ARexxGuide

COLLABORATORS

	TITLE : ARexxGuide	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		June 16, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

ARexxGuide

1.1 ARexxGuide Copyright © 1993, Robin Evans

```
                  AN AMIGAGUIDE® TO ARexx                  Edition: 1.0a
by Robin Evans


              Registration

              Comprehensive contents
                 Introduction
              Tutorial
                Reference
 Basic elements           Instructions            Functions
 Guide to the powders     Syntax & explanation    Syntax & explanation
 & potions in the ARexx   of keywords and         of built-in & support
 chemistry set.           instructions            functions.
 Operators                Commands
 Glue for arithmetic,     Utility programs.
 comparison, & logical
 expressions.

 INDEX
```


1.2 ARexxGuide | Registration (1 of 2) | SEND MONEY NOW!

This guide is shareware. Please use it and pass it on (in its original archive) to your friends and acquaintances. If you learn something from the guide or find it helpful in writing ARexx programs, then please take a moment to fill out the registration card and send in the fee.

Shareware is similar to public broadcasting in this: You can watch or listen to the stations without sending them any money, just as you can use shareware application. It might not make any difference. Your favorite

programs might stay on the air whether you call in your pledge or not. Someone else will still release a cool shareware offering. But if nobody sends money, the station won't stay on the the air for long. If nobody sends money, the pool of shareware will begin to dry up. And if you don't send money, the station won't give much attention to your complaints or suggestions about its programming.

Public broadcast stations have developed guilt-inducing pledge drives into something of an art. I can't even approach that level of begging, so please try to remember the last pledge drive you heard on an NPR or PBS station and imagine that all of those folks were talking about this guide. Imagine that and the send in your pledge now.

Register and send pledge now!
Please send comments or requests to any of the following on-line addresses:

robin@halcyon.com	on Internet
R.EVANS6	on GEnie
r.evans6@genie.geis.com	on Internet

I will upload additions to the tutorial sections in the future and will also send any revised editions of files included here to the same places where this archive was originally posted. Watch for them.

1.3 ARexxGuide | Registration (2 of 2) | THE PITCH

The pitch
~~~~~~~~~

The basic registration fee for this guide is $15.00. For that you get the complete reference you see before you now -- information you would pay $25 to $40 for if it were presented in traditional book form.

But wait... There's more.

For a not-limited time, those who send in a premium registration fee of $21 will receive, by mail within eight weeks, a nicely-printed quick reference guide that includes the syntax guides presented here in the function reference and in the instruction reference.

Sorry, folks, we don't have operators standing by to take your order, but we do have -- yes, standing by -- representatives of the worlds's postal services who will deliver your registration fee to the address below.

To make things easier, and to provide information helpful in making future revisions to this guide, the button below will guide you through an on-line registration form that can be printed immediately or saved to disk.

Fill out registration form
*
Send registration fee (and optional form) to:
Robin Evans
1020 Seneca #405
~~~~~~~~~

```
                  Seattle WA 98101-2720

      And thank you, very much.

One more note: There may be good reasons for not sending in the shareware
fee. Whatever the reasons, though, they should also be good enough reasons
to delete this material from your disks. Please take the extra step of
doing that if you decide against registration.

Next: ARexxGuide contents | Prev: Registration intro | Contents: main
```

1.4 ARexxGuide | CONTENTS

```
            Numeric precision
         Strings
            Treating numbers as strings
         Variables
            Using variables
            Compound variables
               Overview: Using compound variables
               Stem variables
               Extending stem variables
               Substituting values in compound variables
               Using strings as the derived name of a branch
               Setting the default value of a compound variable
               Finding values in a compound variable
            Special variables
               RC
               RESULT
               SIGL
            \
         Function calls
            Internal functions
            Built-in functions
            Library/Host functions
            External functions
            Function arguments
         Operations
            Concatenation
            Arithmetic
            Comparative
            Logical
            Conditional expressions
         Avoiding accidental commands from expressions
      Clauses
         Assignment clauses
         Instructions
         Command clauses
            Command host: what is it?
            The default host
            Determining the initial host
            Entering commands in a script
            Example script
         Label clauses
         Null clauses

Instruction keywords:
      ADDRESS
      ARG
      BREAK
         Breaking structure
      CALL
      DO
         <number>
         Index variable/TO/BY
         FOR
         WHILE/UNTIL
         FOREVER
         END
      DROP
```

```
ECHO
EXIT
IF
   ELSE
INTERPRET
ITERATE
LEAVE
NOP
NUMERIC
OPTIONS
PARSE
   ARG
   EXTERNAL
   NUMERIC
   PULL
   SOURCE
   VALUE <expression> WITH
   VAR
   VERSION
   Templates
      Tokenization
         The period- placeholder token
      Pattern markers
      Positional markers
      Using variables as template markers
      Combining different types of markers
      Using multiple templates
PROCEDURE
   EXPOSE
PULL
PUSH
QUEUE
   PUSH, QUEUE and REXX data-stream I/O
RETURN
SAY
SELECT
   WHEN
   OTHERWISE
SIGNAL
   ON | OFF <interrupt>
      BREAK_C
      | BREAK_D
      | BREAK_E
      | BREAK_F
      ERROR
      FAILURE
      HALT
      IOERR
      NOVALUE
      SYNTAX
   <label name>
TRACE
  Trace options
  Interactive tracing-- ?
  Command inhibition-- !
UPPER
```

```
ARexx functions:
       Comparison functions
          ABBREV
          COMPARE
          FIND
          INDEX
          LASTPOS
             Locating file names with LASTPOS() and MAX()
          POS
          VERIFY
             Checking unique datatypes with VERIFY()
       String manipulation
          CENTER
          COMPRESS
             Counting characters using COMPRESS()
          COPIES
          DELSTR
          INSERT
          LEFT
             Formatting output with RIGHT(), LEFT(), and TRUNC()
          LENGTH
          OVERLAY
          REVERSE
          RIGHT
          STRIP
          SUBSTR
          TRANSLATE
          TRIM
          UPPER
          XRANGE
       Word manipulation
          DELWORD
          SPACE
          SUBWORD
          WORD
          WORDINDEX
          WORDLENGTH
          WORDS
       Char/Num translation
          B2C
          C2B
          C2D
          C2X
          D2C
          D2X
          X2C
          X2D
       Number manipulation
          ABS
          HASH
          MAX
          MIN
          RANDOM
          RANDU
          SIGN
          TRUNC
       Informational
```

```
          REPLY
          TYPEPKT
          WAITPKT
       Memory management
          ALLOCMEM
          BADDR
          EXPORT
          FORBID
          FREEMEM
          FREESPACE
          GETSPACE
          IMPORT
          NEXT
          NULL
          OFFSET
          PERMIT
          STORAGE
       Bit-wise operations
          BITAND
          BITCHG
          BITCLR
          BITCOMP
          BITOR
          BITSET
          BITTST
          BITXOR

ARexx operators:
       Concatenation
       Arithmetic
          Table of arithmetic operators
       Comparison
          Table of comparison operators
       Logical
          Table of logical operators
       About operator precedence

AmigaDOS command programs:
       RexxMast
       RXC
       RX
       HI
       RXLIB
       RXSET
       TCO
       TCC
       TS
       TE
       WaitForPort

   Useful tools:
       WShell
       ExecIO


                GLOSSARY
                INDEX
```

```
Interactive examples
                 *
                    Registration form
    Comparison demonstration
    NUMERIC demonstration
    TRACE demonstration
    Standard I/O demonstration
    Break-key demonstration
```

1.5 ARexxGuide | TURORIAL

```
                   AN AMIGAGUIDE® TO ARexx                 Edition: 1.0a
   by Robin Evans

    UnCrunch.rexx  A shell-based program to undo archives made
                   with different archiving utilities.

   Each line of code in the program is explained in the accompanying
   tutorial sections.

   More tutorials will be made available in the future. Be sure to specify
   what you'd like to see covered when you register.


                Copyright © 1993, Robin Evans.  All rights reserved.

    This guide is
                shareware
               . If you find it useful, please register.
```

1.6 ARexxGuide | GLOSSARY (Press -Retrace- to return to previous node)

```
Boolean value        Either true or false, which -- in ARexx -- is
      considered to be 1 for true and 0 for false. Named after the
      mathematician George Boole.

CON:                 Or: Console Window. A  logical device  that creates
      a text window on the Workbench or other public screen. This device
      cab be used as the <filename> with the  file I/O  functions to
      direct output to a window opened by the script.

Control structure    A programming construct that allows a series of
      statements to be executed as part of a block. The instructions  DO ,
       SELECT , and  IF  create control structures in ARexx.

Debug                To search for and eliminate (eventually) problems or
      'bugs' in a program. The  TRACE  instruction aids debugging in ARexx.

Dyadic               Having two parts. In ARexx, the term refers to
```

operations that have two operands (2 + 2, for instance). Some operations have only one operation (-1, for instance) and are referred to here as 'prefix' operations. The more technical name for the opposite of a dyadic operation is unary operation.

Egregious — It means "very bad," but use of this word shows that the writer has spent too much time in the company of lawyers. (Which may be the same thing, come to think of it.)

Interpreter — A program that translates source code (the program lines you write) into machine instructions. It does that each time the program is run. RexxMast is ARexx interpreter program.

I/O — Input/Output. The term refers to the various ways of obtaining data and displaying or saving it. The I/O system on the Amiga includes disk drives, windows, and requesters.

Iteration — A program-ese synonym for 'repetition'. To a human the instruction to "Do forever" would be a Sysephean punishment. To a computer, it is just another task. In ARexx, iteration is performed by a single instruction, DO , which has a wide range of options to give the programmer control over when the iteration stops.

Keyword — The word that identifies an ARexx instruction or the option to an instruction. Keywords and instructions are detailed in the Instruction reference .

Logical device — A part of the computer system defined through software. In AmigaDOS, logical devices intervene between the application program (including ARexx) and such hardware devices as disk drives, printers, and the monitor screen.

Loop — A section of program code that is repeated (or iterated). Looping in ARexx is controlled by the DO instruction.

Mantra — In Hinduism, a sacred formula, repeated over and over again, that is believed to posses special power. (Looking up this word demonstrates one of two things: either the user wasn't around for the 60's or wasn't paying attention. < put ;-) smiley here >)

Nested — To place one thing within another just as an egg is placed in a bird's nest. A nested function is one function used as an argument to another function as in RIGHT(TRUNC(Amount, 2), 6). Here the TRUNC() function, which truncates the decimal points on a number, is nested within the RIGHT() function, which right-justifies the resulting number.

NIL: — A logical device recognized by AmigaDOS that will throw away input or output directed to it.

Preferences — A series of programs that are part of the Amiga OS. They allow the user to customize most aspects of the system.

Prototyping — The process of developing an initial version of a software application in one language to test the logic of the code and the usefulness of contemplated options.

PRT: A logical device recognized by AmigaDOS that directs output to the printer currently defined in Preferences . This device can be used as the <filename> with the file I/O functions to print material from an ARexx script.

Reserved A token that serves a specialized purpose in the language and cannot be used for any other purpose. REXX has a limited set of reserved tokens. The single characters representing operators and special characters are reserved in all situations. Instruction keywords and sub-keywords are reserved only within the limited range of the instruction itself. The variables [x] and [b] -- although they are not technically reserved -- should be avoided because of possible conflicts with hex and binary strings .

STDERR Standard error device. This is the logical name assigned to a device to which ARexx will send error messages and the output of tracing . If the trace console is open, that will become STDERR. The PARSE EXTERNAL instruction retrieves input from this device.

STDIN Standard input device. This is the logical name assigned to a device from which ARexx will retrieve input then the PARSE PULL instruction is used. It is usually the shell from which a program was launched, although a script started from another environment will often have STDIN assigned to NIL: .

STDOUT Standard output device. This is the logical name of the device to which ARexx will output the expression used in a SAY instruction. It is usually the shell from which a program was launched, although a script started from another environment will often have STDOUT assigned to NIL: .

Subroutine A section of code separated from the main body of a program. In ARexx, subroutines are identified by labels and usually serve as internal functions .

1.7 ARexxGuide | Interactive examples -- Requirements

The registration form and several interactive examples scattered throughout ARexxGuide use ARexx scripts to provide the interactive environment. Because they must try to run a script, the buttons impose some extra requirements:

1. The RX command must be located in a directory that is included in the Workbench command search path.

The RX command is included in a special directory, "rexxc," on Workbench disks distributed by Commodore. That directory is not part of the search path in the standard Startup-Sequence files. The user has two choices: either add "sys:rexxc" to the search path or move RX to a directory like "C:" that is already in the path.

(AmigaGuide has a built-in "RX" link command. It is not used here because scripts launched with the command exhibit some inconsistent behavior.)

2. The #?.rexx files distributed with the ARexxGuide archive must be stored either in the REXX: directory or in ARexxGuide's current directory.

The most versatile place to store any .rexx file is in the REXX: directory since it can then be found and launched by RX no matter what the current directory. That directory can become crowded, however. Since the interpreter looks for files first in the current directory, it can be a useful alternative for task-specific ARexx scripts like those included with this guide.

If the guide is launched from a shell or directory utility, the "CD" command should be used before launching the guide to change the working directory to the location of the .rexx files. If the guide is launched with an icon, the .rexx files should be stored in the same directory as the icon's .info file.

3. AmigaGuide should be launched as a command rather than through a call to the ShowNode() function of amigaguide.library.

Scripts that use the library function to launch AmigaGuide files have circulated on the nets. Using the function limits AmigaGuide's ability to call ARexx scripts.