

The REXX SourceBook

Frequently Asked Questions About REXX

Last Revised: July 22, 1993

Eric Giguère

`giguere@watcom.on.ca`

The REXX SourceBook

Copyright Information

This document is copyright ©1993 by Eric Giguère. Permission is granted to reproduce and distribute all or part of this document for non-commercial purposes only. All other uses must first be cleared with the author. The author may be contacted on the Internet at the address giguere@watcom.on.ca or on paper by writing to WATCOM International, 415 Phillip Street, Waterloo, Ontario, Canada, N2L 3X2. Please note, however, that this document is not published or endorsed by WATCOM.

Introduction

This document is intended to serve as a useful reference for REXX-related information. It aims for *breadth* as opposed to *depth*, and references to other material are given where appropriate. Suggestions and updates should be sent to the author in an attempt to keep this document relevant and up-to-date.

A. What Is REXX?

REXX is a programming language designed by Michael Cowlshaw of IBM 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."

REXX doesn't look that different from any other procedural language. Here's a simple REXX program:

```
/* Count some numbers */  
  
say "Counting..."  
do i = 1 to 10  
    say "Number" i  
end
```

What makes REXX different from most other languages is that it is also designed to be used as a **macro language** by arbitrary application programs. The idea is that application developers don't have to design their own macro languages and interpreters. Instead they use REXX as the macro language and support the REXX programming interface. If a REXX macro comes across an expression or function call that it cannot resolve, it can ask the application to handle it instead. The application only has to support the features that are specific to it, freeing the developer from handling the mundane (and time-consuming) 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.

B. REXX and the Internet

Networks connect computers in various ways for the exchange of data. The terminology is a bit confusing to the new user. Here are the definitions this document uses:

Usenet: Not really a network, just the set of machines that exchange network news. Network news is really an extended form of electronic mail that groups messages from individuals into newsgroups that users can read using special newsreaders.

Internet: The worldwide network based on TCP/IP protocols. Besides being able to receive mail and newsgroups, these machines can use programs like **ftp** and **telnet** to communicate with other machines in real time. Most Internet machines are Unix-based.

BITNET: The worldwide network that connects many IBM mainframes. BITNET users can also transfer files using methods that are incompatible with those of the Internet.

Newsgroups

The Usenet group **comp.lang.rexx** exists for discussion of REXX in all its variations. Anything posted to this newsgroup also gets sent to the REXXLIST mailing list (see below) and vice-versa. Other newsgroups of interest are machine-specific. Recommended groups are **comp.os.os2.programmer** and **comp.sys.amiga.programmer**.

FTP Sites of Interest

FTP is a file transmission protocol used on the Internet to transfer files between machines. The transfers are done in real time and usually require that the user have an account on both machines. However, many machines on the Internet support what is known as **anonymous FTP**, which allows users on other machines access to a limited set of files without requiring an account. Some of the more interesting sites that offer this service are:

rexx.uwaterloo.ca

General repository for REXX-related information, including free REXX interpreters for Unix and DOS. An XEDIT clone for Unix and OS/2 may also be found here. Look under /pub/rexx.