IBM VSE/Enterprise Systems Architecture
VSE Central Functions

REXX/VSE Diagnosis Reference

Version 6 Release 1

SC33-6332-00

IBM VSE/Enterprise Systems Architecture
VSE Central Functions

REXX/VSE Diagnosis Reference

Version 6 Release 1

SC33-6332-00

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 6 Release 1 of IBM REXX/VSE, which is part of VSE/Central Functions, Program Number 5686-066, and to all subsequent releases and modifications until otherwise indicated in new editions.

Publications are *not* stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to:

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

You may also send your comments by FAX or via the Internet:

Internet: s390id@de.ibm.com
FAX (Germany): 07031-16-3456
FAX (other countries): (+49)+7031-16-3456

IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Contents

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

This publication is intended primarily for use by IBM personnel responsible for program service. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. It is not intended as a description of a programming interface. The use of this information is a customer reponsibility. Service for errors, ommissions, accuracy, or completeness will not be provided.

IBM may have patents or pending patent applications covering subject matter 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 the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, New York 10594, USA.

Programming Interface Information

This book is intended to help customers diagnose problems caused by REXX/VSE processing. This book documents information that is Diagnosis, Modification or Tuning Information that REXX/VSE 6.1.0 provides.

WARNING: Do not use this Diagnosis, Modification or Tuning Information as a programming interface.

Trademarks and Service Marks

The following terms, denoted by an asterisk (*) in this publication, are trademarks of the IBM Corporation in the United States or other countries or both:

IBM
SAA
VSE/ESA

About This Book

This book provides diagnosis information for the REstructured eXtended eXecutor (REXX) language, the REXX/VSE interpreter (called the language processor), and the Library for REXX/370 in REXX/VSE (called a compiler's runtime processor).

Using the book, you will be able to:

- Follow diagnostic procedures for each type of problem
- Collect and analyze data needed to diagnose the problem
- Develop a search argument to use in searching problem reporting data bases
- Know what problem data is needed before reporting the problem to IBM*.

Who Should Use This Book

This book is for anyone who diagnoses system problems that REXX/VSE appears to cause.

The writing in this book is at a level of detail that assumes the reader:

- Understands the commonly used diagnostic tasks and aids
- Codes in assembler language and reads assembler and linkage editor output
- Understands basic system concepts and the use of system services
- Understands how to call the external interfaces available to REXX programs. The *VSE/ESA REXX/VSE Reference*, SC33-6642, describes these interfaces.

This book does not provide information about errors that occur in user-written REXX programs. For help in diagnosing problems in REXX programs, REXX/VSE provides:

- The REXX TRACE keyword instruction and TRACE built-in function
- The special variables RC and SIGL
- The interactive debug facility (in a batch environment, interaction is between the current input stream and the program)
- The immediate commands TS (Trace Start) and TE (Trace End)
- The trace and execution control routine ARXIC.

For more information, see *VSE/ESA REXX/VSE Reference* and *VSE/ESA REXX/VSE User's Guide*.

Modules with a prefix of IXX or ARX are part of the language processor. Modules with a prefix of EAG are part of the Library for REXX/370 in REXX/VSE. If your program abends during processing of an IXX, ARX, or EAG module, the error eventually appears as a REXX error, and the appropriate diagnostic procedure (based on the symptom) in this book explains how to diagnose the problem.

How This Book Is Organized

This book contains the following:

- **Chapter 1, "Diagnosing Problems in REXX Processing"** identifies the problem types and gives a diagnostic procedure for each type. You should begin your diagnosis task *in this chapter*.
- **Chapter 2, "Analyzing Problem Data for REXX Processing"** tells how to collect and analyze needed problem data for REXX problems.
- **Chapter 3, "Developing a Search Argument"** summarizes the search argument symptoms for problems in REXX.
- **Chapter 4, "Reporting a Problem to IBM"** summarizes the problem data you need to report the problem to IBM.

Where to Find More Information

You may need the following publications for reference:

REXX/VSE Publications

- *VSE/ESA REXX/VSE Reference*, SC33-6642
- *VSE/ESA REXX/VSE User's Guide*, SC33-6641.

VSE/ESA Publications

- *VSE/ESA System Control Statements*, SC33-6613
- *VSE/POWER Administration and Operation*, SC33-6633
- *VSE/POWER Application Programming*, SC33-6636
- *VSE/ESA Guide to System Functions*, SC33-6611
- *VSE/ESA Library Guide*, GC33-6619
- *VSE/ESA Messages and Codes*, SC33-6607.

Library and Compiler for SAA* REXX/370 Publications

- *IBM Compiler and Library for SAA REXX/370 Release 2: Introducing the Next Step in REXX Programming*, G511-1430
- *IBM Compiler and Library for SAA REXX/370 Release 2 User's Guide and Reference*, SH19-8160
- *IBM Compiler and Library for SAA REXX 370 Release 2 Diagnosis Guide*, SH19-8179

Referenced Program Product

This book refers to the following product:

VSE/ESA* Version 2 Release 1, Program Number 5690-vse.

Chapter 1. Diagnosing Problems in REXX Processing

This chapter discusses:

- Prerequisites
- Identifying problems
- Diagnosing abends and return and reason codes.

Prerequisites

Before using this book, collect the following problem data:

- The problem type, such as an abend
- The product name: REXX/VSE
- The component name (one of the following):
  - 568606616 15I — REXX kernel REXX/VSE interface
  - 568606612 15I — Library for REXX/370 in REXX/VSE.

If you do not have this data, see *VSE/ESA* Diagnosis Tools*, SC33-6614, and perform the procedures it recommends.

Use this book to diagnose problems only for REXX/VSE 6.1.0. If the product name is not REXX/VSE or the component name is not one of those listed earlier, return to *VSE/ESA Diagnosis Tools*, SC33-6614, to identify the component and to diagnose problems in the component.

Identifying Problems

For a problem with these symptoms:	See the following:
Abend occurring in a module with prefix ARX, EAG, or IXX: Modules with a prefix of ARX or IXX are part of the language processor. Modules with a prefix of EAG are part of the Library for REXX/370 in REXX/VSE.	For ARX or IXX modules, see "Diagnosing an Abend in REXX Processing" on page 2. For EAG modules, see the *IBM Compiler and Library for SAA REXX 370 Release 2 Diagnosis Guide*, SH19-8179.

For a problem with these symptoms:	See the following:
Return code and reason code from a REXX/VSE service routine. If you receive a return code or reason code from a program in your batch job's output or from one of the following REXX/VSE service routines, and your program did NOT call any of these routines, note the return code or reason code and go to “Diagnosing a Return Code and Reason Code” on page 4. The following lists the names of the REXX/VSE service routines. • ARXEXEC • ARXEXCOM • ARXERS • ARXHLT • ARXHST • ARXIC • ARXINIT • ARXINOUT • ARXJCL • ARXLIN • ARXLOAD • ARXMSGID • ARXRLT • ARXRTE • ARXSAY • ARXSTK • ARXSUBCM • ARXTERM • ARXTERMA • ARXTXT • ARXUID	“Diagnosing a Return Code and Reason Code” on page 4

Diagnosing an Abend in REXX Processing

Use this procedure if a module with a prefix of ARX or IXX abnormally ends.

Diagnostic procedure	References
1. Look at the explanation for the abend code (and any accompanying reason code). Take the recommended actions. If you are unable to correct the problem, obtain any messages accompanying the abend. Look at their explanations and take the recommended actions. If you are unable to correct the problem, continue with the next step.	See *VSE/ESA Messages and Codes*, SC33-6607, for the abend code and its reason code and for operating system messages
2. Collect problem data, including the dump for the abend. Ensure that the job uses the DUMP option.	See the OPTION and STDOPT JCL statements in the *VSE/ESA System Control Statements*, SC33-6613.
3. Format the dump using the INFO/ANALYSIS program.	See *VSE/ESA Diagnosis Tools*, SC33-6614, and *VSE/ESA Guide for Solving Problems*, SC33-6610.

Diagnostic procedure	References
4. Examine the REXX environment block in the dump with the ARXENVB mapping macro. (This macro is in PRD1.BASE.) The REXX environment block is helpful for diagnosis because it contains pointers to other important REXX control blocks. For example, the PARMBLOCK field in the environment block points to the parameter block. The parameter block contains data about the parameters used to define the environment. One of these parameters is the FLAGS field at offset X'24'. One flag in the FLAGS field is the NOMSGWTO flag, which indicates whether REXX error messages are issued. The environment block also contains the first message issued in the language processor environment active when the abend occurred. If the module name indicates an exit routine (in REXX for example, routines ARXINITX, RXHLT, ARXITMV, and ARXTERMX), then continue diagnosis with that program, using the dump for the abend.	See the *VSE/ESA REXX/VSE Reference* for the format of the environment block.
5. Develop a search argument consisting of: • Component identifier: PIDS/5686066 • Program or phase name: RIDS/ARXccccc or RIDS/IXXccccc • System abend code: AB/Sxxyy (where xx is the first cancel code and yy is the second cancel code or 0) • Abend reason code: PRCS/hhhhhhhh • Message identifier: MS/ARXnnnns • REXX return code or reason code or both (if applicable): PRCS/hhhhhhhh Use the search argument to search problem reporting data bases. If the first search produces no matches, remove some symptoms and search again. If the first search produces too many matches, add one or more of the following and search again: • Register/PSW difference: REGS/xxyyy (xx is the hexadecimal number of the general register containing an address close to the point of failure; yy is the contents of the register minus the PSW instruction address) name: RIDS/cccccccc#R • Input request (REXX/VSE command or REXX instruction): PCSS/cccccccc If the search finds that the problem has been reported before, request the problem fix. If not, continue with the next step.	See Chapter 3, “Developing a Search Argument” on page 11.
6. Analyze the dump for the offset of the failing instruction into the phase. Get the address from the right half of the program status word (PSW) in the INFO/ANALYSIS output. Subtract the instruction length from the PSW address to obtain the address of the failing instruction. Ignore the 8 (that is, the high-order bit of the PSW address), if present before the PSW address.	See the *VSE/ESA Guide for Solving Problems*, SC33-6610.
7. Collect and analyze any other problem data recommended in the procedure for an abend in the *VSE/ESA Guide for Solving Problems*.	See the *VSE/ESA Guide for Solving Problems*.

Diagnostic procedure	References
8. Report the problem to IBM, if further assistance is needed or if the problem is new. Provide the following problem data: • Problem type: abend • The search argument • Product name: REXX/VSE, Version 6 Release 1 • REXX return code or reason code or both • Module name and level • The dump formatted by INFO/ANALYSIS on-line or printed. • Offset of the failing instruction into the module • Name and level of the operating system with a list of program temporary fixes (PTFs) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service • Other problem data developed while using *VSE/ESA Guide for Solving Problems*.	See Chapter 4, "Reporting a Problem to IBM" on page 13.

Diagnosing a Return Code and Reason Code

Use this procedure if you receive an unexpected return code or reason code from an application programming interface (API). If the return code or reason code is related to an abend, then *do not proceed further* in this table. Instead, use the procedure for the abend. See "Identifying Problems" on page 1 to select the appropriate procedure.

Note: Except for reason codes associated with abend failures, you are not likely to encounter an unexpected reason code or return code, *unless you have explicitly called an external interface*. In this case, your program should be able to handle any return codes or reason codes it may receive from the called interfaces. A common reason for explicitly invoking a REXX routine, such as ARXEXEC or ARXINIT, is to initialize a language processor environment.

Diagnostic procedure	References
1. If your program explicitly called the API that returned the code, then check the parameter list passed to the API on the call. If all the addresses and data are correct and there are no other apparent errors on the call, continue with the next step.	See the *VSE/ESA REXX/VSE Reference* for the correct procedure for calling a REXX/VSE service routine. See Chapter 2, "Analyzing Problem Data for REXX Processing" on page 7 for background information on codes from REXX initialization, execution, and termination.
2. Obtain any messages accompanying the return code or reason code. Look at their explanations and take the recommended actions. If you are unable to correct the problem, continue with the next step.	See *VSE/ESA Messages and Codes*, SC33-6607, for operating system messages. See the *VSE/ESA REXX/VSE Reference*, SC33-6642, for REXX error messages.
3. If the code is associated with a REXX module (ARX or IXX), continue with the next step.	

Diagnostic procedure	References
4. Develop a search argument consisting of: • Program identifier: PIDS/5685066 for a REXX code • Message identifier: MS/cccnnnns • The return code: PRCS/hhhhhhhh • The reason code: PRCS/hhhhhhhh Use the search argument to search problem-reporting data bases. If the search finds that the problem has been reported before, request the problem fix. If not, continue with the next step.	See Chapter 3, "Developing a Search Argument" on page 11.
5. Report the problem to IBM, if you need further assistance or if the problem is new. Provide the following problem data: • Problem type: reason code or return code • The search argument • Product name: VSE/ESA Version 2 Release 1 or REXX/VSE Version 6 Release 1 • Return code or reason code • Name and level of the operating system with a list of program temporary fixes (PTFs) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service.	See Chapter 4, "Reporting a Problem to IBM" on page 13.

Chapter 2. Analyzing Problem Data for REXX Processing

This chapter contains background information to help you analyze data about a problem in REXX/VSE. A brief overview of the three main areas of REXX processing introduces this information.

Brief Overview of REXX Processing

REXX processing consists of three main areas, with a REXX routine representing each main area:

- Initialization of a language processor environment (ARXINIT)
- Execution of programs in a language processor environment (ARXEXEC or ARXJCL—the latter calls REXX from JCL)
- Termination of a language processor environment (ARXTERM).

REXX can call each routine internally (without your being aware that REXX called the routine) or externally (you can call the routine through an external interface). The name of the external interface is the same as the name of the REXX routine and the name of the entry point into the routine.

In addition to these three main areas, there are several other areas that affect REXX processing that you can call through an external interface. The *VSE/ESA REXX/VSE Reference* describes these REXX service routines and their return and reason codes.

REXX Initialization (Initializing a Language Processor Environment)

The term *REXX initialization* means the initialization of a language processor environment. Initialization is a crucial part of REXX processing.

REXX/VSE initializes a language processor environment (LPE) when it is needed. VSE/POWER schedules a job to a partition, and this job may or may not be one that calls REXX. If a program calls ARXJCL or ARXEXEC and a language processor environment does not already exist, REXX/VSE automatically creates one.

LPEs are terminated when they are no longer needed (when the job step has completed). When an application is through with an LPE, the application is responsible for terminating the LPE. Only an application that called ARXINIT to create an LPE should terminate the LPE.

The user may be unaware that REXX initialization is occurring. If an error occurs during this “internal” initialization process, the user becomes aware of it only when an abend code or error message is sent to the output stream.

A user can also call REXX initialization directly by calling the ARXINIT initialization routine. In this case, the user must ensure that the program properly handles any return codes or reason codes ARXINIT returns.

The installation has several options to customize the initialization of language processor environments. One option is that you can provide your own parameters module to replace the default module ARXPARMS. The module you provide is then used instead of ARXPARMS to initialize a language processor environment.

See the *VSE/ESA REXX/VSE Reference* for complete information about language processor environments and customizing the environments in which programs run.

The Environment Block

As part of initialization, REXX builds **an environment block**. The environment block is a major REXX control block that contains pointers to other important REXX control blocks. If you encounter a suspected error in REXX code, you need to analyze the environment block to begin gathering diagnostic data from these control blocks.

Errors During Initialization

If an error occurs during initialization, the initialization routine (ARXINIT) returns information in two forms: a return code and a reason code with further information about the error.

If errors occur when running a REXX program, REXX/VSE produces the ARXEXEC return code or the REXX error code. The JCL EXEC command produces message number R003D; see the *VSE/ESA REXX/VSE Reference*, SC33-6642 for details about return codes and REXX error codes.

The information ARXINIT returns is very important for identifying errors in initialization. The *VSE/ESA REXX/VSE Reference* describes the return and reason codes. *This information is particularly valuable if your program has explicitly called ARXINIT*.

REXX Execution (Running REXX Programs)

The main routine that runs REXX programs is ARXEXEC. REXX/VSE calls ARXEXEC when you use the JCL EXEC command or when you use the ARXJCL external interface. User programs can call ARXJCL explicitly. ARXJCL then calls ARXEXEC to run the program. User programs can also explicitly call ARXEXEC to run a REXX program.

Regardless of how it is called, ARXEXEC returns return codes. Return code information from ARXEXEC comes in two forms. The first form is a return code that ARXEXEC returns to its caller in register 15. This return code indicates the processing status of the ARXEXEC routine.

The second form is a numeric return code that ARXEXEC passes back in the EVDATA field of the evaluation block (EVALBLOCK). This numeric return code indicates an error in the program itself, rather than an error during ARXEXEC processing.

For example, when a program runs successfully, ARXEXEC processing returns a zero in register 15. Sometimes, even though the program runs properly, the language processor might encounter a syntax error in the program. A numeric code indicating the syntax error is returned to ARXEXEC's caller in the evaluation block, while the return code in register 15 remains zero.

The numeric code passed back in the evaluation block is in the range of 20003 to 20099. The digits 3 to 99 represent the respective ARX message identifier for the message associated with error. For example, error 26 corresponds to message ARX0026I. The *VSE/ESA REXX/VSE Reference* explains the messages.

The EVDATA field in the EVALBLOCK can contain a value returned from the program when no errors (such as syntax errors) occur during program processing.

For more information about running REXX programs, see the *VSE/ESA REXX/VSE Reference*.

REXX Termination (Ending a Language Processor Environment)

The term *REXX termination* means the process of terminating a language processor environment. The termination routine ARXTERM terminates a language processor environment. REXX/VSE automatically terminates a language processor environment when the program is done. Users can also explicitly call ARXTERM to terminate an environment. For example, if you explicitly call the ARXINIT initialization routine to initialize an environment, you must call ARXTERM to terminate the environment.

If register 0 does not contain the address of an environment block, ARXTERM terminates the last environment created under the current task. If register 0 contains the address of an environment block, ARXTERM terminates the environment to which register 0 points. However, the environment is **not** terminated under any of the following conditions:

- If it was not initialized under the current task
- If there are any active programs running under the environment
- If the environment was the first environment created under the task, but is not the only remaining environment under the task.

Note: REXX/VSE also provides the ARXTERMA termination routine, which terminates all active programs under a language processor environment and optionally terminates the environment. See the *VSE/ESA REXX/VSE Reference* for information about ARXTERMA.

REXX termination processing closes all files opened under the terminating environment. REXX also deletes a data stack, if one was initialized under the environment.

The first environment initialized on a task must be the last environment terminated on the task because of information shared between environments. If a user program tries to terminate the first environment on a task when there are other environments under the first one, the termination attempt fails. In this case, ARXTERM returns a return code of 20 in register 15 and issues a message. The *VSE/ESA REXX/VSE Reference* describes return codes for ARXTERM.

REXX Footprint Description

REXX processing provides a footprinting mechanism to determine which module was in control or which subfunction was running at the time of an abend. Footprint information is maintained in a 48-byte (12 word) footprint data area that contains the CSECT name and module level for the module that is running and for the module that ran previously. The following figure describes the footprint data area.

<----------8 bytes---------><-----------------16 bytes----------------->

Current phase CSECT	Current module level
Previous phase CSECT	Previous module level

Figure 1. Footprint Data Area for REXX Processing

An example of typical module level data is '93183 ARXEXEC'.

The address of the footprint area is in the internal portion of the environment block.

REXX Tracing Functions

REXX provides the following to help you analyze a REXX program, one instruction at a time:

- The TRACE instruction and TRACE built-in function
- The interactive debug facility
- The TS (Trace Start) and TE (Trace End) immediate commands.

The *VSE/ESA REXX/VSE Reference* and *VSE/ESA REXX/VSE User's Guide* describe the use of these facilities.

Chapter 3. Developing a Search Argument

The *VSE/ESA Guide for Solving Problems*, SC33-6610, describes search procedures and arguments. The following tables show symptoms for search arguments (for VSE/ESA with REXX in use). The first table summarizes the symptoms recommended in the diagnostic procedures in Chapter 1, "Diagnosing Problems in REXX Processing." For the symptoms for a particular problem, see the procedure for that problem. The symptoms in each procedure are listed in the order that provides the most efficient search. The second table lists symptoms to add if searches produce too many matches.

Search Argument Symptoms

Description	Free Format Symptom	Structured Format Symptom	Examples
Component identifier	• 568606616 15I — REXX kernel and REXX/VSE Interface • 568606612 15I — REXX/VSE Library.	PIDS/5686066	5686066 PIDS/5686066
Program or phase name	ARXccccc	RIDS/ARXccccc	ARXINIT RIDS/ARXINIT
System abend code	abendhhh	AB/Sxxyy	Abend
Abend reason code	rchhhhhhhh	PRCS/hhhhhhhh	rc0C000111 PRCS/0C000111
Message identifier	msgcccnnnnns	MS/cccnnnnns	msgARX0255E MS/ARX0255E
Return code or reason code	rchhhhhhhh	PRCS/hhhhhhhh	rc0000000E PRCS/0000000E

Additional Search Argument Symptoms

Description	Free Format Symptom	Structured Format Symptom	Examples
Control block field name	cccccccc	FLDS/cccccccc	field SUBCROUT FLDS/SUBCROUT
Other search arguments developed while using the *VSE/ESA Guide for Solving Problems*, SC33-6610.			

Chapter 4. Reporting a Problem to IBM

The checklist in the following table identifies the data that IBM needs for a problem in REXX processing. Collect the data appropriate for the problem type before calling to report a problem.

Problem data:	**Example**
Problem type	Abend
Search argument from Chapter 3, “Developing a Search Argument.”	
Product name, version, and release	REXX/VSE, Version 6, Release 1
Module name and level	ARXINIT at PTF level UY00934
Message identifier and variable data in the message text of all REXX messages associated with the problem	
SVC or ABEND dump, formatted by INFO/ANALYSIS on-line or printed	
REXX return code and reason code	Return code 20, reason code 21
Offset of the failing instruction into the module	0A0
Name and level of the operating system(s) with a list of program temporary fixes (PTFs) applied at the time of the problem and all installation modifications, exits, and products with other than Class A service	Level VSEESA210

Bibliography

This bibliography lists some publications that provide additional information about REXX or the VSE/ESA system.

- *VSE/ESA REXX/VSE Reference*, SC33-6642
- *VSE/ESA REXX/VSE User's Guide*, SC33-6641
- *VSE/ESA System Control Statements*, SC33-6613
- *VSE/POWER Administration and Operation*, SC33-6633
- *VSE/ESA Guide to System Functions*, SC33-6611
- *VSE/ESA Library Guide*, GC33-6619
- *VSE/ESA Messages and Codes*, SC33-6607
- *VSE/ESA Guide for Solving Problems*, SC33-6610
- *VSE/ESA Diagnosis Tools*, SC33-6614
- *SAA Common Programming Interface REXX Level 2 Reference*, SC24-5549
- *IBM Compiler and Library for SAA REXX/370 Release 2: Introducing the Next Step in REXX Programming*, G511-1430
- *IBM Compiler and Library for SAA REXX/370 Release 2 User's Guide and Reference*, SH19-8160
- *IBM Compiler and Library for SAA REXX 370 Release 2 Diagnosis Guide*, SH19-8179.

Index

S

T

Communicating Your Comments to IBM

IBM VSE/Enterprise Systems Architecture
VSE Central Functions
REXX/VSE Diagnosis Reference
Version 6 Release 1

Publication No. SC33-6332-00

If you especially like or dislike anything about this book, please use one of the methods listed below to send your comments to IBM. Whichever method you choose, make sure you send your name, address, and telephone number if you would like a reply.

Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of the book. However, the comments you send should pertain to only the information in this manual and the way in which the information is presented. To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

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

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

- If you prefer to send comments by mail, use the RCF form and either send it postage-paid in the United States, or directly to:

  IBM Deutschland Entwicklung GmbH
  Department 3248
  Schoenaicher Strasse 220
  D-71032 Boeblingen
  Federal Republic of Germany

- If you prefer to send comments by FAX, use this number:
  - (Germany): 07031-16-3456
  - (Other countries): (+49)+7031-16-3456

- If you prefer to send comments electronically, use this network ID:

  INTERNET: s390id@de.ibm.com

Make sure to include the following in your note:

- Title and publication number of this book
- Page number or topic to which your comment applies.

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

IBM VSE/Enterprise Systems Architecture
VSE Central Functions
REXX/VSE Diagnosis Reference
Version 6 Release 1

Publication No. SC33-6332-00

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

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	□	□	□	□	□

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

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	□	□	□	□	□
Complete	□	□	□	□	□
Easy to find	□	□	□	□	□
Easy to understand	□	□	□	□	□
Well organized	□	□	□	□	□
Applicable to your tasks	□	□	□	□	□

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. _____

Readers' Comments — We'd Like to Hear from You
SC33-6332-00

Cut or Fold
Along Line

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

PLACE
POSTAGE
STAMP
HERE

IBM Deutschland Entwicklung GmbH
Department 3248
Schoenaicher Strasse 220
D-71032 Boeblingen
Federal Republic of Germany

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

Cut or Fold
Along Line

File Number: S370/390-37
Program Number: 5686-066

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

SC33-6332-00