format

Such as "fixed blocked" or "variable"

Block size

Physical record size

Record Size

Logical record size

Program used to Create

The name of the program used to create the tape, such as VM TAPE
DUMP

IBM will respect the confidentiality of your documentation and, if requested, will return tapes to you.

Appendix A. Obtaining Interphase Compiler Dumps

If you report a compilation problem, your IBM Support Center representative might ask you to provide formatted interphase dumps. These might also be requested if you report a Library failure and the IBM support personnel suspect that the Compiler generated faulty code. The dumps show the control blocks and the text being compiled after selected phases of the Compiler. The IBM Support Center representative will tell you which dumps are required.

To obtain the dumps, recompile the program, using the same compiler options as when the problem occurred and add the DUMP compiler option (described in *IBM Compiler and Library for REXX on zSeries: User's Guide and Reference*).

In place of the *n* in the syntax, use one of the numbers below. You can get several dumps in one compilation by adding the numbers of the required dumps. For example, to dump the **token stream (1)** and the **abstract syntax tree after the parser phase (2)**, specify **DUMP(3)**.

DUMP[*n*]

where *n* is a number that determines which dumps are to be produced, as follows:

<i>n</i>	Dump Output
0	No dumps
1	Token stream
2	Abstract syntax tree after the parser phase
4	Abstract syntax tree after the global analyzer and optimizer phase
8	Abstract REXX machine instructions
16	Preliminary code
32	Machine code after the post-optimizer phase
64	Final code
128	Symbol table after the parser phase
256	Symbol table after the global analyzer and optimizer phase
512	Symbol table after the coder phase
1024	Symbol table after final make-up
2047	All dumps

The resulting dump file is sent to the virtual printer under z/VM or to the ddname SYSDUMP under z/OS. Note that the spool file may be large.

Under z/VM, if you compile a program to obtain interphase dumps and an ABEND occurs before closing the virtual printer, enter the following to close the virtual printer and get the interphase dumps up to the point at which the ABEND occurred:

```
#cp close print
```

Appendix B. Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

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 NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Deutschland
Informationssysteme GmbH
Department 3982
Pascalstrasse 100

70569 Stuttgart
Germany

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming Interface Information

This *Diagnosis Guide* documents information NOT intended to be used as Programming Interfaces of the IBM Compiler and Library for REXX on zSeries.

Trademarks

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

- C/370
- IBM
- MVS
- MVS/ESA
- NetView
- OpenEdition
- OS/390
- SAA
- SP
- Systems Application Architecture
- VM/ESA
- VSE/ESA
- z/OS
- zSeries
- z/VM

Other company, product, and service names may be trademarks or service marks of others.

List of Abbreviations

Abend	Abnormal end of task
APAR	Authorized program analysis report
CMS	Conversational Monitor System
DCSS	Discontiguous saved segment
EBCDIC	Extended binary-coded decimal interchange code
GETVIS	Get virtual storage
ISPF	Interactive System Productivity Facility
PSW	Program status word
PTF	Program temporary fix
PUT	Program update tape
REXX	Restructured EXtended eXecutor language
SDAID	System debugging aids
SVA	Shared virtual area
VM	Virtual Machine
VSE/ESA	Virtual Storage Extended/Enterprise Systems Architecture
XA	Extended Architecture
SP™	System Product
ESA	Enterprise Systems Architecture

Related Publications

This section lists each book in the REXX library. There is also a list of publications for other IBM products that you might use with REXX.

IBM Compiler and Library for REXX on zSeries Publications

The following books are part of the IBM Compiler and Library for REXX on zSeries publications:

IBM Compiler and Library for REXX on zSeries: User's Guide and Reference, SH19-8160, describes how to compile and run programs written in the REXX language.

IBM Compiler and Library for REXX on zSeries: Diagnosis Guide, SH19-8179, provides information for system programmers and other data processing professionals responsible for maintaining the IBM Compiler and Library for REXX on zSeries. It explains how to diagnose suspected errors in the product and how to report them to the appropriate IBM personnel.

IBM Compiler and Library for REXX on zSeries: Licensed Program Specifications, GH19-8161, describes the software and hardware requirements of IBM Compiler and Library for REXX on zSeries.

IBM Compiler for REXX on z/OS: Program Directory, GI10-8170 describes the requirements and installation of IBM Compiler for REXX on z/OS.

IBM Library for REXX on z/OS: Program Directory, GI10-9910 describes the requirements and installation of IBM Library for REXX on z/OS.

IBM Alternate Library for REXX on z/OS: Program Directory, GI10-3243 describes the requirements and installation of the IBM Alternate Library for REXX on z/OS.

You can also find the library of the IBM Compiler and Library for REXX on zSeries on the home page at:
<http://www.ibm.com/software/awdtools/rexx/>

The unlicensed REXX books with prefix SH are also available on the following collection kits:

- *IBM eServer zSeries Online Library VM Collection* CD-ROM, SK2T-2067
- *IBM eServer zSeries Online Library VSE Collection* CD-ROM, SK2T-0060
- *IBM eServer zSeries Online Library z/OS Software Products Collection* CD-ROM, SK3T-4270

Other IBM Publications

These books contain information related to REXX or its related products.

ISPF Publications

ISPF V4 R2.0 Dialog Developer's Guide and Reference, SC34-4486

ISPF V4 R2.0 Services Guide, SC34-4485

ISPF V4 R2.0 User's Guide, SC34-4484

ISPF/PDF Guide (ISPF 3.2 & ISPF/PDF 3.1) for VM, SC34-4299

ISPF/PDF Guide and Reference V3.4 for MVS, SC34-4258

ISPF/PDF Guide Version 3, Release 2 for VM, SC34-4306

Learning REXX

- *TSO/E Version 2 REXX/MVS: User's Guide*, SC28-1882
- *VM/SP System Product Interpreter: User's Guide*, SC24-5238
- *VM/XA SP Interpreter: User's Guide*, SC23-0375
- *VM/ESA REXX/VM: User's Guide*, SC24-5465

REXX Reference

- *TSO/E Version 2 Procedures Language MVS/REXX*, SC28-1883
- *VM/XA SP Interpreter: Reference*, SC23-0374
- *VM/ESA Release 2 REXX/VM: Reference*, SC24-5466
- *IBM VSE/ESA REXX/VSE Reference*, SC33-6529, is interesting for experienced programmers, particularly those who have used a structured high-level language. They list the REXX

messages and describes instructions, functions, debugging aids, and parsing.

- *Systems Application Architecture*[®] *Common Programming Interface: REXX Level 2 Reference*, SC24-5549, describes the SAA[®] REXX interface.

TSO/E and MVS/ESA™ Publications

- *TSO/E Version 2: Primer*, GC28-1879
- *TSO/E Version 2: Customization*, SC28-1872
- *TSO/E Version 2 REXX/MVS: User's Guide*, SC28-1882
- *TSO/E Version 2 REXX/MVS: Reference*, SC28-1883
- *TSO/E Version 2: Command Reference*, SC28-1881
- *MVS/DFP 3.3: Linkage Editor and Loader*, SC26-4564
- *MVS/ESA SP V4 Planning: Operations*, GC28-1625
- *MVS/ESA SP V4 Assembler Programming Guide*, GC28-1644

OpenEdition[®] Publication

- *OpenEdition MVS Command Reference*, SC23-3014

VM/SP Publications

- *VM/SP CMS: Primer*, SC24-5236
- *VM/SP CMS: Primer for Line-Oriented Terminals*, SC24-5242
- *VM/SP CMS: User's Guide*, SC19-6210
- *VM/SP CMS: Command Reference*, SC19-6209
- *VM/SP System Product Editor: User's Guide*, SC24-5220
- *VM/SP: Administration*, SC24-5285
- *VM/SP System Messages and Codes*, SC19-6204

VM/XA SP Publications

- *VM/XA SP CMS: Primer*, SC23-0368
- *VM/XA SP CMS: User's Guide*, SC23-0356
- *VM/XA SP CMS: Command Reference*, SC23-0354
- *VM/XA SP System Product Editor: User's Guide*, SC23-0373
- *VM/XA SP: Administration*, SC23-0353
- *VM/XA SP Interpreter Reference*, SC23-0374
- *VM/XA SP CP Command Reference*, SC23-0358

VM/ESA Publications

- *VM/ESA R2.2 REXX/VM User's Guide*, SC24-5465
- *VM/ESA V2R3.0 Diagnosis Guide*, GC24-5854
- *VM/ESA V2R4.0 REXX/VM Reference*, SC24-5770
- *VM/ESA V2R4.0 CP Command and Utility Reference*, SC24-5773

VSE/ESA Publication

- *VSE/ESA V2R1.0 System Control Statements*, SC33-6613
- *VSE/ESA V2R1.0 System Utilities*, SC33-6617
- *VSE/ESA V2R4.0 Guide for Solving Problems*, SC33-6710

C Publication

- *IBM C/370™ Programming Guide Version 2 Release 1*, SC09-1384

CMS Publications

- *VM/ESA CMS: Primer*, SC24-5458
- *VM/ESA CMS: User's Guide*, SC24-5775
- *VM/ESA CMS: Command Reference*, SC24-5776
- *VM/ESA XEDIT: User's Guide*, SC24-5779
- *z/VM V4R1.0 CMS User's Guide*, SC24-6009
- *z/VM V4R3.0 CMS Command and Utility Reference*, SC24-6010
- *z/VM V4R3.0 CMS Planning and Administration*, SC24-6042

z/VM Publications

- *z/VM V3R1.0 CP Command and Utility Reference*, SC24-5967
- *z/VM V4R3.0 CP Command and Utility Reference*, SC24-6008
- *z/VM V4R3.0 Saved Segments Planning and Administration*, SC24-6056

z/OS Publications

- *z/OS V1R2.0 TSO/E REXX User's Guide*, SA22-7791
- *z/OS V1R4.0 TSO/E REXX Reference*, SA22-7790
- *z/OS V1R4.0 TSO/E Command Reference*, SA22-7782
- *z/OS V1R3.0 MVS Planning: Operations*, SA22-7601

OS/390 Publications

- *OS/390 V2R9.0 TSO/E REXX User's Guide*, SC28-1974
- *OS/390 V2R10.0 TSO/E REXX Reference*, SC28-1975
- *OS/390 V2R9.0 TSO/E System Programming Command Reference*, SC28-1972
- *OS/390 V2R10.0 MVS Planning: Operations*, GC28-1760

Index

A

ABEND 11
ABEND Procedure 12
APAR, Preparing an 25

C

Compilation Problems 8
Compiler Dumps, Interphase 29
Component Identifier 1, 7

D

DOC 11
DOC Procedure 13
Dumps, Interphase Compiler 29

E

EAGALT (message identifier) 15
EAGQLIB EXEC 10
Elements, Identifying REXX
Language 22
EXEC, EAGQLIB 10

F

Failure, Type of 1, 11
FANxxx (message identifier) 15

H

Help, Service vi

I

Identifier, Component 1, 7
Identify the Module 18
Identifying REXX Language
Elements 22
INCORROUT 11
Interphase Compiler Dumps 29

L

Language Elements, Identifying
REXX 22
Level, Release 1, 7
Level, Service 1, 9
Library Problems 8

M

Modifiers 2
Module 1
Module, Identify the 18
MSG 11

N

Notices 31

P

PERFM 11
PERFM Procedure 16
Preparing an APAR 25
Problems, Compilation 8
Problems, Library 8
Procedure, ABEND 12
Procedure, DOC 13
Procedure, PERFM 16
Procedure, WAIT/LOOP 17

R

Release Level 1, 7
REXX Language Elements,
Identifying 22

S

Search 21
Service Help vi
Service Level 1, 9
Syntax notation vi

T

Trademarks 33
Type of Failure 1, 11

W

WAIT/LOOP 11
WAIT/LOOP Procedure 17
Worksheet 1

Readers' Comments — We'd Like to Hear from You

IBM Compiler and Library for REXX on zSeries
Diagnosis Guide
Version 1 Release 4

Publication No. SH19-8179-02

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

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

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

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you? Yes No

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.

Name

Address

Company or Organization

Phone No.



Fold and Tape

Please do not staple

Fold and Tape

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Entwicklung GmbH
Information Development, Dept 0446
Schönaicher Strasse 220
71032 Böblingen
Germany

Fold and Tape

Please do not staple

Fold and Tape



Program Number: 5695-013
5695-014

Printed in Denmark by IBM Danmark A/S

SH19-8179-02

