

10
4-14-93 JSD

CONF-9303144--1

SLAC-PUB--6066

DE93 011230

REXX
THE UNIVERSAL MACRO AND SCRIPTING LANGUAGE *

CATHIE BURKE DAGER
STANFORD LINEAR ACCELERATOR CENTER
STANFORD UNIVERSITY, STANFORD, CA 94309

ABSTRACT

THE SPEAKER, WHO IS THE FOUNDER OF THE ANNUAL REXX SYMPOSIUM,
WILL SHARE HER PERSPECTIVE ON THE STATUS OF REXX WITHIN AND BEYOND
IBM. IN THIS SESSION, CURRENT USES OF REXX WILL BE CITED AND IMPLICATIONS
FOR THE FUTURE WILL BE EXPLORED.

ACKNOWLEDGEMENTS

The author is grateful to many colleagues for assistance with the content herein and to
Ginger Brower who helped write the paper.

- * Work supported by the Department of Energy contract DE-AC03-76SF00515

PRESENTED AT
SHARE 80
SAN FRANCISCO, CA
MARCH 1 - 5, 1993

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

INTRODUCTION

REXX is a programming language designed by Michael Cowlshaw of IBM's UK Laboratories. In his own words, "REXX is a procedural language that allows programs and algorithms to be written in a clear and structured way."¹

Syntactically, REXX doesn't look that different from any other procedural language. What makes it different from most other languages is its capacity for being used as a macro language for arbitrary applications. Application developers no longer have to design their own macro languages and accompanying interpreters. Instead, they use REXX as the basis for that language and support the REXX programming interface. Then, when a REXX program runs and comes across an expression or function that isn't part of the base language definition, it can ask the application if it knows how to handle it. The application only has to support features that are specific to it, freeing the developer from handling the mundane task of writing a language interpreter. And, if all applications use REXX as their macro language, the user only has to learn one language instead of a dozen.²

Indeed, REXX is an easy-to-use, powerful procedures/macro language. It is IBM's System Applications Architecture (SAA) procedures language and is included in OS/2 1.3 and above. There are REXX implementations for DOS, UNIX, Amiga, and other operating systems, i.e., REXX is a cross-platform language. The American National Standards Institute (ANSI) began a committee to develop a REXX standard. Seeing all this I am convinced that REXX is becoming a universal macro and scripting language.

This paper describes how I developed my REXX perspective, the current REXX world (including the available implementations, applications, and support), and expands my ideas as to why and how REXX will become ubiquitous.

HOW REXX CAME TO SLAC AND ME...

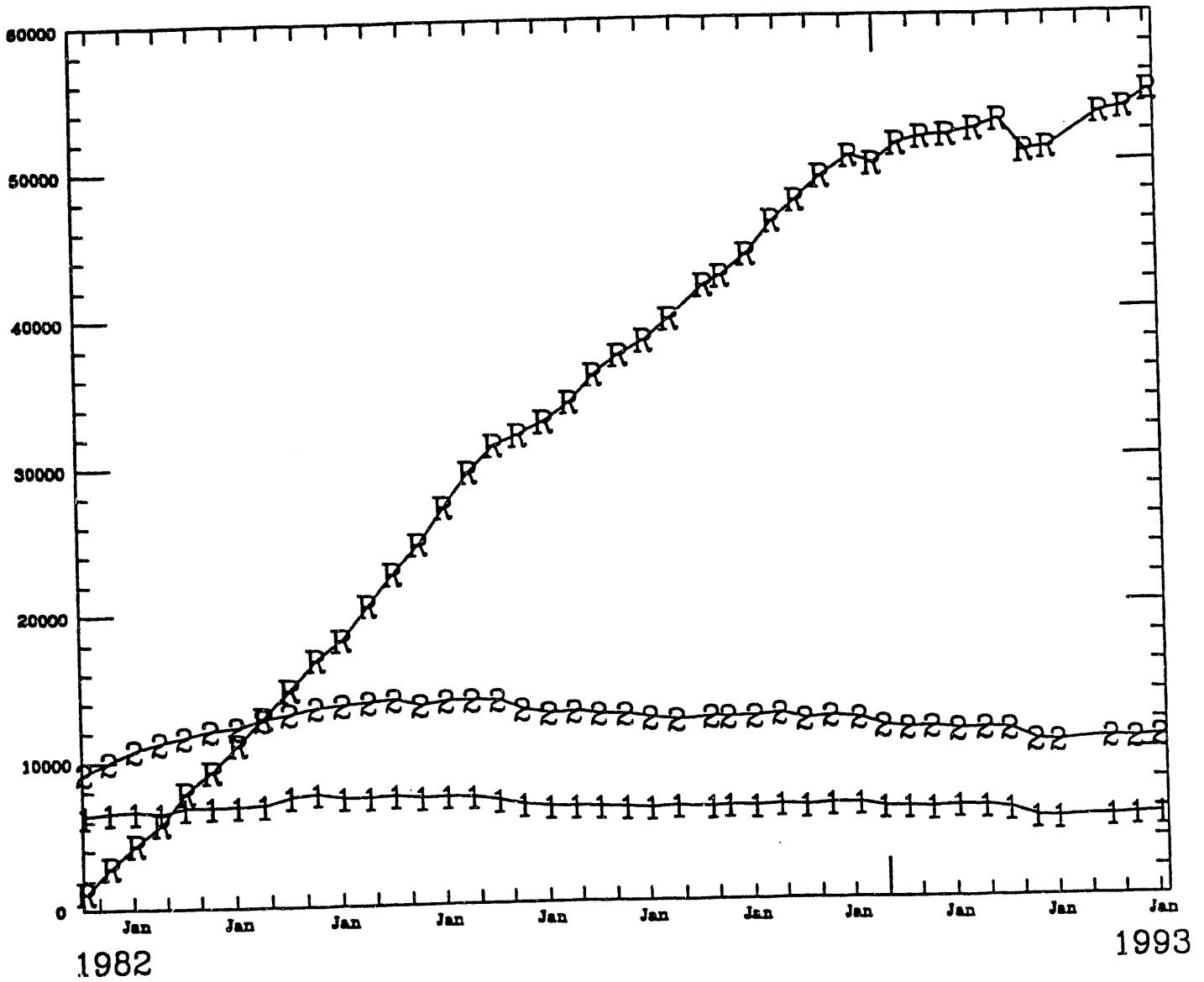
In 1981 at a March SHARE in Houston, Mike first discussed "REX" with people outside of IBM. Ted Johnston, Manager of Systems at Stanford Linear Accelerator (SLAC), attended the session. He realized how much this language could do for SLAC and knew he had to get it for us. So, by March of 1982 SLAC had REX, long before IBM made it a product. Actually, they gave it to us with the understanding that it might *never* become a product. Soon, we were doing much of our work in REX and told IBM we couldn't get along without it. (See Figure 1) We were so dependent upon it that we told IBM if they did not make it a product we would maintain it ourselves. The rest is known--REX became REXX, an IBM product released with VM in 1983, and has continued to grow in popularity and use ever since.

¹Mike Cowlshaw, IBM

²Eric Giguere, University of Waterloo

- FIGURE 1 -

Stanford Linear Accelerator Center Exec Files



Number of exec files total

1 - Exec 1 2 - Exec 2 R - REXX

NEW VERSIONS AND A GATHERING....

By the mid 80s, I was moving some of my work from VM to a PC. In doing this, I realized that REXX had a great future with PCs because DOS had nothing but a very primitive batch language. Beyond SLAC, I noticed REXX enjoyed slow but growing acceptance in the PC world. I encouraged people to speak about REXX at PC conferences but there was little interest. Then, exactly three years ago at SHARE in Anaheim, I decided to instigate a gathering of REXX developers and users. By then I knew of two versions of REXX for DOS, one for Unix, one for the Tandem, and ARexx for the Amiga. The implementors had never met and I thought they would enjoy meeting each other and that their meeting would draw a crowd of other REXX aficionados. Mike Cowlshaw agreed to participate and my managers at SLAC, Bill Johnson and Chuck Dickens, agreed to support my effort to organize such a meeting. (Bill says he doesn't remember, but he chose the name *REXX Symposium*.)

We chose a date to fit Mike's schedule, then I contacted all of the known implementers and they all agreed to come. We held that first one-day meeting in the SLAC auditorium. It was advertised electronically and people spread the word. The majority of the approximately 150 people attending were from the Bay Area.

Since then, we've had annual two-day meetings (the 1993 meeting will last three days), we've added many more speakers and topics, and we now address an international audience. Each year we have had exciting news. The first year, we had a REXX for Unix demonstration and Bob O'Hara gave the famous *Why REXX Died* speech that he repeated later at SHARE. That speech was part of what helped to get REXX embedded in OS/2 1.3. The second year we had a demonstration of IBM's object-oriented REXX and a panel discussion about REXX in Unix. The third year, there were two presentations on REXX for Windows, three new implementers announced free versions for Unix, and Borland and Lotus sent speakers to demonstrate their products that use REXX. This year we will hear about the Macintosh version, three OS/2 visual versions, and probably a few surprises.

People who write REXX programs or implementations have a particular point of view. I've done REXX programming in the past, but currently, I consider myself a REXX observer and advocate. Through my work on the Symposium and subsequent contact with the REXX developers and user community, I have an overview of REXX activity. Being involved has given me a unique perspective on the growth of REXX and I've seen great developments: the first Symposium in 1990 served as a springboard for the ANSI effort, the Share REXX Project has grown in influence, and many people have worked to spread REXX throughout the world.

Once, Gabe Goldberg called me a REXX *spark plug*. I appreciated the compliment and I'm here to line up other spark plugs to help REXX become the universal scripting and macro language it's so well suited to be.

REXX IMPLEMENTATIONS AND PRODUCTS

There are numerous REXX implementations and products, both commercial and free. I am aware of the following:

COMMERCIALY AVAILABLE IMPLEMENTATIONS OF REXX:

There are many commercial implementations of REXX. For example, there are stand-alone versions for various UNIX platforms, the Macintosh, VMS, DOS, Windows, OS/2, and Tandem. REXX interpreters come bundled with the Amiga (Workbench 2.0 and higher) and embedded in the IBM VM, TSO, OS/2 and AS/400 operating systems. IBM has a single compiler which will work on both MVS and VM.³ Also, there are other companies that offer compilers. See Appendix A for addresses of the companies that sell these implementations.

AMIGA INFLUENCE

I do not want to overlook the influence of the Amiga community on the growth of REXX. In early 1991 Commodore began bundling it with the operating system, and the Amiga users were the first beyond IBM to exploit its use system-wide. The Amiga developers all know that they have REXX available and, therefore, they depend upon it. Since the first REXX Symposium in 1990, Amiga applications have continued to lead the personal computer world in showing the power of REXX, particularly when it is an integral part of the operating system.

FREE IMPLEMENTATIONS OF REXX:

Currently there are three freely-distributable interpreters (with source) for Unix machines, all available for anonymous FTP from rexx.uwaterloo.ca:

/pub/freerexx/regina/regina-0.04a.tar.Z	254939	ANSI C
/pub/freerexx/alrexx/rx102.tar.Z	671904	C++ (using GNU G++)
/pub/freerexx/imc/rexx-imc-1.3.tar.Z	491691	ANSI C

To compile, the first two interpreters require Lex and Yacc (or suitable clones). All are still beta versions, though imc, which runs only on SunOS machines has been around the longest and claims to be quite stable.

IBM has made available VREXX, a simple "visual" REXX system for OS/2. It's a function library that allows you to put up simple windows and some simple dialogs from any REXX program running in an OS/2 session. It is currently available for FTP from software.ibm.watson.com as the file /pub/os2/vrexx2.zip, and can also be found on

³ Linda Suskind Green, IBM.

ftp-os2.nmsu.edu. Note that the term "Visual REXX" is used pretty loosely here. It's really more like the various UI libraries that are available for use with AREXX on the Amiga.⁴

Also, the Workstaion Group offers a free version of uniREXX, their REXX for Unix, for academic institutions.

Non-IBM Products and Applications

WRITTEN IN REXX:

Batch	SLAC
Netnews	Linda Littleton
RDM	University of Victoria
RiceMail	Richard Shafer
Search4	Larry Chase
Trucker	Simon Husin

USING REXX:

Amiga Applications List	Daniel Barrett	
AFOperator	Candle	
AFRemote	Candle	
Automate	Legent	
CC:Mail	Lotus Development	Macros, Scripting
DB/REXX	VM Systems Group	
EdWord	Trax Softworks	Macros
ESS	Trax Softworks	Macros
JobTrace	Legent	
Kedit	Mansfield Software	Macros
KPROBE	VM Systems Group	
Notes	Lotus Development	
Objectvision	Borland International	Macros
Ops MVS	Legent	
Oracle Pro REXX	Oracle	Macros
REX/Fullscreen	REXXpert Systems	
REX/VSAM	REXXpert Systems	
REXXTerm	Quercus Systems	Scripting
RLX (REXX Lang. Exts.)	Relational Architects	
RXD	IBM EWS	Debugger
SPFPC	Command Technology	Macros

⁴Eric Giguere, University of Waterloo

TE/2	Oberon	Scripting
THE	Mark Hessling	
Tritus SPF	Tritus	Macros
Uni-SPF	Workstation Group	Macros
Uni-Xedit	Workstation Group	Macros
VL T	Willy Langeveld	Macros, Scripting
VM Dialout	Trax Softworks	Scripting
VMSTAGE	HEPVM	
1-2-3 for OS/2	Lotus Development	Macros

TRANSLATING REXX:

REXXtacy	Ruddock and Associates	REXX to C
----------	------------------------	-----------

ADD-ON PRODUCTS:

FaAA_LSIE/VM	Failure Analysis
REXXLIB	Quercus Systems
REXXArpLib	Willy Langeveld, SLAC
REXXMathLib	Willy Langeveld, SLAC
REXX Protect	Sidney
REXX Rainbow Library	Dineen Edwards Group

SUPPORT ACTIVITIES

REXX REFERENCE BOOK:

The REXX reference book is Mike Cowlshaw's *The REXX Language: A Practical Approach To Programming*, currently in its second edition and often referred to as *TRL*. Until the ANSI committee releases its standard, *TRL* defines REXX. (See the bibliography below for details on *TRL*.)

REXX STANDARDS:

The American National Standards Institute (ANSI) sets national standards for programming languages as well as for various other things in the United States. It established the X3J18 working group to define a formal standard for the REXX language using *TRL* as its base document. X3J18 meets three or four times a year and holds ongoing discussions throughout the year. The members are mostly REXX implementers and users, but anyone may participate. For more information on X3J18 contact Brian Marks (chair), marks@winvmd.vnet.ibm.com or Neil Milsted (vice-chair), nfm@wrkgrp.com.

CONFERENCES:

The SHARE REXX Project

SHARE is a large IBM user group which includes an active REXX Languages project. The participants provide IBM with language requirements from the user community.

The REXX Symposium

The REXX Symposium is an annual gathering for REXX users and implementers. The Stanford Linear Accelerator Center (SLAC), a heavy user of REXX on several platforms, hosts this yearly event. The 1993 Symposium is being held in San Diego, California from May 18th to 20th. For more information about the May meeting contact one of the following:

Cathie Dager	cathie@slac.stanford.edu.
Forrest Garnett	garnett@vnet.ibm.com
Jim Weissman	jhw@cup.portal.com
Bebo White	bebo@slac.stanford.edu

If you want a copy of the 1992 REXX Symposium Proceedings, send a request containing the following line of text: Subject: "REXX Symposium Proceedings, SLAC Report 401". Include (left justified, please) your complete postal address in the request.

Send this information by e-mail to techpub@slac.stanford.edu, or by postal mail to

Publications Department
SLAC, MS 68
PO Box 4309
Stanford, CA 94309

Others:

REXX is also discussed at a variety of other conferences, including the Windows and OS/2 Conference, the Software Development Conference, and OS/2 technical conferences.

ELECTRONIC INFORMATION:

Newsgroups and mailing lists:

VM SHARE

CompuServe

Internet/Usenet On Usenet the group comp.lang.rexx exists for discussion of REXX in all its variations. Currently the comp.lang.rexx newsgroup is also gatewayed with the REXXLIST mailing list (i.e., anything that gets posted to the news group is sent to everyone on the mailing list and anything that is sent to the mailing list gets posted to the news group).

REXXLIST

FTP SITES

REXX-related materials for specific systems can also be found on FTP sites serving those systems:

Small general REXX repository	rexw.uwaterloo.ca
Amiga	wuarchive.wustl.edu
OS/2	ftp-os2.nmsu.edu
OS/2 official IBM support	software.watson.ibm.com

THE REXX FREQUENTLY ASKED QUESTIONS LIST (REXX FAQ)

REXX FAQ contains general information about REXX, pointers to other sources of information, and a list of questions and answers. The file is maintained by Eric Giguere (giguere@csg.uwaterloo.ca) and is posted at regular intervals to Internet/Usenet comp.lang.rexx. (It is also available on VM via NetNews.)

REXXLIB

REXXLIB is a collection of REXX library routines contributed by various people and is accessible from Bitnet and from Internet.

BIBLIOGRAPHIES:

The complete REXX bibliography is in Linda Suskind Green's chapter in the *REXX Handbook*. This bibliography contains references to all of the REXX books, reference manuals, user guides, and articles. Send any updates to the bibliography to Linda Green at greensl@gdlvm7.vnet.ibm.com. The following is a partial list⁵ of REXX books (not including reference manuals or user guides):

The REXX Language, A Practical Approach to Programing, Mike Cowlshaw (1985,1990)
Modern Programming Using REXX, Bob O'Hara and Dave Gomberg(1985,1988)
REXX in the TSO Environment, Gabriel F. Gargiulo (1990)
Practical Usage of REXX, Anthony Rudd (1990)
Using ARexx on the Amiga, Chris Zamara and Nick Sullivan (1991)
Amiga Programmers Guide to ARexx, Eric Giguere (1991)
REXX Handbook, edited by Gabe Goldberg and Phil Smith (1992)
Programming in REXX, Charles Daney (1992)
REXX--Advanced Techniques for Programmers, Peter Kiesel (1992)
The ARexx Cookbook, Merrill Callaway (1992)
Command Language Cookbook, Hal German (1992)
REXX Tools and Techniques, Barry Nirmal (1993)

⁵Linda Suskind Green, IBM

FUTURE

I've presented the REXX implementations and activities I am aware of; what about the future? I foresee:

- *Free DOS implementation from Anders Christensen*
- *Three NT implementations*
- *Additional work on free versions for UNIX*
- *More for the Macintosh*
- *IBM Object-Oriented REXX and VREXX Enhancements*
- *AIX REXX and AIX REXX Compiler (unconfirmed)*
- *REXX Journal*
- *Acceptance by Borland and Novell*
- *REXX User Organization*

COMMERCIAL PRODUCTS FOR USING PM WITH REXX

There are two products announced and soon to be available which offer "visual" REXX:

<i>Ucandu</i>	Dave Hock	(919) 387-7391
<i>Watcom</i>	Terry Stepien	(519) 886-3700

REXXHORTATIONS

Is REXX a good candidate for a universal macro and scripting language? The answer is a definitive "yes." REXX exists on numerous platforms and is used by a growing number of products. It is expanding in popularity among university students. (Several implementations for Unix and the one for the Macintosh were done by students.) It's technically sound and well-suited for continued growth. Actually, I predict an explosion of REXX awareness. How will this happen?

Microsoft is a unified force with a strategy to promote Visual Basic as a universal macro and scripting language. I am not aware of a unified effort to promote REXX, at least not an effort with the same commitment. If Mike Cowlishaw were CEO of IBM and was chair of the government committee that gave us ADA, he could issue a REXX ultimatum, but that's not the case. I predict that REXX will proliferate because--just like Ted Johnston and others at SLAC belived more than a decade ago--*It's too good for us to do without it.*

At SLAC our early experiences with moving REXX programs to Unix have proved quite succesful. Recently Les Cottrell, our networking manager ported a 1000 line CMS/REXX program for looking at Ethernet topology information in a data base and creating various reports. After two days it was essentially ported (that included understanding how to use uniREXX and the differences compared to CMS/REXX.) He added 140 lines of code so that the program runs on both CMS and AIX/6000. (Refer to Figure 2 for the code and his notes on timing comparisions.)

Chuck Boeheim, our systems software manager, moved a VM REXX program to Unix and was pleased that the conversion took only an hour. Also he ported REXXTRY to Unix and it ran unmodified. These are positive experiences with non-trivial code and indicate the likelihood of tremendous growth of REXX usage in Unix at SLAC, as well as elsewhere.

So my question to you is "Does REXX help you get your work done?" If so, are you willing to make the effort to ensure your continued use of REXX?

- Tell your coworkers about REXX tips and techniques.
- Tell your management that you need REXX when you move to Unix, AIX, Windowss, OS/2 or NT. Be aware that your REXX skills are transferable.
- Participate in the SHARE REXX Project, the REXX Symposium, and other activities.
- Use the networks, mailing lists, forums, bulletin boards to share your ideas and questions with others.
- Write REXX articles for publication.
- Respond to magazine articles with letters to the editor about the universality of REXX.

- Figure 2 -

Les Cottrell's Notes on Converting a REXX CMS Program to uniREXX

```
/* ***** */
/* Notes on conversion of a 1000 line REXX program from CMS to Unix: */
/* Code is supposed to run in both VM & Unix environments */
/* Check OS environment by using: */
/* PARSE SOURCE Environ . */
/* The result is Environ='UNIX' or Environ='CMS' */
/* Provide a LINEOUT function called LINEOUTF to replace EXECIO */
/* 'EXECIO 1 DISKW' filename ( STRING' String gets replaced by */
/* CALL LINEOUT filename, String */
/* 'EXECIO * DISKR' filename '( STEM LINE.' replaced by use of */
/* Line.0=0 */
/* DO L=1 BY 1 WHILE LINES(filename) */
/* Line.0=Line.0+1; Line.L=LINEIN(filename) */
/* END L */
/* 'EXECIO 0 DISKW' filename '( FINIS' {close file} replace by */
/* CALL LINEOUT filename */
/* 'EXECIO' N 'DISKW' filename { Writes N lines from the queue} */
/* is replaced by: */
/* DO L=1 TO N; */
/* PULL String; */
/* Fail=LINEOUT(filename, String); */
/* END L */
/* Use of ADDRESS COMMAND 'ESTATE' Filename to see if file exists */
/* replaced by use of LINES(filename) {if 0 result then does not */
/* exist is OK for the current program} */
/* 'ERASE' Filename is replaced by CALL POPEN 'rm' Filename */
/* 'PIPE <' filename '| COUNT LINES | STEM DevType.' */
/* gets replaced by: */
/* DevType.0=QUEUED() */
/* CALL POPEN 'cat' IPFile '| wc -l'; */
/* DevType.0=QUEUED()-DevType.0 */
/* DO Q=1 TO DevType.0; PULL DevType.Q; END Q */
/* If in Unix do not execute GIMES or DROPS, instead one will */
/* need to use the fully qualified path name, e.g. */
/* /nfs/slacvm/ascii/cottrell.191/profile.exec or */
/* /u/sf/cottrell/.cshrc */
/* CMS filenames consist of fn ft fm and are in capitals with */
/* fn ft fm being separated by spaces. The Unix equivalent */
/* is fn.ft with no spaces and usually all lower case. */
/* Presuming that the variable Filespec holds a CMS file name */
/* of the form NAME TYPE A, then the following inserts a */
/* period between NAME & TYPE and coverts to lower case: */
/* PARSE VAR Filespec.F Uname Utype . */
/* Filespec.F=Uname.'Utype */
/* Filespec.F=TRANSLATE(Filespec.F,, */
/* 'abcdefghijklmnopqrstuvwxyz',, */
/* 'ABCDEFGHIJKLMNOPQRSTUVWXYZ') */
/* Not sure what to do with MAKEBUF & DROPBUF at the moment. */
/* */
```

```

/* Results:
/* About 2 days to convert EXEC, including devising ways to do it
/* and learning UniREXX.
/*
/* Number of references to Environ variable=40, number of extra
/* lines of code to make run under Unix as well as CMS ~ 140
/*
/* Times (secs):  ----- RS/6000 -----   Sun   CMS       Unix/CMS*/
/*                320H  340   550   570   Sparc2 9000-580 320H/CMS*/
/* Elapsed time  77.12 64.70 65.63 34.57 89.61  125.0   0.62 */
/* System Time   5.22  4.4   2.99  1.43  7.85   0.35   15   */
/* User Time     51.41 40.59 31.92 20.72 58.99  21.74   2.4 */
/*
/* More Information:
/* "The Rexx Language" by Mike Cowlshaw, Prentice Hall.
/* uni-REXX readme file {tells how to invoke, I/O functions, Unix-
/* specific functions, limitations, external functions, APIs ...}
/* REXX FAQ available via anon FTP from rexx.uwaterloo.ca, cd pub
/* get rexxfaq.
/* Netnews group comp.lang.rexx
/* *****

```

APPENDIX A
Commercial REXX Interpreters and Compilers

IBM VM/CMS, MVS/TSO, AS/400:
Contact your local support person.

IBM OS/2:
(800) 3IBMOS2
(800) 465-7999 in Canada

DOS, Windows, and OS/2:
Quercus Systems
Box 2157
Saratoga, CA 95070
(408) 867-7399

DOS, Windows, Tandem
Kilowatt Software
Box 209
Reynoldsburg, Ohio 43068
(800) 848-9470 (US only)
(614) 866-4300

Amiga:
Commodore
Wilson Drive
Westchester, PA 19380
(215) 431-9100

Various UNIX platforms and VMS:
The Workstation Group
River Road
Rosemont, IL 60018
(800) 228-0255 (US only)

Amiga Compiler:
Dineen Edwards Groups
W. 12 Mile Rd., Suite 305
Southfield, MI 48076-2553
(313) 352-4288

MVS/TSO and VM/CMS Compilers:
Contact your local IBM representative.
Program numbers:
5695-013 (compiler)
5695-014 (library)

MVS Compiler
Open Software, Inc.
1000 Savage Court, Suite 102
Longwood, FL 32750
(407) 834-2822

VM Compiler:
Systems Center
Alexander Bell Dr.
Reston, VA 22091
(703) 264-8000

END

**DATE
FILMED**

6 / 17 / 93

