ARx_Index

COLLABORATORS

	TITLE : ARx_Index		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		August 3, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 ARx_Index 1

1.1 ARexxGuide | Index 1

1.2 ARexxGuide | Index 1

1.3 ARexxGuide | Index 2

Chapter 1

ARx_Index

1.1 ARexxGuide | Index

```
This file is designed to be viewed from the main ARexxGuide
node.

Press either <Browse> key to see the index contents.
```

1.2 ARexxGuide | Index

```
                 AN AMIGAGUIDE® TO ARexx                 Second edition (v2.0)
   by Robin Evans
```

Non-alphabetic characters (opeators and special characters)

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Copyright © 1993,1994 Robin Evans. All rights reserved.

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

1.3 ARexxGuide | Index

```
Non-alphabetic
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    %  , arithmetic operator
    &  , logical operator
    && , logical operator
    () , Grouping/Argument list
    () , Used in PARSE template
    *  , arithmetic operator
    ** , arithmetic operator
    +  , arithmetic operator
    +  , Used in PARSE template
    ,    Comma
    -  , arithmetic operator
    -  , 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
    About this guide
    ABS() Built-in function
    Absolute marker
    Abuttal, concatenation operator
    Accidental commands
    Acknowledgements
    AddComma(): add commas to integer
    ADDLIB() Built-in function
    ADDRESS: Instruction keyword
       Command host
       Determine original address
    ADDRESS and the macro
    ADDRESS COMMAND
    Address string, defined
       Used in functions
    Address, Find location in memory
    ADDRESS() Built-in function
    ALLOCMEM() Support function
    Alphanumeric, DATATYPE() argument
    Amiga system functions
    AmigaDOS command programs
    AmigaDOS, Reading output of command
    Anchor position, in PARSE template
    ANSI REXX
    APPEND
```

Arbitrary precision of number
Arc: Program example, Uncrunch.rexx
ARexx Applications List
ARexx control functions
ARexx, host name
ARexx search path
ARexxGuide Online Help System
ARG() Built-in function
ARG, PARSE option
ARG: Instruction keyword
Argument
Argument string, commas
Argument, exists
Arithmetic operator: Table
Arithmetic operators
Array
Assign, SHOWLIST() option
Assigned variable
Assigning
Assignment clause
Astonishment factor
Author

B
~~~~~~
B, identifying a binary string
B2C() Built-in function
Back up in PARSE source string
BADDR() Support function
Basic Elements
Binary string
Binary, DATATYPE() argument
Bit-wise operation 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
BoolReq: Help system function
Branch name
Branch name, Use of constant
Branch variable
BREAK: Instruction keyword
Breaking structure
BREAK_C
BREAK_D
BREAK_E
BREAK_F
Building a new help-key macro
Built-in function
Busy-loop
BY, DO subkeyword
~~~~~~

```
    Byte boundaries

C
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    C2B() Built-in function
    C2D() Built-in function
    C2X() Built-in function
    CACHE, NUMERIC option
    CALL: Instruction keyword
    Case-sensitivity in PARSE template
    CD, changed directory with PRAGMA()
    CENTER() Built-in function
    Char/Num translation functions
    Character position
    Character translat
    Check unique datatypes
    Clause
       Assignment
       Commands
       Instructions
       Label
       Null
       Including several on one line
    Clause information, Help system
    Clip-list
       Removing entries
    CLOSE() Built-in function
    CLOSEPORT() Support function
    CMS
    Colon: Label identifier
    Color strings
    Combining markers in PARSE template
    Comma
       Continuation character
       Used in an argument string
    Comma: AddComma(): add to integer
    COMMAND, ADDRESS option
    Command clause
       Assigning output to stem
       Default host
       Determine initial host
       Host: what is it?
       Entering in a script
       Example script
       Retrieve result
    Command error
    Command failure
    Command inhibition, in TRACE
    Command programs
    Command returned 20
    Command returned 5/1
    Comment tokens
       Nested comment
    Comparative
    COMPARE() Built-in function
    Comparison functions
    Comparison operator
```

```
      Table
   Compatibility issues
   Complete program example
   Compound symbol
   Compound variable
      Case sensitivity
      Default value
      Finding values
      Overview
      Stem
      Tail
      Strings in tail
   COMPRESS() Built-in function
   CON: Open window, example
   CON: I/O functions
   CON:, defined
   Concatenate, defined
   Concatenation operator
      Used for formatted table
   CONDITION()
   Conditional expression
      In DO instruction
      Using functions
   Console device, defined
   Constant symbol
      Used as file-handle name
   Continuation token
   Control loop
   Control-structure, defined
   COPIES() Built-in function
   Copy data from the program code
   Count of stem variables
   CountChar(): count characters
   Counter
   CountWords(): count words in file
   Cowlishaw, M.F.
   Current host

D
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   D2C() Built-in function
   D2X() Built-in function
   Data file: pull single record from
   Data scratchpad
   Data storage/retrieval, examples
   Database, read records
   Datatype
   DATATYPE() Built-in function
      Arguments
   DATE() Built-in function
      Arguments
      Persistence of settings
   Debug, defined
   Debugging a macro
   Default address
   Default address, PARSE result
   Default host
```

```
    Default precision
    Default value, compound variable
    Default value, variable
    DELAY() Support function
    DELETE() Support function
    DELSTR() Built-in function
    DELWORD() Built-in function
    Derived name of compound variable
    Devices, SHOWLIST() argument
    DIGITS, NUMERIC option
    DIGITS() Built-in function
    DIGITS: Numbers larger than
    Directory, SHOWLIST() argument
    Directory, PRAGMA() option
    DisplayAG: Help System function
    DisplayStatus: Help System function
    DO: Instruction keyword
       Empty loop
    DROP: Instruction keyword
    Dummy instruction
    Dyadic, defined

E
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    ECHO: Instruction keyword
    EditorExit: Help System function
    Egregious, defined
    Elapsed-timer
    ELSE: Instruction keyword
    END: Instruction keyword
    Environmental variable
    Environmental variable: Get and set
    EOF() Built-in function
    Error code
    Error message
    Error message: redirect to a file
    ERRORTEXT() Built-in function
    ExecIO
    EXISTS, ARG() argument
    EXISTS() Support function
    EXIT: Instruction keyword
    Exponential notatation, defined
    Exponential notation
    EXPORT() Support function
    EXPOSE: Instruction keyword
    Expression
       Accidental commands
       Function
       Number
       Operation
       String
       Variable
    Extending stem variable
    EXTERNAL, PARSE option
    External function

F
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    FAILAT, OPTIONS option
    FAILURE, SIGNAL condition
    Field divider
    File extension
    File handle
    File input/output functions
       Access protection
       File pointer
       Error conditions
       Information about
       Logical file name
       Moving to new location
       Other devices
       Overview of I/O functions
       Redirection
       Standard I/O files
       Storing ARexx files
    File management functions
    Filename: split name from path
    Filename extension
    File size, determining
    FIND() Built-in function
    Finding values, compound variable
    Fixed symbol
    FOR, DO subkeyword
    FORBID() Support function
    FOREVER, DO subkeyword
    Foreword
    FORM, NUMERIC option
    FORM() Built-in function
    Format data into table form
    Format(): round and format number
    Formatting strings
    FREEMEM() Support function
    FREESPACE() Support function
    Function
       Argument
       Host
       Built-in
       External
       Internal
       Library/Host
          Listing
       Named by a variable
    Function reference
       ARexx control
       Bit-wise
       Char/Num
       Comparison
       File I/O
          File management
          Informational
          Memory control
          Message port
          Number changes
          String changes
```

```
      Word changes
FUZZ, NUMERIC option
FUZZ() Built-in function

G
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Get and send message packet
    Get and set environmental variable
    GETARG() Support function
    GetCLine: Help System function
    GETCLIP() Built-in function
    GetCPos: Help System function
    GetCWord: Help System function
    GETPKT() Support function
    GETSPACE() Support function
    Getting ARexx started
    GetWinInfo: Help System function
    Global tracing console
    Global variables
    Global variable, in subroutine
    Glossary of terms
    Goldberg, Gabriel
    Growing a macro
    GUI, defined

H
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    HALT, SIGNAL condition
    Handlers, SHOWLIST() argument
    HASH() Built-in function
    Hawes, William S.
    Hello World!
    Help system
       Setup
    Hex string
    Hex, DATATYPE argument
    Hexadecimal, changing to character
    HI: Command utility
    Host, current and previous

I
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    I/O, defined
       Functions
       Other devices
       Overview of I/O functions
       Setting the logical file name
       Standard I/O files
       Technique examples
    IF: Instruction keyword
       A closer look
    Implied semicolons
    IMPORT() Support function
    In-line script
       Stored on clip-list
    Index variable
    Index variable, with END
```

```
    INDEX() Built-in function
    Informational functions
    Input/Output, defined
       Technique examples
    INSERT() Built-in function
    Instruction
    Instruction reference
    Integer: AddComma(): add commas to
       Requirements
    Interactive examples, list
    Interactive trace, TRACE option
       TS command
    Internal function
    INTERPRET: Instruction keyword
       Similarity to VALUE()
       Using BREAK
    Interpreted variable name
    Interpreter, defined
    Interprocess Communication
    Interrupts, SHOWLIST() argument
    Interval timer
    Introduction to ARexx
    IO, defined
    IOERR, SIGNAL condition
    Iterate, defined
    ITERATE: Instruction keyword

J
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Julian, TIME() argument
    Jump

K
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Keyword conflict
    Keyword, defined

L
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Label clause
       Identified by colon
       Stacked
    LASTPOS() Built-in function
    Launch a new script
    LEAVE: Instruction keyword
    LEFT() Built-in function
    LENGTH() Built-in function
    Lha: Program example, Uncrunch.rexx
    Library function
       Adding to list
       Removing from list
    Library: Determining version number
    Line-feed character
    LINES() Built-in function
    Literal string
    Logical device, defined
    Logical expression
```

```
    Logical file name
    Logical operator
       Table
    Loop, defined
    Lowercase, converting to
    Lowercase, DATATYPE() option

M
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Macro language
    Macro: Adding ARexx control
    MAKEDIR() Support function
    Mantra, defined
    Matched pattern
    Matching strings
    Math coprocessor
    MAX() Built-in function
    MED/MOD
    Memory addresses
    Memory management functions
    Memory-types, SHOWLIST() argument
    Message port functions
       Using in script
    MIN() Built-in function
    Mixed, DATATYPE() argument
    Month, DATE() argument
    Multiple argument strings
    Multiple template

N
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Natural typing
    Navigating hints
    Nested comment
    Nested, defined
    NEXT() Support function
    NIL:, defined
    Node-Surfing
    NOP: Instruction keyword
    NOVALUE
    Null clause
    NULL() Support function
    Number
       Possible loss of precision
       Precision
       Precision, Setting
       Treated as string
    Number manipulation functions
    Numeric
    NUMERIC: Instruction keyword
    NUMERIC: PARSE option

O
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    OFFSET() Support function
    OMITTED
    ON | OFF, SIGNAL options
```

```
    Open console windows for I/O
    OPEN() Built-in function
    OPENPORT() Support function
    Operation
    Operator
       priority
       Table of arithmetic
       Table of comparison
       Table of logical
    OPTIONS: Instruction keyword
    OTHERWISE: Instruction keyword
       Implied semicolon
    Output text to a printer
    OVERLAY() Built-in function
    Overview: Compound variables
    Overview: I/O functions

P
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Parentheses
    Parentheses: Change priority
    PARSE: Instruction keyword
       Template
          Back up in source string
          End position
          Pattern
          Pattern marker
          Period: Placeholder token
    ParseFileName(): split name/path
    Pattern
    Pattern marker
    Pause for port to open
    Period: Placeholder token
    PERMIT() Support function
    Pipe: Get command output with pipe
    Placeholder token
    Pointer
       Moving
    Port, SHOWLIST() argument
    Port, Waiting for a port to open
    POS() Built-in function
    Positional marker
    PRAGMA() Built-in function
    Precision of number
    Preface
    Preferences, defined
    Prefix, defined
    Previous host
    Printing, example
    Print: I/O functions
    Printer, sending output to
    Priority
    Priority of operator
    Priority, parentheses
    PROCEDURE: Instruction keyword
    Processor, Determining system's
    Program, start script from another
```

    Program not found
    Prompt, OPTIONS option
    Prototyping, defined
    PRT:, defined
       I/O functions
    Public message ports
    PULL, PARSE option
    PULL: Instruction keyword
    PUSH, Data-stream I/O
    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: Special variable
    READ, OPEN() argument
    READCH() Built-in function
    READLN() Built-in function
    Redirection
    Registration form
    Relative marker
    REMLIB() Built-in function
    RENAME() Support function
    Repeating macro with an ARexx loop
    Repetitor in DO instruction
       TO/BY options
    REPLY() Support function
    ReqTools.library
    Reserved, defined
       Keywords
       Characters
       Operator tokens
       X and B
    Resolved name of the program
    Resources, SHOWLIST() argument
    RESULT: Special variable
    RESULTS, OPTIONS option
    Retrieve data from source code
    Retrieve result of AmigaDOS command
    Return code
    Return control
    RETURN: Instruction keyword
    Returned
    REVERSE() Built-in function
    REXX host, Commands understood
    REXX standard
    REXX: directory
    rexxarplib.library
    RexxDosSupport.library

    RexxMast
    rexxreqtools.library
    rexxserdev.library
    rexxsupport.library
    RIGHT() Built-in function
    Round and format a number
    Running a script
    RX: Command utility
    RXC: Command utility
    RXLIB: Command utility
    RXSET: Command utility
    rx_intui.library

S
~~~~~~
    SAY: Instruction keyword
    Scan, in PARSE template
    Search path
    Search priority, in Library List
    Seed
    SEEK() Built-in function
    SeekToRecord(): technique example
    SELECT: Instruction keyword
    Semaphores, SHOWLIST() argument
    Semicolon: Clause end symbol
    Serial port
    SetAddress: Help System function
    SETCLIP() Built-in function
    SetExecStr: Help System function
    Shareware
    SHOW() Built-in function
    SHOWDIR() Support function
    SHOWLIST() Options
    SHOWLIST() Support function
    SIGL: Special variable
       Used to locate a subroutine
    SIGN() Built-in function
    SIGNAL: Instruction keyword
       Trapping error conditions
       Unconditional jump (Goto)
    Simple macro: recording keystrokes
    Simple symbol
    Simple variable
    Smith, Philip H. III
    Source code: Retrieve data from
    SOURCE, PARSE option
    SOURCELINE() Built-in function
    Space operator
    SPACE() Built-in function
    Spaces, removing from string
    Special character
    Special variable
       RC
       RESULT
       SIGL
    SQLdb
    Stack, PRAGMA() argument
~~~~~~

```
    Stacked labels
    Standard I/O file
    STATEF() Support function
    STDERR, defined
       Standard I/O file
    STDIN, defined
       Stacking data to
       Standard I/O file
    STDOUT, defined
       Pulling input from
       Standard I/O file
    Stem symbol
       Assignment to
       Count of variables
    Step value
    STORAGE() Support function
    Storing ARexx Files
    String manipulation functions
    String
       Token
       Used as file-handle name
       Technique examples
    STRIP() Built-in function
    Structured programming
    Subroutine, defined
    Subroutine, local settings
    SUBSTR() Built-in function
    SUBWORD() Built-in function
    Sullivan, Nick
    Symbol token
       Compound
       Fixed
       Stem
       Translated to uppercase
       Variable
    SYMBOL() Built-in function
    Symbol, DATATYPE argument
    SYNTAX, SIGNAL condition
    System libraries
    System requester, disable

T
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Table: Format data into table form
    Target variable in PARSE template
    Task ID, PRAGMA() argument
    Task-ready, SHOWLIST() argument
    TCC: Command utility
    TCO: Command utility
    TE: Command utility
    Techniques:
    Template
       Variable
    THEN, Instruction subkeyword
       Implied semicolon
    TIME() Built-in function
       Arguments
```

```
      Elapsed time counter
      Persistence of settings
   TO, DO subkeyword
   Toggle addresses
   Token
      Arithmetic
      Colon label identifier
      Comment token
      Comparative
      Concatenation
      Delimiter
      Fixed symbol
      Hex and binary string
      Logical
      Operator token
      Parenthesis
      Reserved character
      Semicolon clause end symbol
      Strings
      Symbol token
      Comma
      Variable symbol
   Tokenization in PARSE template
   TRACE() Built-in function
   TRACE: Instruction keyword
      Options
   TRANSLATE() Built-in function
   TRIM() Built-in function
   TRL1
   TRL2
   TRUNC() Built-in function
   Truncate, defined
   TS: Command utility
   TurboText: Example script
      Assigning macros to keys
   Tutorials
   Typeless variable
   TYPEPKT() Support function

U
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   Uncrunch.rexx: Program example
   Uninitialized
   UNTIL, DO subkeyword
   UPPER() Built-in function
   Upper, DATATYPE() argument
   UPPER, PARSE option
   UPPER: Instruction keyword
   Useful tools
   User function
   User settings, storing on clip-list
   Using variable as template marker

V
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
   VALUE() Built-in function
   VALUE, ADDRESS subkeyword
```

```
    VALUE, PARSE subkeyword
    VAR, PARSE option
    Variable
       Compound
       Data type
       Default value
       In PARSE template
       Mandatory declaration
       Used as file-handle name
       Overview
       Scope
       Special variable
       Symbol
       Used as file-handle name
    Stored on the clip list
    VERIFY() Built-in function
    VERSION, PARSE option
    Version, determining library
    Video frequency, determining
    Video standard, determining
    VM
    Volumes, SHOWLIST() argument

W
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    WaitForPort: Command utility
    Waiting, SHOWLIST argument
    WAITPKT() Support function
    Weekday, DATE() argument
    Weinstein, Marvin
    WHEN: Instruction keyword
       Implied semicolon
    WHILE, DO subkeyword
    Whole, DATATYPE() argument
    Why ARexx?
    Window, sending output to
    Window, example
    WITH, PARSE subkeyword
    Word count, example function
    Word manipulation functions
    WORD() Built-in function
    WORDINDEX() Built-in function
    WORDLENGTH() Built-in function
    WORDS() Built-in function
    WordWrap(): technique example
    Write data to a new file
    WRITE, OPEN() argument
    WRITECH() Built-in function
    WRITELN() Built-in function
    Writing programs
    WShell

X
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    X, identifying hex string
    X2C() Built-in function
    X2D() Built-in function
```

```
    XRANGE() Built-in function

Z
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    Zamara, Chris
    zoo: Program example
```