

ARx_Index

COLLABORATORS

	<i>TITLE :</i> ARx_Index		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		January 8, 2025	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	ARx_Index	1
1.1	ARexxGuide Index	1
1.2	ARexxGuide Index	1

Chapter 1

ARx_Index

1.1 ARexxGuide | Index

AN AMIGAGUIDE® TO ARexx
by Robin Evans

Edition: 1.0c

```

Operator characters / Punctuation
A           B           C           D
E           F           G           H
I           J           K           L
M           N           O           P
Q           R           S           T
U           V           W           X Y Z

```

Copyright © 1993, Robin Evans. All rights reserved.

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

1.2 ARexxGuide | Index

Non-alphabetic

```

~~~~~
% , arithmetic operator
& , logical operator
&& , logical operator
() Grouping / Function argument list
() , Used in PARSE template
* , arithmetic operator
** , arithmetic operator
+ , arithmetic operator
+ , Used in PARSE template
+ , Used in PARSE template
, Comma
- , arithmetic operator
- , Used in PARSE template

```

- , Used in PARSE template
 / , arithmetic operator
 // , arithmetic operator
 ; Clause end symbol
 < , comparative operator
 <= , comparative operator
 <> , Used in templates
 = , Used on TRACE console
 = , comparative operator
 = , Assignment token
 = , Used in PARSE template
 == , comparative operator
 > , comparative operator
 >+> , TRACE console output
 >.> , TRACE console output
 >= , comparative operator
 >>> , TRACE console output
 >>> , Used in templates
 >C> , TRACE console output
 >F> , TRACE console output
 >L> , TRACE console output
 >O> , TRACE console output
 >P> , TRACE console output
 >U> , TRACE console output
 >V> , TRACE console output
 [] , Used in templates
 ^ , logical operator
 {} , Used in templates
 | , logical operator
 | , Used in templates
 || , concatenation operator
 ~ , logical operator
 ~< , comparative operator
 ~= , comparative operator
 ~== , comparative operator
 ~> , comparative operator

A

~~~~~~  
 ABBREV() , built-in function  
 ABS() , built-in function  
 Abuttal , concatenation operator  
 Accidental command from expression , Avoiding  
 Acknowledgements  
 ADDLIB() , built-in function  
 ADDRESS instruction keyword  
     Determining original address of script  
     Host  
 ADDRESS() , built-in function  
 Address strings  
 ALLOCMEM() , support function  
 Alphanumeric , DATATYPE() option  
 Amiga system functions  
 Amiga-specific extensions  
 AmigaDOS , Reading the output of a command  
 Anchor position , in PARSE template  
 Angle brackets , Used in templates  
 APPEND

---

Arbitrary  
 ARExx control functions  
 AREXX, Host name  
 ARG instruction keyword  
 ARG instruction subkeyword, PARSE  
 ARG(), built-in function  
 Argument to function  
     Determining if an argument exists  
 Arithmetic operator  
 Arithmetic operator, table  
 Arrays  
 Assigns, listing  
 Assigning command output to a stem variable  
 Assignment clause  
 Astonishment factor  
 Author

## B

~~~~~  
 B, Binary string identifier
 Reserved token
 B2C(), built-in function
 Backing up in the PARSE source string
 BADDR(), support function
 Bar, Used in templates
 Basic Elements
 Binary string
 Binary, DATATYPE() option
 Bit-wise functions
 BITAND(), built-in function
 BITCHG(), built-in function
 BITCLR(), built-in function
 BITCOMP(), built-in function
 BITOR(), built-in function
 BITSET(), built-in function
 BITTST(), built-in function
 BITXOR(), built-in function
 Blank lines
 Boolean, defined
 Branch name, Use of constant for name
 Branch variable
 BREAK instruction keyword
 Breaking structure
 BREAK_C instruction subkeyword, SELECT
 BREAK_D instruction subkeyword, SELECT
 BREAK_E instruction subkeyword, SELECT
 BREAK_F instruction subkeyword, SELECT
 Built-in function
 Busy-loop
 BY instruction subkeyword, DO
 Byte boundaries

C

~~~~~  
 C2B(), built-in function  
 C2D(), built-in function  
 C2X(), built-in function  
 CACHE, Instruction subkeyword, NUMERIC  
 CALL instruction keyword

---

---

- Case-sensitivity, in PARSE template
- CENTER(), built-in function
- Char/Num translation functions
- Character position, in PARSE template
- Clause
  - Including several clauses on one line
- Clip list
  - Removing entries
- CLOSE(), built-in function
- CLOSEPORT(), support function
- CMS
- Colon
- Color strings
- Combining different types of PARSE markers
- Comma
- Command clause
  - Assigning output to a stem variable
  - Avoiding accidental command from expression
  - Determining command used to launch a program
  - Entering in a script
- Command host
- Command inhibition
- COMMAND instruction subkeyword, ADDRESS
- Command returned 20
- Comment token
- COMPARE(), built-in function
- Comparison functions
- Comparison operator
  - Table
- Compound Symbol
- Compound variable
  - Case sensitivity
  - Default value
  - Derived name
  - Finding values
  - Using
    - Using non-printable characters
- COMPRESS(), built-in function
  - Count specific characters
- Con:, defined
- Concatenation operator
  - Used for formatted table
- Conditional expression
  - In DO instruction
  - Using functions
- Console device
- Console window, defined
- Constant
  - Used as file-handle name
- Continuation character
- Control loop, defined
- Control structure, defined
- COPIES(), built-in function
- Copy data from the program code
- Count specific characters in a string
- Counter
- Cowlishaw, M.F

---

Curly braces, Used in templates

## D

~~~~~  
D2C(), built-in function
D2X(), built-in function
Data-stream I/O
Database, reading records of set length
DATATYPE(), built-in function
 arguments
DATE(), built-in function
DATE() arguments
Debug, defined
Default host
Default precision
Default value of a variable
Default value of compound variable
DELAY(), support function
DELETE(), support function
DELSTR(), built-in function
DELWORD(), built-in function
Derived name of compound variable
Devices, listing
DIGITS(), built-in function
DIGITS(), Using numbers larger than the current setting
DIGITS, Instruction subkeyword, NUMERIC
Directory, changing CD
Directory, listing
DO instruction keyword
 Empty loop
DROP instruction keyword
Dummy instruction
Dyadic, defined

E

~~~~~  
ECHO instruction keyword  
Elapsed time counter  
ELSE instruction secondary keyword  
END instruction secondary keyword  
End position, in PARSE template  
EOF(), built-in function  
ERROR instruction subkeyword  
Error 8: Unrecognized token  
Error 8: Unrecognized token  
Error 13: Host environment not found  
Error 14: Requested library not found  
Error 17: Wrong number of arguments  
Error 18: Invalid argument to function  
Error 26: Missing or unexpected END  
Error 31: Symbol expected  
Error 41: Invalid expression  
Error 47: Arithmetic conversion error  
Error 47: Arithmetic conversion error  
Error 47: Arithmetic conversion error  
Error: Command returned 5/1  
ERRORTXT(), built-in function  
EXECIO  
EXISTS(), support function

---

EXISTS, function keyword, ARG()  
 EXIT instruction keyword  
 Exponential notation  
 EXPORT(), support function  
 EXPOSE instruction subkeyword  
 Expression  
 Expression, Avoiding accidental command from expression  
 EXTERNAL instruction subkeyword  
 External function

## F

~~~~~  
 FAILAT, Instruction subkeyword, NUMERIC
 FAILURE instruction subkeyword
 False
 Field divider
 File extension, Determining default ARexx extension
 File handle, Use of constant for name
 File input/output functions

- Pointer
 - Access protection
 - Handling error conditions
 - Information about
 - Moving to a new location in file
 - Overview of I/O functions
 - Parsing file name from full path
 - Redirection
 - Setting the logical file name
 - Standard I/O files
 - Storing arexx files
 - Using I/O functions with other devices

 File management functions
 Filename extension
 FIND(), built-in function
 Fixed symbol
 FOR instruction subkeyword
 FORBID(), support function
 FOREVER instruction subkeyword
 FORM(), built-in function
 FORM, Instruction subkeyword, NUMERIC
 Format, space
 Formatting output with RIGHT(), LEFT(), and TRUNC()
 Formatting strings
 FREEMEM(), support function
 FREESPACE(), support function
 Function argument
 Function call
 Function libraries, listing
 Function, calling a function named by a variable
 Functions index
 FUZZ(), built-in function
 FUZZ, Instruction subkeyword, NUMERIC

G

~~~~~  
 GETARG(), support function  
 GETCLIP(), built-in function  
 GETPKT(), support function  
 GETSPACE(), support function

---

Getting started  
Global tracing console  
Global variables, clip list  
Global variables, exposed in subroutines  
Glossary  
Goldberg, Gabriel

## H

~~~~~  
HALT instruction subkeyword
Handlers, listing
HASH(), built-in function
Hawes, William S.
Hello world, program
Hex and binary string
Hex, DATATYPE() option
Hexadecimal
HI
Host
 Current and previous
 Default
 Initial
Host/Library function

I

~~~~~  
I/O, defined  
I/O, File input/output functions  
IF instruction keyword  
Implied semicolon  
Implied semicolons  
IMPORT(), support function  
In-line data  
In-Line script  
In-line script, stored on clip list  
Index variable  
Index variable, Using variable name with END  
INDEX(), built-in function  
Informational functions  
Inhibiting commands  
Initial host  
INSERT(), built-in function  
Instruction  
Instruction keywords  
Interactive trace, starting from shell  
Interactive tracing  
Internal function  
Internal subroutine  
INTERPRET instruction keyword  
    Similarity to VALUE()  
    Using BREAK  
Interpreter, defined  
Interrupts, listing  
Interval, timing  
IOERR instruction subkeyword  
Iterate, defined  
ITERATE instruction keyword

## J

~~~~~

Julian, DATE() option
Jump

K

~~~~~  
Keyword, defined  
Keyword conflict

## L

~~~~~  
Label clause
Labels, stacked
LASTPOS(), built-in function
LASTPOS() Parsing file name from full path
LEAVE instruction keyword
LEFT(), built-in function
LEFT(), Formatting output
LENGTH(), built-in function
Library List, adding library to
Library List, listing entries
Library List, removing an entry
Library/Host function
Line-feed character
LINES(), built-in function
Link-surfing
List a directory -- SHOWDIR()
Logical device, defined
Logical operator
 Table
 Used in conditionals
Loop, defined
Lowercase
Lowercase, DATATYPE() option

M

~~~~~  
Macro language  
MAKEDIR(), support function  
Mantra, defined  
Marker, Combining different types in PARSE  
Matched pattern, in PARSE template  
Matching strings  
Math coprocessor, Determining system's coprocessor  
MAX(), built-in function  
MAX() Parsing file name from full path  
Memory addresses  
Memory management functions  
Memory-types, listing  
Message port  
Message port functions  
MIN(), built-in function  
Minutes, TIME() option  
Mixed, DATATYPE() option  
Month, DATE() option  
Multiple argument strings  
Multiple templates

## N

~~~~~  
Natural data typing
Navigating hints

Nested, defined
 NEXT(), built-in function
 NIL:, defined
 NOP instruction keyword
 NOVALUE instruction subkeyword
 Null clause
 NULL(), support function
 Number
 Number manipulation functions
 Number, treated as string
 NUMERIC instruction keyword
 NUMERIC instruction subkeyword, PARSE
 Numeric precision
 Setting
 Numeric, DATATYPE() option

O

~~~~~  
 OFF instruction subkeyword  
 OFFSET(), support function  
 OMITTED, function keyword ARG()  
 ON instruction subkeyword  
 OPEN(), built-in function  
 OPENPORT(), support function  
 OPENPORT(), support function  
 Operation  
 Operator precedence  
 Operator token  
 OPTIONS instruction keyword  
 OTHERWISE instruction secondary keyword  
 OTHERWISE, Implied semicolon  
 OVERLAY(), built-in function

P

~~~~~  
 Parentheses
 PARSE instruction keyword
 Path specification, parsing from full file name
 Path, Determining full path of executing program
 Pattern marker
 Pause for port to open
 PERMIT(), support function
 Placeholder token
 Pointer, moving file pointer
 Port, Waiting for a port to open
 Ports, example
 Ports, listing
 POS(), built-in function
 Positional marker
 PRAGMA(), built-in function
 Precedence of operation
 Precision, Using numbers larger than current setting
 Preferences, defined
 Prefix, defined
 Previous host
 Printer, sending output to
 Priority, changing
 PROCEDURE instruction keyword
 Processor, Determining system's processor

Program not found
PROMPT, Instruction subkeyword, NUMERIC
Prototyping, defined
Prototyping
PRT:, defined
Public message ports, listing
PULL instruction keyword
PULL instruction subkeyword, PARSE
PUSH instruction keyword

Q

~~~~~  
QUEUE instruction keyword  
Quotation marks  
    Using marks within a string  
    Used in commands

## R

~~~~~  
RANDOM(), built-in function
RANDU(), built-in function
RC
RC
READ
READCH(), built-in function
READLN(), built-in function
Redirection
References
REMLIB(), built-in function
RENAME(), support function
Repetitor in DO instruction
 TO/BY options
REPLY(), support function
Reserved, defined
 Keywords
 Characters
 Operator tokens
 X and B
Resources, listing
Restriction
RESULT
 Determining if OPTIONS RESULTS is set
 Returned from command
RESULTS Instruction subkeyword, OPTIONS
RETURN instruction keyword
RETURN, Using EXIT in external function
REVERSE(), built-in function
REXX: directory
REXX host, Commands understood
Rexxc
RexxMast
RIGHT(), built-in function
RIGHT(), Formatting output
Running a script
RX
RXC
RXLIB
RXSET

S

~~~~~

- SAY instruction keyword
- Scan, in PARSE template
- Scope of variables
- Search path
- Search priority, in the Library List
- Seconds, TIME() option
- Seed to RANDOM()
- SEEK(), built-in function
- SELECT instruction keyword
- Semaphores, listing
- Semicolon symbol
- SetBookmark
- SETCLIP(), built-in function
- SHOW(), built-in function
- SHOWDIR(), support function
- SHOWLIST(), support function
- SHOWLIST() arguments
- SIGL
  - Used to locate line number of a subroutine
- SIGN(), built-in function
- SIGNAL instruction keyword
- Smith, Phillip H. III
- SOURCE instruction subkeyword (find script's name)
- SOURCELINE(), built-in function
  - Retrieving in-line data
- SPACE(), built-in function
- Space, concatenation operator
- Spaces, removing from string
- Spaces, adding to string
- Special variable
- Square brackets, Used in templates
- Stack, changing
- Stacked labels
- STATEF(), support function
- STDERR
  - Defined
  - Pulling input from
- STDIN
  - Defined
  - Pulling input from
  - Stacking output to
- STDOUT
  - Defined
- Stem Symbol
- Stem variable
  - Assignment
  - STEM.0, Using 0 branch to hold count of branches
- Step value in DO instruction
- STORAGE(), support function
- String
  - Used as compound variable branch name
  - Used as file-handle name
- String manipulation functions
- String token
- STRIP(), built-in function
- Structured programming

---

- Subroutine
  - Defined
    - Local setting of interval counter
- SUBSTR(), built-in function
- SUBWORD(), built-in function
- Sullivan, Nick
- Symbol token
  - Compound
  - Fixed
  - Stem
  - Translated to uppercase
  - Variable
- SYMBOL, DATATYPE() option
- SYMBOL(), built-in function
- SYNTAX instruction subkeyword
- System libraries, listing
- System requesters, controlling

## T

~~~~~

- Table, formatting
- Tail, of compound variable
- Target
- Task ID, Determining
- Task-ready, listing
- TCC
- TCO
- TE
- Template
 - Multiple
- THEN, Implied semicolon
- TIME(), built-in function
 - Arguments
 - Elapsed time counter
- TO instruction subkeyword, DO
- Toggle host address
- Token
- Token delimiter
- Tokenization
- TRACE instruction keyword
- Trace options
- TRACE(), built-in function
- Transfer of control
- TRANSLATE(), built-in function
- TRIM(), built-in function
- True
- TRUNC(), built-in function
 - Formatting output
- TS
- TurboText, assigning macros to keys
- TurboText, example macro
- Tutorial
- TYPEPKT(), support function

U

~~~~~

- Uncrunch.rexx
- Uninitialized
- UNTIL instruction subkeyword, DO

---

UPPER instruction keyword  
 UPPER(), built-in function  
 UPPER, DATATYPE() option  
 UPPER, Instruction subkeyword, PARSE  
 UPPERCASE, Used in templates  
 User settings  
 Using VALUE()  
 Utility programs

## V

~~~~~  
 VALUE(), built-in function
 VALUE, instruction subkeyword, ADDRESS
 VALUE, instruction subkeyword, PARSE
 VAR instruction subkeyword, DO
 Variable symbol
 Variable
 Compound
 Controlling scope
 Data type
 Default value
 In PARSE template
 Scope
 Used as file-handle name
 Using
 VERIFY(), built-in function
 Checking unique datatypes
 VERSION instruction subkeyword, PARSE
 Video frequency, Determining system's frequency
 Video std, Determining system's video standard
 VM
 Volumes, listing

W

~~~~~  
 WaitForPort  
 Waiting tasks, listing  
 WAITPKT(), support function  
 Weekday, DATE() option  
 Weinstein, Marvin  
 WHEN instruction secondary keyword  
   Implied semicolon  
 WHILE instruction subkeyword, DO  
 Whole, DATATYPE() option  
 Window, sending output to  
 WITH instruction subkeyword, PARSE VALUE  
 Word manipulation functions  
 WORD(), built-in function  
 Word-counting  
 WORDINDEX(), built-in function  
 WORDLENGTH(), built-in function  
 WORDS(), built-in function  
 WRITE  
 WRITECH(), built-in function  
 WRITELN(), built-in function  
 Writing programs  
 WShell  
   Entering string programs

X Y Z

---

~~~~~

X, Hexadecimal string identifier

Reserved token

X2C(), built-in function

X2D(), built-in function

XRANGE(), built-in function

Zamara, Chris