

Index

SYMBOLS

&& (ampersand, double), logical EXCLUSIVE OR operator, 30

& (ampersand), logical AND operator, 30

angle brackets

- > (append to file), 76, 214
- > (greater than operator), 29
- >= (greater than or equal to operator), 29
- <> or >< (greater than or less than operator), 29
- < (less than operator), 29
- <= (less than or equal to operator), 29
- < (redirect input), 76, 214
- > (redirect output), 76, 214
- > (strictly greater than operator), 29
- >= (strictly greater than or equal to operator), 29
- << (strictly less than operator), 29
- <<= (strictly less than or equal to operator), 29
- >.> in trace file, 139
- >> in trace file, 138
- >C> in trace file, 139
- >F> in trace file, 139
- >L> in trace file, 139
- >O> in trace file, 139
- >P> in trace file, 139
- >V> in trace file, 138, 139

* (asterisk)

- multiplication operator, 27
- *.* in trace file, 138

** (asterisk, double), raise to a power operator, 27

backslash

- as ANSI-standard “not” sign, 204, 496
- \ or ¬ (logical NOT operator), 30
- \= or ¬= (not equal operator), 29

\> or ¬> (not greater than operator), 29

\< or ¬< (not less than operator), 29

\== or ¬== (strictly not equal operator), 29

\> or ¬> (strictly not greater than operator), 29

\<< or ¬<< (strictly not less than operator), 29

{ } (braces), vector class reference, 454

[] (brackets)

- adding or retrieving from collection, 470
- in array indexes, 397
- arraylike reference, 453

^ (caret), method invocation prefix, 453

^^ (caret, double), instance creation, 453

:: (colon, double), preceding directives, 465, 470

: (colon), following a label, 42

, (comma), line continuation character, 111, 122, 173, 186

/*...*/ (comment delimiters), 22

= (equals sign)

- assignment, 26
- equal comparison operator, 29

== (equals sign, double), strictly equal operator, 29

! (exclamation), command identifier, 454

- (minus sign)

- negative number prefix operator, 27
- subtraction operator, 27

() (parentheses)

- affecting order of precedence, 31–32
- enclosing function arguments, 23

% (percent), integer division operator, 27

. (period), placeholder variable, 116

+ (plus sign)

- addition operator, 27
- positive number prefix operator, 27
- +++ in trace file, 139

? (question mark), placeholder variable

? (question mark), placeholder variable, 247, 252

“...” (quotes, double)

enclosing character strings, 23, 26

enclosing OS commands, 181–182

‘...’ (quotes, single), enclosing character strings, 23, 26

; (semicolon), line separation character, 59, 173

/ (slash), division operator, 27

// (slash, double), remainder division operator, 27

~~ (tilde, double), method invocation, 466, 470

~ (tilde), method invocation, 453, 465, 466, 470

_ (underscore), in variable names, 171

|| (vertical bar, double), concatenation operator, 30–31, 80

| (vertical bar), logical OR operator, 30

A

abbrev function, 89, 90–91, 547

abs function, 103–104, 547–548

abstract classes, Open Object Rexx, 467

abstraction, root, 452

abuttal concatenation, 80

acos function

BRexx, 364

Rexx/imc, 349

active environment, 211

ActiveState Web site, 258

addition operator (+), 27

address function, 196, 204, 220, 548

address instruction

for commands to other environments, 221

definition of, 216–218, 535–536

example using, 218–219, 222–225, 536–537

mainframe Rexx, 504

using stack for command I/O, 225–226

Administration tool, 387, 399, 615

Aggregate class, root, 455

AIX, Rexx for, 321

Alarm class, Open Object Rexx, 469, 620

American National Standards Institute. See ANSI

Amiga Forum, 532

Amiga Rexx (ARexx), 323, 573

ampersand, double (&&), logical EXCLUSIVE OR operator, 30

ampersand (&), logical AND operator, 30

angle brackets

> (append to file), 76, 214

> (greater than operator), 29

>= (greater than or equal to operator), 29

<> or >< (greater than or less than operator), 29

< (less than operator), 29

<= (less than or equal to operator), 29

< (redirect input), 76, 214

> (redirect output), 76, 214

> (strictly greater than operator), 29

>= (strictly greater than or equal to operator), 29

<< (strictly less than operator), 29

<<= (strictly less than or equal to operator), 29

>.> in trace file, 139

>> in trace file, 138

>C> in trace file, 139

>F> in trace file, 139

>L> in trace file, 139

>O> in trace file, 139

>P> in trace file, 139

>V> in trace file, 138, 139

ANSI (American National Standards Institute)

ANSI screen I/O, BRexx library for, 366, 371–376

ANSI-1996 standard, 8, 193, 194–195, 309, 310

ANSI-standard “not” sign, 204, 496

answers to study questions, 637–655

ANY condition, 472

Apache Web server, programming, 281–288, 624–625

API, running Rexx as, 324

APILOAD function, VM Rexx, 498

APPCMVS interface, 505

applet, Java, 525–526

application interfaces, using Rexx for, 10

ARexx (Amiga Rexx), 323, 573

AREXX, functions supported by Regina, 336

arg function, 117, 195, 548–549

arg instruction, 81, 115–117, 537

arguments of functions, 23, 25

arithmetic operators, 27–28

AROS, 323

Array class, Open Object Rexx, 468, 619

array names (stem variables), 55

array references (compound symbols), 54–55

arraylike reference operator ([]), 453

arrays. See also do instruction

addressing elements of, 54

associative, 54, 60–62

data structures based on, creating, 63–64

declaration of, 53

definition of, 53–54

dense, 53

dimensionality of, 53, 63

example using, 58–60, 61–62

first element of, as 0 or 1, 57
 indexing, Reginald features for, 397
 initializing, 55, 56–57
 NetRexx, 522
 number of elements in, storing, 57
 processing all elements in, 56
 redirecting command I/O to and from, 216, 217
 referencing uninitialized element of, 55
 sorting, in Reginald, 396
 sparse, 53
Ashley, W. David (“Harnessing Apache for REXX Programming”), 288
asin function
 BRexx, 364
 Rexx/imc, 349
ask special name, NetRexx, 522, 524, 629
assignment statements, 25, 26
associative arrays, 54, 60–62
Associative Arrays for Rexx package, 615
associative memory, 54. See also arrays
asterisk (*)
 multiplication operator, 27
 . in trace file, 138
asterisk, double (), raise to a power operator, 27**
atan function
 BRexx, 364
 Rexx/imc, 349
attributes (variables), Open Object Rexx, 465
a2u function, BRexx, 368
AuroraWare! tool, r4 and rool! interpreters, 450

B

backslash
 as ANSI-standard “not” sign, 204, 496
 \ or ¬ (logical NOT operator), 30
 \= or ¬= (not equal operator), 29
 \> or ¬> (not greater than operator), 29
 \< or ¬< (not less than operator), 29
 \== or ¬== (strictly not equal operator), 29
 \> or ¬> (strictly not greater than operator), 29
 \<< or ¬<< (strictly not less than operator), 29
Bag class, Open Object Rexx, 468, 619
balanced tree (B-tree), 63
Barron, David (*The World of Scripting Languages*), 11
base reference, rool!, 453
batch procedures, 210
b2c function, Regina, 336, 573–574
b2d function, Rexx/imc, 350
beaming, 441
Bean Scripting Framework, 615
beep function
 Regina, 338, 574
 Reginald, 399
bifurcation, 79–80
binary numbers, 26
binding variables, 247, 249–250
bit function, Reginald, 399
bit manipulation functions, Regina, 336–337
bit map indexes, 97
bit string functions, 96–97
bitand function, 97, 106–107, 549
bitchg function, Regina, 336, 574
bitclr function, Regina, 336, 574–575
bitcomp function, Regina, 336, 575
bitor function, 97, 549
bitset function, Regina, 337, 575
bittst function, Regina, 337, 575–576
BitVector class, rool!, 455
bitxor function, 97, 549–550
blank lines, 23
Bochs emulator, 424
books. See publications
box style comments, 22
braces {}, vector class reference, 454
brackets []
 adding or retrieving from collection, 470
 in array indexes, 397
 arraylike reference, 453
BRexx interpreter
 advantages of, 316, 318, 360
 command I/O, stack for, 368–369
 definition of, 13, 312, 359–360
 documentation for, 360, 361
 downloading, 360–361
 example using, ANSI screen I/O, 371–376
 example using, C-language I/O, 369–371
 example using, direct data access, 376–378
 example using, DOS functions, 378–382
 external function libraries, 366–367
 extra features in, 363–366
 installing, 361–363
 I/O functions, 364–366
 mathematical functions, 364
 platforms supported by, 359
 stack functions, 363
 system functions, 363
 Windows CE functions, 367–368

BSD platforms, 8

B-tree (balanced tree), 63

buffers, relationship to stacks, 167–168

buftype function

Regina, 342, 576

Reginald, 396, 398

built-in functions, 110, 120. See also functions

built-in subroutines, 41. See also subroutines

b2x function, 105, 550

byte-oriented I/O. See character-oriented I/O

C

C Developer's Kit for Reginald, 388

>C> in trace file, 139

CALL construct, 34

call import function, BRexx, 363

call instruction

definition of, 41–43, 112, 185, 537–538

error trapping with, 144–146, 152–154

example using, 70

as structured construct, 34

when to use, 71

call off instruction, 148

call on instruction, 148

Callable service library (CSL) routines, 505

Callback class, roo!, 455

callback function, roo!, 454

callbacks, Rexx/DW, 264

call-level interface (CLI), 230

camel case, 171

capitalization, using effectively, 170–171

caret, double (^), instance creation, 453

caret (^), method invocation prefix, 453

CASE construct, 34. See also select instruction

case sensitivity

automatic uppercase conversion, 186

capitalization, using effectively, 170–171

of Rexx language, 22–23

catch...finally instructions

NetRexx, 521

roo!, 454

c2b function, Regina, 337, 576

c2d function, 105, 553

cd function, Regina, 338, 576

cell phones, 423. See also handheld platforms

center function, 89, 550

centre function, 550

CFLOW tool, r4 and roo! interpreters, 451

CGI (Common Gateway Interface)

BRexx library for CGI-scripting functions, 367

definition of, 273

Internet/REXX HHNS WorkBench, 276–281, 616

Reginald support for, 388

Reginald tutorial for, 392

Rexx/CGI library (*cgi-lib.rxx*), 274–276, 615

cgidie function, CGI/Rexx, 274

CgiEnd function, HHNS WorkBench, 278

cgiererror function, CGI/Rexx, 274

CgiHref function, HHNS WorkBench, 278

CgiImg function, HHNS WorkBench, 278

CgiInit function, HHNS WorkBench, 277

cgi-lib.rxx library (CGI/Rexx), 274–276, 615

CgiRefr function, HHNS WorkBench, 278

CGI/Rexx, 274–276, 615

CGI-scripting functions, BRexx library for, 367

CgiSetup function, HHNS WorkBench, 277

CgiWebit function, HHNS WorkBench, 278

changestr function

definition of, 90, 550–551

mainframe Rexx, 504

Open Object Rexx, 472

roo!, 454

character encoding stream, 96

character folding, 97

character map, 96

character sets, affecting portability, 203

character strings, 23. See also literals

character-oriented I/O, 68, 72–75, 191

CharacterVector class, roo!, 455

charin function

definition of, 72, 73, 551

return string for, 180, 196

charout function

definition of, 73, 551

example using, 74

explicitly controlling file positions, 72

return string for, 180, 196

chars function

definition of, 73, 552

return string for, 180, 196–197, 205

chdir function

Regina, 338, 576

Reginald, 391

Rexx/imc, 348, 352

Chill tool, r4 and roo! interpreters, 451

chkpwd function, CGI/Rexx, 274

Christensen, Anders (developer of Regina), 312, 331

CICS (Customer Information Control System), Rexx for, 320

C-language interface, for RexxXML, 294

C-language I/O functions

BRexx, 364–365

Rexx/imc, 351, 354–356

Class class, Open Object Rexx, 469, 620

::class directive, Open Object Rexx, 470

class hierarchies, Open Object Rexx, 466–467

class instruction

NetRexx, 521, 630

rool, 453

classes

NetRexx, 520

Open Object Rexx, 465–467, 468–469, 476, 619–622

rool, 452, 455–456

cleanquery function, CGI/Rexx, 274

clear command, Linux or Unix, 45

CLI (call-level interface), 230

Clipboard class, rool, 455

clipboard function, BRexx, 368

close function

BRexx, 365

Regina, 339, 577

Rexx/imc, 351

clreol function, BRexx, 368

clrscr function, BRexx, 367

cls command, Windows, 45

CMD environment, Reginald, 393–394

CMS assembler macros and functions, 505

CMS commands, VM Rexx, 498, 499, 506–511

CMS functions, HHNS WorkBench, 278

CMSFLAG function, VM Rexx, 498

Code Comments community, 531

code pages (character sets), affecting portability, 203

code reviews, 183–184

collection classes, Open Object Rexx, 468, 481–485

Collier, Ian (developer of Rexx/imc), 312

colon, double (::), preceding directives, 465, 470

colon (:), following a label, 42

column-position files, 86

comma (,), line continuation character, 111, 122, 173, 186

comma-delimited files, 86

command identifier operator (!), 454

command line, passing parameters on, 115–116, 185

command procedure language, Rexx as, 309

command procedures

definition of, 209–210

example using, 222–225

stack for command I/O, 225–226

commands in other environments, issuing, 220–221

commands, OS

affecting portability, 191–192, 201–202

building string for, 212

capturing output from, 213–215, 216–218

directing input lines to, 214, 216–218

environment for, 211, 216–217, 220

environmental functions, Regina, 338

environmental functions, Rexx/imc, 348, 352–354

error trapping for, 213, 216–218

example using, 218–219, 222–225

issuing, 211–212

issuing, with Reginald, 393–394

issuing, with VM Rexx, 496

quotes enclosing, 181–182

result of (return code), 154, 211, 213

comment delimiters (/...*/), 22

comments

guidelines for, 176–177

in rool, 454

syntax for, 22, 175–176, 184

in VM Rexx, 494

commercial Rexx interpreters, 322–323

Common Gateway Interface. See CGI

Common Programming Interface (CPI), 505

Comparator class, rool, 455

compare function, 90, 552

comparison operators, 28–30, 187

compilers. See also mainframe Rexx

definition of, 323–324

VM Rexx, 498–499

compound comparisons, 30

compound symbols (array references), 54–55

compound variable name, 25

compress function, Regina, 337, 577

concatenation, 30–31, 79–80

concatenation operator (||), 30–31, 80

concurrency, Open Object Rexx, 468, 489–491

condition function, 154, 155–156, 197, 552–553

condition trapping. See error trapping

conditions. See error conditions

Console class, rool, 455

CONSOLE interface, 505

console I/O, BRexx library for, 367

constants, 54

constructs (control structures), 33, 34, 47

content-addressable (associative) arrays, 54, 60–62

ContextVector class, rool, 455

control structures (constructs), 33, 34, 47

conversational I/O, 75

conversions between numeric representations, 105–107

convertdata **function, Reginald, 399**

copies **function, 90, 553**

COPYFILE **command, CMS, 510**

copyfile **function**

BRexx, 368

Reginald, 391

cos **function**

BRexx, 364

Rexx/imc, 349

cosh **function, BRexx, 364**

countstr **function**

definition of, 90, 553

Open Object Rexx, 472

rool!, 454

Cowlishaw, Michael (Rexx inventor)

The Rexx Language: A Practical Approach to Programming (TRL-1), 193

The Rexx Language (TRL-2), 8, 193, 310, 532

role in Rexx language, 6, 308, 312

CP assembler macros, 505

CP commands, VM Rexx, 506–511

CP DIAGNOSE instructions, 505

CP system services, 505

CPI (Common Programming Interface), 505

CPI Resource Recovery Routines, 505

CPICOMM interface, 505

crypt **function, Regina, 337, 577**

CSL (Callable service library) routines, 505

CSL function, VM Rexx, 498

CUR for Rexx, 616

courses package, 111

cursor processing, 245–247, 252

Customer Information Control System (CICS), Rexx for, 320

c2x **function, 97, 105, 554**

D

data conversions, of variables, 24

Data Definition Language (DDL), 230

Data Manipulation Language (DML), 230

data structures based on arrays, 63–64

data type, of variables, 24, 25

database. See also Rexx/SQL package

binding variables, 247, 249–250

connections to, 233, 234–236, 248–250

cursor processing, 245–247, 252

information about, retrieving, 234–238

interfaces to, 203, 250–253

issuing SQL statements, 233, 250, 251

for Palm OS, 439

relational, 229

tables, creating and loading, 239–241

tables, selecting results from, 241–244, 245–247

tables, updating, 243–244

transactions, 233

database functions, Rexx for Palm OS, 440

Database Manager API, Palm OS, 440

datatype function, 88, 90–92, 105–106, 454, 554

date function, 196, 555

date functions

BRexx library for, 367

HHNS WorkBench, 278

d2b function, Rexx/imc, 350

DB2 UDB database, 248–249, 250–253

DB2 UDB SQL interfaces, 505

dbclose function, BRexx, 365

dbconnect function, BRexx, 365

DBCS (Double-Byte Character Set), 496–497

dberror function, BRexx, 365

dbescstr function, BRexx, 365

dbfield function, BRexx, 365

DBForums Web site, 531

dbget function, BRexx, 365

dbinfo function, BRexx, 365

dbisnull function, BRexx, 365

DBMS (database management system). See database

dbsql function, BRexx, 365

d2c function, 105, 556

DDL (Data Definition Language), 230

debugging. See also error conditions; error trapping

interactive, 4, 6, 140–142

methods of, 134

say instruction for, 133–135

trace function for, 139–140, 196, 197, 566–567

trace instruction for, 135–139, 140–142, 545–546

decimal (fixed-point) numbers, 26

dedicated devices, programming. See embedded programming

default environment, 211, 636

delay function, HHNS WorkBench, 277

DeleteFile function, Reginald, 391

delfile function, BRexx, 368

delfquery function, CGI/Rexx, 274

DELSTACK command

mainframe Rexx, 167

OS/TSO Rexx, 502

delfstack function, rool!, 454

- `delstr` **function**, 90, 555
 - `delword` **function**, 92, 556
 - dense arrays**, 53
 - `DESBUF` **command**
 - CMS, 499
 - mainframe Rexx, 166
 - `desbuf` **function**
 - BRexx, 363
 - Regina, 167, 342, 577–578
 - Reginald, 396, 398
 - `deweb` **function**, CGI/Rexx, 274
 - `DIAG` **function**, VM Rexx, 498
 - `DIAGRC` **function**, VM Rexx, 498
 - Dialog Editor**, Reginald, 390
 - `digits` **function**, 196, 556
 - `digits` **special name**, NetRexx, 522, 629
 - `dir` **function**
 - BRexx, 368
 - Reginald, 391
 - directives**, Open Object Rexx, 465–466, 470
 - `Directory` **class**, Open Object Rexx, 468, 619
 - `directory` **function**
 - Regina, 338, 578
 - Reginald, 391
 - division operator (/)**, 27
 - DLLs**, Reginald, 395–396
 - DML (Data Manipulation Language)**, 230
 - DO construct**, 34
 - `do forever` **instruction**, 47–49
 - `do` **instruction**
 - definition of, 37–38, 538
 - example using, 38–40, 43–46
 - NetRexx, 521, 630
 - as structured construct, 34
 - subscripts for, 181
 - `do over` **instruction**, Reginald, 396–397
 - `do until` **instruction**, 47–48
 - `do while` **instruction**, 37, 47–48
 - document trees**, 293, 294, 298
 - documentation**. *See also* publications
 - for BRexx interpreter, 360, 361
 - IBM Rexx manuals, 533
 - for NetRexx interpreter, 517
 - for r4 interpreter, 447, 449
 - for Regina interpreter, 14, 333
 - for Reginald interpreter, 386, 387, 392–393
 - for Rexx/imc interpreter, 346
 - for roo! interpreter, 447, 449
 - `do-end` **pair**, 35–36, 37–38
 - DOS emulation**, 424–425, 426–429
 - DOS functions**, BRexx library for, 367, 378–382
 - DOS platforms**
 - definition of, 8
 - return codes for OS commands, 154
 - double quotes (“...”)**
 - enclosing character strings, 23, 26
 - enclosing OS commands, 181–182
 - Double-Byte Character Set (DBCS)**, 496–497
 - DO-WHILE construct**, 34
 - Dr. Dialog interface**, 257
 - `DriveContext` **class**, roo!, 455
 - `DriveInfo` **function**, Reginald, 391, 401
 - `DriveMap` **function**, Reginald, 391, 401
 - driver**, 42
 - `drop` **instruction**, 115, 539
 - `DROPEBUF` **command**
 - CMS, 499
 - mainframe Rexx, 166
 - OS/TSO Rexx, 502
 - `dropbuf` **function**
 - BRexx, 363
 - Regina, 167, 342, 578
 - Reginald, 396, 399
 - roo!, 454
 - DW (Dynamic Windows)**, GUI package based on. *See* Rexx/DW package
 - DW package**, 111
 - `d2x` **function**, 105, 556
 - `DYLD_LIBRARY_PATH` **variable**, 340
 - Dynamic Windows (DW)**, GUI package based on. *See* Rexx/DW package
- ## E
- EBCDIC functions**, BRexx library for, 367
 - edit macros**, VM Rexx, 496
 - `EditName` **function**, Reginald, 391
 - editors**, Rexx-aware, 183
 - embeddable language**, using Rexx as, 10
 - embedded Linux**, 430
 - embedded programming**
 - definition of, 429–430
 - interpreters supporting, 321–322, 430
 - types of dedicated devices, 422
 - Embedded Systems Programming Magazine**, 430
 - `Emitter` **class**, roo!, 455
 - “An Empirical Comparison of Seven Programming Languages” (Prechelt), 11

emulation, for handhelds, 423–425, 426–429

encapsulation

- Open Object Rexx, 465
- rool!, 452

endless loop, 47

end-of-file (EOF) character, 73, 74

environment

- active, 211
- default, 211, 636
- determining, 195–199
- other than operating system, issuing commands to, 220
- specifying for OS commands, 216–217

.environment object, Open Object Rexx, 470

environmental functions

- Regina, 338
- Rexx/imc, 348, 352–354
- Rexx/SQL, 233

environmental variables, Open Object Rexx, 464

EOF (end-of-file) character, 73, 74

eof function

- BRexx, 365
- Regina, 339, 579

EPOC32 platform, 425, 426, 428–429

equals sign (=)

- assignment, 26
- equal comparison operator, 29

equals sign, double (==), strictly equal operator, 29

ERROR condition, 144, 148

error conditions

- list of, 144
- in Open Object Rexx, 472
- in OS TSO/E Rexx, 504
- in Reginald, 395
- in rool!, 454
- untrapped, default actions for, 148

.error object, Open Object Rexx, 470

error trapping

- affecting portability, 204
- call instruction for, 144–146, 152–154
- condition function for, 154, 155–156, 197, 552–553
- example using, 146–151
- generic routine for, 155–156
- guidelines for, 143–146, 179–180
- limitations of, 156
- for OS commands, 213
- Reginald features for, 395
- signal instruction for, 144–146, 147–148, 152–154
- special variables for, 151
- for SQL, 237–239

errors. See also debugging; error conditions

- insufficient storage error, 102
- overflow error, 28, 102
- underflow error, 28, 102

errortext function, 153, 197, 557

event handlers, Rexx/DW, 264

event-driven scripts, 259, 265

example programs

- address instruction, 218–219
- balanced parentheses, 118–120
- BRexx interpreter, ANSI screen I/O, 371–376
- BRexx interpreter, C-language I/O, 369–371
- BRexx interpreter, direct data access, 376–378
- BRexx interpreter, DOS functions, 378–382
- command procedures, 222–225
- database information, retrieving, 234–239
- database input, 85–89
- database tables, creating and loading, 239–241
- database tables, selecting results from, 241–244, 245–247
- database tables, updating, 243–244
- four letter words, identifying, 38–40
- interpreter for user input, 146–151
- key folding, 106–107
- mainframe Rexx, 506–511
- menu of transactions, 43–46
- NetRexx interpreter, applet, 525–526
- NetRexx interpreter, squaring a number, 523–525
- Number Game (random numbers), 21–24
- Open Object Rexx, concurrency, 489–491
- Open Object Rexx, file I/O, 476–477
- Open Object Rexx, squaring a number, 477–479
- Open Object Rexx, stack implementation, 481–485
- Open Object Rexx, user interaction, 479–481
- Open Object Rexx, video circulation, 485–489
- poetry scanner, 93–96
- Reginald interpreter, file and drive management, 400–404
- Reginald interpreter, MIDI files, 414–415
- Reginald interpreter, speech recognition, 412–414
- Reginald interpreter, Windows GUI, 404–412
- Reginald interpreter, Windows registry, 416–418
- Rexx for Palm OS interpreter, 435–444
- Rexx Server Pages (RSPs), 287–288
- Rexx/gd library, 268–271
- Rexx/imc C-language I/O functions, 354–356
- Rexx/imc environmental functions, 352–354
- Rexx/Tk package, 260–264
- RexxXML library, 299–302
- rightmost position of byte in string, 128–130

script environment, 200–201
 stack for interroutine communication, 165–166
 stack, using, 162–165
 telephone area code lookup, 61–62
 weighted retrieval, 58–60

Exception class, root!, 455

exception handling. See error trapping

exclamation (!), command identifier, 454

EXE Conversion Utility, r4 and root! interpreters, 451

EXEC interface, 505

EXECDROP command, CMS, 499

EXECIO command
 CMS, 499, 508, 510
 definition of, 504
 OS/TSO Rexx, 502

EXECLOAD command, CMS, 499

EXECMAP command, CMS, 499

EXECs, VM Rexx, 496

EXECSTAT command, CMS, 499

EXECUTIL command, OS/TSO Rexx, 502

exists function
 Regina, 339, 579
 root!, 454

exit instruction
 definition of, 539
 example using, 43, 46
 NetRexx, 630
 placement of, 113–114, 115, 186
 as structured construct, 34

exp function
 BRexx, 364
 Rexx/imc, 349

expand function, Reginald, 399

explicit concatenation, 80

exponential numbers, 25–26

export function, Regina, 338, 579

expose instruction, Open Object Rexx, 471, 481

exposed variables, 124–128

Extensible Markup Language (XML), 291–292. See also RexxXML library

Extensible Stylesheet Language Transformations (XSLT), 292

extensions, 110, 111

external access functions, Reginald, 398

external data queue. See stack

external functions
 accessing from BRexx, 366
 accessing from Regina, 339–342
 definition of, 120

external routines, 110

external subroutine, 41

ExternalClass class, root!, 455

externals function

mainframe Rexx, 593

VM Rexx, 497

F

>F> in trace file, 139

FAILURE condition, 144, 148

.false object, Open Object Rexx, 471

fdopen function, Rexx/imc, 351

FIFO (first-in, first-out) queue, 160–161

file associations for Windows interpreters, 326–327

File class, root!, 455

file functions

BRexx library for, 367

Rexx for Palm OS, 440

File Manager API, Palm OS, 440

fileno function, Rexx/imc, 351

FileRexx Function Library, 616

files

appending to, 76

closing, explicit, 69, 71

closing, implicit, 68

encryption/decryption of, Open Object Rexx, 473

end-of-file (EOF) character, 73, 74

opening, implicit, 68

positions of, moving explicitly, 72

positions of, moving implicitly, 68

redirecting command I/O to and from, 217

filesize function, HHNS WorkBench, 277

FileSpec function, Reginald, 391

FileUt package, 616

finalize method, root!, 453

find function

mainframe Rexx, 594

Regina, 337, 580

VM Rexx, 497

FINIS command, CMS, 499

first-in, first-out (FIFO) queue, 160–161

fixed-point (decimal) numbers, 26

floating point numbers, 26

flush function, BRexx, 365

FolderContext class, root!, 455

fork function, Regina, 338, 580

form function, 196, 557

form special name, NetRexx, 522, 629

FORMAT command, CMS, 510

format function, 102–103, 557–558

formatdate function, CGI/Rexx, 274
forums, 531–532
forward instruction, Open Object Rexx, 471
forward slash
 / (division operator), 27
 // (remainder division operator), 27
free-form (free-format) languages, 7, 23
freespace function, Regina, 338, 580
French, Rexx forums in, 532
FrmHdr function, HHNS WorkBench, 278
FrmInp function, HHNS WorkBench, 278
ftell function, Rexx/imc, 351, 356
fullurl function, CGI/Rexx, 274
The FUNCDEF feature, Reginald, 388
funcdef function, Reginald, 398
functions. See also call instruction; specific functions
 arguments of, 23, 25
 bit string functions, 96–97
 for BRexx, 363–366, 367–368
 built-in functions, 110, 120
 calling, 111–112, 184
 external functions, 120
 HHNS WorkBench, 278
 internal functions, 120
 list of, 547–571
 for mainframe Rexx, 593–595
 for Mod_Rexx, 282, 623–625
 nesting, 111–112, 174–175
 for numbers, 103–106
 for Open Object Rexx, 472
 passing parameters to, 116–117, 128, 185
 placement of, 113–115, 123, 177
 recursive, 121–123, 128–130
 for Regina, 335–339, 573–591
 for Reginald, 387–392, 398–399
 result of, 25, 111, 184–185
 for Rexx for Palm OS, 440
 for Rexx/imc, 348–356
 for Rexx/SQL, 233, 597–606
 for Rexx/Tk, 607–613
 for RexxXML, 293–294
 for rool, 448
 scope of variables in, 123–128
 search order for, 120
 string functions, 89–90, 92–97
 syntax for, 23
 user-defined functions, 110
 for VM Rexx, 497–498
fuzz function, 196, 558

G

Gargiulo, Gabriel (REXX in the TSO Environment), 513
GCS (Group Control System) interface and assembler macros, 505
gd package, 111
gdImageColorAllocate function, Rexx/gd, 267, 271
gdImageCreate function, Rexx/gd, 271
gdImageDestroy function, Rexx/gd, 267, 271
gdImageFilledRectangle function, Rexx/gd, 267
gdImageJpeg function, Rexx/gd, 267
gdImageLine function, Rexx/gd, 267
gdImageRectangle function, Rexx/gd, 267
German, Rexx forums in, 532
getch function, BRexx, 368
GetCookie function, HHNS WorkBench, 278
getcwd function, Rexx/imc, 348, 352
getenv function
 HHNS WorkBench, 277
 Regina, 338, 581
 Reginald, 398
 Rexx/imc, 348, 352
getfullhost function, CGI/Rexx, 274
getkwd function, HHNS WorkBench, 277
GETMSG function, OS/TSO Rexx, 501
getowner function, CGI/Rexx, 274
getpid function
 HHNS WorkBench, 277
 Regina, 338, 581
 Reginald, 398
getspace function, Regina, 338, 581
gettid function, Regina, 338, 582
GetTitle function, MIDI interface, 443
getwparm function, HHNS WorkBench, 277
Gimp Toolkit, 256
Glatt, Jeff (developer of Reginald), 312, 385
global variables, 127–128, 172, 178, 185
GLOBALV command, CMS, 499
glue languages, 4, 9, 434
gmImagePng function, Rexx/gd, 267
Goldberg, Gabe (The Rexx Handbook), 513
gotoxy function, BRexx, 368
Graphical Applications with Tcl and Tk (Johnson), 264
graphical user interface packages. See GUI packages
greater than operator (>), 29
greater than or equal to operator (>=), 29
greater than or less than operator (<> or ><), 29
Große-Coosmann, Florian (current developer for Regina), 312, 331

Group Control System (GCS) interface and assembler macros, 505

GTK+ package, 616

GTK toolkit, 256, 257

guard instruction, Open Object Rexx, 471

GUI packages. See also Windows GUI functions

list of, 255–257

Rexx/DW package, 256–257, 264–266, 616

Rexx/gd library, 266–271

Rexx/Tk package, 256, 258–264, 607–613, 617

GUI trace panel, Reginald, 395

H

Hack package, 616

HackMaster, 435

HALT condition, 144, 148

Handheld PC Magazine, 430

handheld platforms. See also Rexx for Palm OS

interpreter

BRexx support for, 367–368

definition of, 8, 10

emulation for, 423–425, 426–429

interpreters supporting, 314, 321–322, 425

native programming for, 423–425

operating systems for, 423

types of handheld devices, 422–423

“Harnessing Apache for REXX Programming” (Ashley), 288

hash function, Regina, 337, 582

HE command, OS/TSO Rexx, 502

help. See also documentation

for BRexx, 361

for Reginald, 387, 390

user groups, 531

for VM Rexx, 495

Web forums, 531–532

help command, example using, 222–225

Henri Henault & Sons Web site, 277

The Hessling Editor (THE), 183

Hessling, Mark (current developer for Regina), 312, 331

hexadecimal numbers, 26

HHNS WorkBench, 276–281, 616

HI command

OS/TSO Rexx, 502

VM Rexx, 498

high-level languages, 3

HT command

OS/TSO Rexx, 502

VM Rexx, 498

HTML CGI-scripting functions, BRexx library for, 367

HTML (Hypertext Markup Language), 292

htmlbot function, CGI/Rexx, 274, 276

htmlbreak function, CGI/Rexx, 274

HtmlGadgets package, 616

HtmlStrings package, 616

HTMLToolBar package, 616

htmltop function, CGI/Rexx, 274

httab function, CGI/Rexx, 274

HX command, VM Rexx, 498

Hypertext Markup Language (HTML), 292

I

IBM. See also mainframe Rexx

history of Rexx and, 308–309

Rexx interpreters for, 321

IBM DB2 UDB Administrative Reference API, 252

IBM DB2 UDB Application Development Guide, 252

IBM mainframe Rexx. See mainframe Rexx

IBM Object REXX interpreter, 257. See also Open Object Rexx interpreter

IBM Rexx Family Web site, 533

IBM Rexx Language Web site, 533

IDENTIFY command, CMS, 499, 508

if instruction

definition of, 35–37, 539

example using, 23, 38–40

NetRexx, 630

as structured construct, 34

if-else-if ladder, 36–37

IF-THEN construct, 34

IF-THEN-ELSE construct, 34

implicit concatenation, 80

import function

BRexx, 363

Regina, 338, 582

import instruction, NetRexx, 521, 631

IMS (Information Management System) Rexx interface, 505

indentation, 172–173

index function

mainframe Rexx, 594

Regina, 337, 582–583

VM Rexx, 497

indexed strings, NetRexx, 521–522

Information Management System (IMS) Rexx interface, 505

initialize method, rool!, 453

inkey function, HHNS WorkBench, 277

InLineFile **class, rool!, 455**

InOutLineFile **class, rool!, 455**

input

redirecting, 76

standard, 68

input instructions. See parse instruction; pull instruction

.input object, Open Object Rexx, 470

input/output. See I/O

insert function, 90, 558

install scripts, 209

instance creation operator (^ ^), 453

instance methods, Open Object Rexx, 466

instantiation, Open Object Rexx, 465, 478

InStream **class, rool!, 455**

instructions. See also specific instructions

list of, 535–546

multiple instructions on one line, 59

for NetRexx, 521, 630–633

for Open Object Rexx, 471

operands of, 25

for OS/TSO Rexx, 501

for Reginald, 394, 396–397, 398

for structured constructs, 34

for unstructured constructs, 47

for VM Rexx, 496–497

insufficient storage error, 102

integer division operator (%), 27

integers, 26. See also numbers

interactive debugging, 4

Interactive System Productivity Facility (ISPF), 183, 505

internal functions, 120. See also functions

internal routines, 110

internal subroutines, 41, 110. See also subroutines

international support, by Regina, 333

Internet/REXX HHNS WorkBench, 276–281, 616

interpret instruction

definition of, 540

example using, 146, 439

Reginald, 398

interpreted languages, 4

interpreters. See also compilers; tokenizers; specific interpreters

affecting portability, 203

choosing, 13–14, 313–321

commercial, 322–323

for embedded programming, 321–322, 430

free, 311–312

for handheld platforms, 314, 321–322, 425

for IBM, 321

information from, 202

for Java environment, 320

list of, 12–13

location of, as first line of script, 204

for mainframe platforms, 320

multiple, on one computer, 325–327

for new Rexx programmers, 13

object-oriented, 314, 319–320

standardization of, 8

thread-safe, 281–288, 333

intr function, BRexx, 363

I/O (input/output)

BRexx functions for, 364–366, 368–369

character-oriented, 68, 72–75, 191

command I/O, controlling, 216–219

command I/O, stack for, 225–226, 342, 368–369

conversational, 75

definition of, 67–69

line-oriented, 68, 69–72, 191

Open Object Rexx methods for, 476–477

OS-specific features for, 76–77

portability of, 76–77, 205

redirected, 75–76, 213–215

Regina functions for, 339, 342

Reginald functions for, 390–392

Rexx/imc functions for, 351, 354–356

standard input, 68

standard output, 68, 75

IrDA communications, 441–442

ISAM package, 111

iSeries OS/400 Rexx, 321

ISPEXEC interface, 505

ISPF (Interactive System Productivity Facility), 183, 505

ISREDIT interface, 505

iterate instruction

definition of, 47, 50, 540

NetRexx, 631

iterateexe function, Reginald, 399

J

Java applet, 525–526

Java Native Interface, for rool!, 454

Java, Rexx interpreter for. See NetRexx interpreter

Java Runtime Environment (JRE), 517

Java SDK (Software Development Toolkit), 517

Jaxo. See Rexx for Palm OS interpreter

JINI, for rool!, 454

Johnson, Eric F. (*Graphical Applications with Tcl and Tk*), 264

JRE (Java Runtime Environment), 517

justify function

- mainframe Rexx, 594
- Regina, 337, 583
- Reginald, 399
- Rexx/imc, 350
- VM Rexx, 497

K

kbhit function, BRexx, 368

key folding, 97, 106–107

keyboard handhelds, 422. See also handheld platforms

key-value pairs, 60–61, 63

Kiesel, P. (*Rexx: Advanced Techniques for Programmers*), 513

Kilowatt Software. See r4 interpreter; rool! interpreter

L

>L> in trace file, 139

labels. See also symbols

- for driver (main routine), 123
- for subroutine, 42–43
- as target of signal instruction, 49–50

last-in, first-out (LIFO) stack, 160

lastpos function, 90, 558

LD_LIBRARN32_PATH variable, 340

LD_LIBRARY_PATH variable, 340

leaf-node processing, 63

Learn REXX Programming in 56,479 Easy Steps, 386, 392

leave instruction

- definition of, 47, 48–49, 540
- NetRexx, 631

left function, 90, 559

length function, 40, 90–92, 559

length special name, NetRexx, 522, 629

less than operator (<), 29

less than or equal to operator (<=), 29

LIBPATH variable, 340

LIFO (last-in, first-out) stack, 160

line continuation character (,), 111, 122, 173, 186

line separation character (;), 173

linefeed character, 74

linein function

- definition of, 69, 559
- example using, 70

- explicitly controlling file positions, 72
- return string for, 180, 196

line-oriented I/O, 68, 69–72, 191

lineout function

- closing file with, 71
- definition of, 69, 559–560
- example using, 70
- explicitly controlling file positions, 72
- return string for, 180, 196

lines function

- definition of, 69, 71, 560
- example using, 70
- return string for, 180, 196–197, 205

linesize function

- mainframe Rexx, 595
- VM Rexx, 497

Linux platforms

- definition of, 8
- embedded Linux, 430
- tiny Linux, 430

List class

- Open Object Rexx, 468, 619
- rool!, 455

list (one-dimensional array), 53, 63

list processing, 79, 95

LISTDSI function, OS/TSO Rexx, 501

LISTFILE command, CMS, 499, 508

literals

- case sensitivity in, 22–23
- definition of, 26
- errors in, 184
- quotes in, 23, 26

ln function, Rexx/imc, 349

load function

- BRexx, 363
- Rexx for Palm OS, 439

LoadText function, Reginald, 391, 404

.local object, Open Object Rexx, 470

local variable, rool!, 453

log function, BRexx, 364

log10 function, BRexx, 364

logical AND operator (&), 30

logical EXCLUSIVE OR operator (&&), 30

logical NOT operator (\ or ~), 30

logical operations on binary strings, 97

logical operators, 30

logical OR operator (|), 30

loop instruction, NetRexx, 521, 524, 631

loop over instruction, rool!, 454

loops. See do instruction

LOSTDIGITS **condition**, 102, 144, 148, 472, 504

lower **function**

Regina, 337

rool!, 454

LU62 **interface**, 505

M

Mac OS **platforms**, 8

macro **language**, 111, 324–325

macro **programming**, 10

MacroEd **package**, 616

mainframe **platforms**

definition of, 8

interpreters for, 320

stack implementation, 166–168

mainframe **Rexx**

advantages of, 320

definition of, 493–494

example using, 506–511

extended functions for, 593–595

interfaces to, 504–506

migrating scripts to other platforms from, 512–513

OS/TSO **Rexx**, 500–503

platforms supported by, 493

standards supported by, 503–504

VM **Rexx**, 494–499

maintenance, 5, 6

MAKEBUF **command**

CMS, 499

mainframe **Rexx**, 166

OS/TSO **Rexx**, 502

makebuf **function**

BRexx, 363

Regina, 167, 342, 583

Reginald, 396, 399

rool!, 454

Map **class**, rool!, 455

MatchName **function**, Reginald, 391, 403–404, 411

Math **class**, rool!, 455

mathematical **applications**, using **Rexx** for, 11

mathematical **functions**

BRexx, 364

HHNS WorkBench, 278

Reginald, 387

Rexx/imc, 349

max **function**, 103–104, 560–561

McPhee, Patrick T. J. (author of **RexxXML**), 299

MenuObject **class**, Open Object **Rexx**, 622

Message **class**, Open Object **Rexx**, 469, 620

methget **function**, CGI/**Rexx**, 274

Method **class**, Open Object **Rexx**, 469, 620

::method **directive**, Open Object **Rexx**, 470

method **instruction**

NetRexx, 521, 631

rool!, 453

method invocation infix operator (~), 453

method invocation operator (~), 465, 466, 470

method invocation operator (~~), 466, 470

method invocation prefix operator (^), 453

methods

NetRexx, 520

Open Object **Rexx**, 465, 466, 476, 619–622

rool!, 452

.methods **object**, Open Object **Rexx**, 471

methpost **function**, CGI/**Rexx**, 274

MIDI **interface**, **Rexx** for Palm OS, 443–444

MIDI I/O **Function Library**, 616

MIDI **Rexx** **Function Library**, 387, 414–415, 616

MIDIctlName **function**, MIDI **Rexx**, 415

MIDIctlNum **function**, MIDI **Rexx**, 415

MIDIEventProp **function**, MIDI **Rexx**, 415

MIDIGetEvent **function**, MIDI **Rexx**, 415

MIDIGetGMDrum **function**, MIDI **Rexx**, 415

MIDIGetInfo **function**, MIDI **Rexx**, 415

MIDIGetVMPgm **function**, MIDI **Rexx**, 415

MIDINoteName **function**, MIDI **Rexx**, 415

MIDINoteNum **function**, MIDI **Rexx**, 415

MIDIOpenFile **function**, MIDI **Rexx**, 415

MIDIPortName **function**, MIDI **Rexx**, 415

MIDISaveFile **function**, MIDI **Rexx**, 415

MIDISetEvent **function**, MIDI **Rexx**, 415

MIDISysex **function**, MIDI **Rexx**, 415

MIDITrack **function**, MIDI **Rexx**, 415

migration, using **Rexx** for, 10

min **function**, 103–104, 561

minus sign (-)

negative number prefix operator, 27

subtraction operator, 27

mkdir **function**, BRexx, 368

mobile phones, 423. *See also* handheld platforms

mod operator (remainder division operator) (//), 27

Mod_Rexx **interface**

definition of, 281–282, 616

example using, 287–288

functions and variables in, 282, 623–627

installing, 282–283

resources for, 288

scripting with, 283–287

modularity

- definition of, 7, 33, 109
- example using, 118–120
- guidelines for, 177–178

Monitor **class**, **Open Object Rexx**, 469, 620

movefile function

- BRexx, 368
- Reginald, 391

MQ-Series Rexx interface, 505

MSG **function**, **OS/TSO Rexx**, 501

msgbox **function**, **BRexx**, 368

multiple inheritance, **Open Object Rexx**, 467

multiplication operator (*), 27

MutableBuffer **class**, **Open Object Rexx**, 469

MVS Forums, 532**MVS Help forum, 532**

MVSVAR **function**, **OS/TSO Rexx**, 501

MySQL **database**, 230, 249

MySQL I/O **functions**, **BRexx**, 365–366

myurl **function**, **CGI/Rexx**, 274

N

NAMEFIND **command**, **CMS**, 508

nap **function**, **rool**, 454

Nash, Simon (developer of **Open Object Rexx**), 312

native programming, for handhelds, 423–425, 434

natural language processing, 93–96

negative number prefix operator (-), 27

nesting functions, 111–112, 174–175

NetRexx interpreter

- advantages of, 317, 320, 515–516
- definition of, 13, 311, 312, 515
- documentation for, 517
- downloading and installing, 517–518
- example using, 523–526
- extra features in, 519–523
- instructions, 521, 630–633
- Java knowledge required for, 516–517
- Java software required for, 517
- running scripts, methods for, 518–519
- special methods, 523, 630
- special names, 522–523, 629

Netview, 505

Netware, **Rexx** for, 321

new **method**, **Open Object Rexx**, 465

newline character, 74

NEWSTACK **command**

- mainframe **Rexx**, 167
- OS/TSO **Rexx**, 502

newstack **function**, **rool**, 454

.nil **object**, **Open Object Rexx**, 470

Nirmal, B. (*REXX Tools and Techniques*), 513

NOMETHOD **condition**, 472

nonsparse arrays (dense arrays), 53

nop instruction

- definition of, 37, 541
- NetRexx, 632

NOSTRING **condition**, 472

not equal operator (\= or !=), 29

not greater than operator (\> or >), 29

not less than operator (\< or <), 29

“not” sign, **ANSI-standard**, 204, 496

NOTREADY **condition**, 144, 148

NOVALUE **condition**, 144, 148

null **special name**, **NetRexx**, 522, 629

numbers

- calculation results identical across platforms, 27, 99
- calculation rules for, 100–101
- conversion functions for, 105–106
- definition of, 25–26, 100
- errors from calculations, 28, 102
- example using, 106–107
- exponential notation, 101–102, 103
- functions for, list of, 103–104
- parsing by, 83–84
- significant digits (precision), 28, 101–102

numeric comparisons, 28

numeric digits **instruction**, 101–102

numeric form **instruction**, 102

numeric fuzz **instruction**, 101–102

numeric **instruction**

- definition of, 28, 541
- NetRexx, 632

O

>O> in trace file, 139

Object **class**

- Open Object **Rexx**, 469, 621
- rool, 455

Object **REXX GTK+ Project**, 257

Object **REXX interpreter**, 257, 310, 460. *See also* Open Object **Rexx interpreter**

ObjectCUR for Object **REXX**, 616

OBJECTION **condition**, 454

object-oriented interpreters, 314, 319–320. *See also*

- NetRexx **interpreter**; Open Object **Rexx interpreter**; r4 **interpreter**; rool **interpreter**

object-oriented programming, learning, 475
objects, Open Object Rexx, 465, 470–471
ODBC API (Open Database Connectivity Application Programming Interface), 230, 232
ODBC (Open Database Connectivity), connecting with, 249
ODBC (Open Database Connectivity) drivers, 387, 391, 616
OLE/ActiveX automation, Open Object Rexx, 473
OLEObject class, Open Object Rexx, 622
OODialog, 257
ooRexx. See Open Object Rexx interpreter
Open Database Connectivity Application Programming Interface (ODBC API), 230, 232
Open Database Connectivity (ODBC), connecting with, 249
Open Database Connectivity (ODBC) drivers, 387, 391, 616
open function
 BRexx, 365
 Regina, 339, 583–584
 Rexx/imc, 351, 355
Open Object Rexx interpreter (ooRexx)
 advantages of, 317, 319–320
 built-in objects, 470–471
 classes, 465–467, 468–469, 476, 619–622
 definition of, 13, 310, 312, 459
 directives, 470
 downloading and installing, 462–464
 environmental variables for, 464
 error conditions, 472
 example using, concurrency, 489–491
 example using, file I/O, 476–477
 example using, squaring a number, 477–479
 example using, stack implementation, 481–485
 example using, user interaction, 479–481
 example using, video circulation, 485–489
 features of, 460–462
 functions, 472
 history of, 460
 instructions, 471
 learning object-oriented programming with, 475
 object-oriented features of, 464–468
 operators, 469–470
 platforms supported by, 462, 472–473
 Rexx API, 472
 RexxUtil package, 472
 roo! interpreter as alternative to, 448
 special variables, 471
 Windows features, 472–473

OPENVM routines, 505
operands, of instructions, 25
operating system extensions, using Rexx for, 10
operating systems. See platforms
operatorless condition test, 120
operators
 arithmetic, 27–28
 comparison, 28–30
 concatenation operator (||), 30–31, 80
 logical, 30
 object-oriented, in Open Object Rexx, 469–470
 object-oriented, in roo!, 453–454
 order of precedence for, 31–32
options instruction
 definition of, 203, 334–335, 541–542
 NetRexx, 521, 632
 portability and, 205
 Reginald, 394
 VM Rexx, 496–497
Oracle database, connecting to, 248
oraenv function, CGI/Rexx, 274
order of precedence for operators, 31–32
OrderedVector class, roo!, 455
OS commands. See commands, OS
OS platforms, 8
OS simulation interface, 505
OS/2 platforms, 8
OS/2 Rexx
 definition of, 321
 functions supported by Regina, 336
OS/400 platforms, 8
OS/TSO Rexx, 500–503
otherwise keyword, select instruction, 40–41
Ousterhout, John (“Scripting: Higher Level Programming for the 21st Century”), 11
OutlineFile class, roo!, 455
output
 redirecting, 76
 standard, 68, 75
output function, 73
 .output **object, Open Object Rexx, 470**
OutputStream class, roo!, 455
OUTTRAP function, OS/TSO Rexx, 501
overflow error, 28, 102
overlay function, 90, 561

P

>P> in trace file, 139
package instruction, NetRexx, 521, 632

- packages for Rexx, list of, 615–618. See also specific packages**
- Palm OS platforms, 9, 423. See also handheld platforms; Rexx for Palm OS interpreter**
- parameters**
- example using, 118–120, 128–130
 - passing, Reginald features for, 396
 - passing to script on command line, 115–116
 - passing to subroutines and functions, 116–117, 128, 185
- parent classes, Open Object Rexx, 466–467**
- parentheses (())**
- affecting order of precedence, 31–32
 - enclosing function arguments, 23
- parse arg instruction, 115–117, 185, 195**
- parse instruction**
- definition of, 26, 542
 - NetRexx, 632
 - parsing by template, 82–85
 - system information strings returned by, 635
 - VM Rexx, 496–497
- parse linein instruction, 165**
- parse pull instruction**
- affecting stack, 160–162, 164
 - reading input, 39, 75
- parse source instruction, 198, 352**
- parse value instruction, Rexx/imc, 350**
- parse version instruction, 198–199, 352**
- PARSECMD command, CMS, 499**
- parsefid function, HHNS WorkBench, 277**
- parsing, 79–80, 81–89**
- Path function, Reginald, 391**
- Pattern class, rool!, 455**
- pattern matching, 80**
- pattern, parsing by, 82, 83**
- PatternMatch class, rool!, 456**
- PC-DOS Rexx, 321**
- pclose function, Rexx/imc, 351**
- PDAs (personal digital assistants). See handheld platforms**
- percent sign (%), integer division operator, 27**
- performance**
- of Apache Web server, 281
 - of BRexx, 359, 360
 - of database interface, 230, 247
 - of DOS emulation, 426–427
 - of I/O, portability and, 76–77
 - of Rexx/DW, 264
 - of Rexx/Tk, 256
 - of scripting language, 5–6, 11
- period (.), placeholder variable, 116**
- persistent streams, 67–68**
- personal digital assistants (PDAs). See handheld platforms**
- Personal Rexx (Quercus Systems), 323**
- PIPE command, CMS, 499**
- placeholder variable (.), 116**
- placeholder variable (?), 247, 252**
- platforms. See also portability; specific platforms**
- choosing interpreters based on, 310, 311–312, 313–314
 - OS-specific editors, 183
 - OS-specific I/O features, 99
 - retrieving information from, 202
 - supported by BRexx interpreter, 359
 - supported by mainframe Rexx, 493
 - supported by Open Object Rexx, 462, 472–473
 - supported by r4 and rool!, 447
 - supported by Regina, 14, 332
 - supported by Reginald, 385
 - supported by Rexx, 8–9, 309
 - supported by Rexx/imc, 345
- PlayASong function, MIDI interface, 443**
- plus sign**
- + (addition operator), 27
 - + (positive number prefix operator), 27
 - +++ (in trace file), 139
- Pocket PC Magazine, 430**
- pocket PCs. See handheld platforms**
- PocketConsole emulator, 424**
- PocketDOS emulator, 424, 427–428**
- polymorphism**
- NetRexx, 520
 - Open Object Rexx, 467
 - rool!, 452
- Poof! tool, r4 and rool! interpreters, 451**
- poolid function, Regina, 584**
- popen function**
- HHNS WorkBench, 277
 - Regina, 338, 584
 - Reginald, 394
 - Rexx/imc, 351
- portability. See also platforms**
- calculation results identical across platforms, 27, 99
 - command procedures and, 210
 - definition of, 190
 - example using, 200–201
 - factors affecting, 190–192, 202–205
 - I/O and, 76–77, 205

portability (continued)

migrating mainframe scripts to other platforms, 512–513

OS commands and, 191–192, 201–202

Rexx for, 10

RexxUtil package for, 206–207

script environment and, 195–199

standards and, 191, 192–195

pos function, 87, 90–92, 561

positive number prefix operator (+), 27

pow function, BRexx, 364

pow10 function, BRexx, 364

Prechelt, Lutz (“An Empirical Comparison of Seven Programming Languages”), 11

precision. See numeric instruction

prefix operators, 27

preinitialize method, root!, 453

printhead function, CGI/Rexx, 274, 276

printvariables function, CGI/Rexx, 274, 276

private methods, Open Object Rexx, 466

procedure expose instruction, 124–128, 177

procedure hide instruction, Rexx/imc, 350

procedure instruction, 124–128, 543

PROCESS construct, 34

program maintenance, 5, 6

Programming Language Rexx Standard, 532

programming style

- capitalization and, 170–171
- code reviews, 183–184
- comments, 175–177
- common coding errors, avoiding, 184–187
- error handling, 179–180
- global variables, 178
- methods of, 169–170
- modularity, 177–178
- nesting, 174–175
- Rexx-aware editors, 183
- site standards for, 183
- structured code, 178–179
- subscripts, 181
- variable naming, 171–172
- variables, declaring, 182
- white space (spacing and indentation), 172–173

Programming with REXX Dialog, 392

PROMPT function, OS/TSO Rexx, 501

properties instruction, NetRexx, 521, 632

properties, NetRexx, 520

protect instruction, NetRexx, 521

prototyping, using Rexx for, 10

public methods, Open Object Rexx, 466

publications. See also documentation; standards

- “An Empirical Comparison of Seven Programming Languages” (Prechelt), 11
- ANSI-1996, 8
- Embedded Systems Programming Magazine*, 430
- Graphical Applications with Tcl and Tk* (Johnson), 264
- Handheld PC Magazine*, 430
- “Harnessing Apache for REXX Programming” (Ashley), 288
- IBM DB2 UDB Administrative Reference API*, 252
- IBM DB2 UDB Application Development Guide*, 252
- Learn REXX Programming in 56,479 Easy Steps*, 386, 392
- Pocket PC Magazine*, 430
- Programming Language Rexx Standard*, 532
- Programming with REXX Dialog*, 392
- Rexx: Advanced Techniques for Programmers* (Kiesel), 513
- The Rexx Handbook* (Goldberg), 513
- REXX in the TSO Environment* (Gargiulo), 513
- The Rexx Language: A Practical Approach to Programming (TRL-1)* (Cowlshaw), 193
- The Rexx Language (TRL-2)* (Cowlshaw), 8, 193, 310, 532
- REXX Tools and Techniques* (Nirmal), 513
- REXX/VM Reference*, 494, 593
- REXX/VM User’s Guide*, 511
- RexxXML Usage and Reference*, 299
- “Scripting: Higher Level Programming for the 21st Century” (Ousterhout), 11
- Systems Application Architecture Common Programming Reference*, 494
- Tcl/Tk in a Nutshell* (Raines, Tranter), 264
- TSO/E REXX Reference*, 500, 593
- TSO/E REXX User’s Guide*, 511
- Using Mailslots with Reginald*, 392
- Using Reginald to Access the Internet*, 392
- Using Reginald with a Common Gateway Interface (CGI)*, 392
- Web sites listing, 533
- The World of Scripting Languages* (Barron), 11

pull instruction

- affecting stack, 160–162, 164
- compared to **parse pull** instruction, 39
- definition of, 23, 26, 81, 543

push instruction, 160–162, 163, 543

putenv function, Rexx/imc, 348

Q

- QBUF command**
 - mainframe Rexx, 166
 - OS/TSO Rexx, 502
- QELEM command, OS/TSO Rexx, 502**
- QSTACK command**
 - mainframe Rexx, 167
 - OS/TSO Rexx, 502
- qualify function**
 - definition of, 562
 - Reginald, 391
- Quercus Systems, Personal Rexx, 323**
- QUERY command, CMS, 499, 508**
- querymacro function, Reginald, 398**
- question mark (?), placeholder variable, 247, 252**
- questions at end of chapter, answers for, 637–655**
- queue. See stack**
- Queue class**
 - Open Object Rexx, 468, 620
 - rool, 456
- queue instruction, 160–162, 163–164, 543–544**
- queued function**
 - definition of, 160, 163, 164, 562
 - Reginald, 399
- quotes, double (“...”)**
 - enclosing character strings, 23, 26
 - enclosing OS commands, 181–182
- quotes, single (‘...’), enclosing character strings, 23, 26**

R

- r4 interpreter**
 - advantages of, 316, 319, 447–448
 - definition of, 13, 312, 447
 - documentation for, 447, 449
 - downloading and installing, 448–449
 - support for, 448
 - tools for, 450–451
 - Windows GUI functions, 448
- Raines, Paul (Tcl/Tk in a Nutshell), 264**
- raise instruction**
 - Open Object Rexx, 471, 472
 - Reginald, 395
- raise to a power operator (**), 27**
- raiseObjection function, rool, 454**
- random function**
 - definition of, 23, 25, 103–104, 562
 - Reginald, 399
- random numbers, example program using, 21–24**
- randu function, Regina, 338, 585**
- rc variable, 147, 151, 197, 211, 213**
- read function, BRexx, 365**
- read position, 68**
- readch function, Regina, 339, 585**
- readform function, CGI/Rexx, 274, 276**
- readln function, Regina, 339, 585**
- readpost function, CGI/Rexx, 274**
- real numbers, 26**
- real-time operating system (RTOS), 430**
- record-oriented files, 86**
- recursion, 121–123, 128–130**
- redirected I/O, 75–76, 213–215**
- Regina interpreter**
 - advantages of, 332–333
 - benefits of, 13–14
 - bit manipulation functions, 336–337
 - command I/O, stack for, 342
 - definition of, 12, 312, 317–318
 - documentation for, 14, 333
 - downloading, 14–15
 - environmental functions, 338
 - example using, 343
 - extended functions for, 335–339, 573–591
 - external function libraries, accessing, 339–342
 - extra features in, 316, 334
 - for handheld platforms, 425
 - history of, 331
 - installing, 15–19
 - international support, 333
 - interpreter options for, 334–335
 - I/O functions, 339
 - open source, 333
 - platforms supported by, 14, 332
 - Rexx API support, 333
 - SAA API, 343
 - stack functions, 342
 - standards supported by, 332
 - string manipulation functions, 337
 - supercompatibility with other interpreters, 333
 - superstacks, 333
 - support community for, 332
 - thread-safe, 333
- Regina Rexx language project, 531**
- Reginald interpreter**
 - Administration tool, 387, 399, 615
 - advantages of, 316, 318, 385–386

Reginald interpreter (continued)

array indexing, 397
definition of, 13, 312, 385
do over instruction, 396–397
documentation for, 386, 387, 392–393
downloading and installing, 386
error conditions, defining and raising, 395
example using, file and drive management, 400–404
example using, MIDI files, 414–415
example using, speech recognition, 412–414
example using, Windows GUI, 404–412
example using, Windows registry, 416–418
extended functions, 387–388
external access functions, 398
GUI trace panel, 395
interpret instruction, 398
I/O functions, 390–392
MIDI Rexx function library, 387, 414–415, 616
miscellaneous functions, 399
ODBC drivers, 391
options instruction, 394
OS commands, issuing, 393–394
parameter passing, 396
platforms supported by, 385
REXX Dialog, 387, 388–390, 404–409
REXX Dialog IDE, 390, 400
Rexx Text Editor (RexxEd), 183, 398, 399
Script Launcher, 399, 618
scripting with, 399–400
sorting array items, 396
speech function library, 412–414
SQLite driver, 387, 392
stack functions, 396, 398–399
standards supported by, 386
system information functions, 398
tools for, 386–387
Windows DLLs, 395–396
Windows GUI functions (REXX Dialog), 388–390, 404–412
Windows registry, accessing, 395, 416–418

Reginald Rexx Forum, 532
regular expressions library, Reginald, 388
Regular Expressions package, 616
RegUtil package, 617
Relation class, Open Object Rexx, 468, 620
relational database, 229. *See also* database
remainder division operator (//), 27
reply instruction, Open Object Rexx, 471
request record pointer, 286

::requires directive, Open Object Rexx, 470
resources. See also publications; Web sites
standards for Rexx, 8, 191, 192–195, 532
user groups, 531
Web forums, 531–532

response handlers, 284
REstructured eXtended eXecutor. See Rexx language
result variable
definition of, 41, 110, 111, 151, 197
example using, 112–113
signal instruction and, 152

return code. See rc variable
return instruction
definition of, 41, 43, 110, 115, 544
nesting, 111–112
NetRexx, 632
placement of, 186
signal instruction and, 152
as structured construct, 34

reverse function, 90, 123, 562

Revu tool, r4 and roo! interpreters, 451
REXREF3 package, 616
REXX. See mainframe Rexx
Rexx 2 Exe utility, 387, 616
Rexx: Advanced Techniques for Programmers (Kiesel), 513
Rexx API, 324, 333, 472
REXX Dialog IDE (RxDlgIDE), 390, 400
Rexx Dialog package, 257, 616
REXX Dialog, Reginald, 387, 388–390, 404–409
Rexx Exits interface, 505
Rexx for CICS, 320
Rexx for Palm OS Forum, 532
Rexx for Palm OS interpreter
advantages of, 317, 318, 434
database functions, 440
definition of, 13, 312, 433
downloading and installing, 434–435
file functions, 440
IrDA communications, 441–442
MIDI interface, 443–444
scripting with, 435–444
standards supported by, 434
TCP/IP communications, 442

The Rexx Handbook (Goldberg), 513
Rexx home page, 532
REXX in the TSO Environment (Gargiulo), 513
Rexx interpreters. See interpreters
Rexx LA (Rexx Language Association), 460, 531

Rexx language

- benefits of, 6–7
- elements of, 24–26
- features of, 7–8
- free implementations of, 310, 311–312
- future of, 327
- history of, 6, 189–190, 192–193, 308–311
- interpreter for, choosing, 13–14, 313–321
- limitations of, 11
- packages and tools supported by, 615–618
- platforms supported by, 8–9, 309
- standards for, 8
- uses of, 9–11, 12

The Rexx Language: A Practical Approach to Programming (TRL-1) (Cowlshaw), 193**Rexx Language Association (Rexx LA), 460****The Rexx Language (TRL-2) (Cowlshaw), 8, 193, 310, 532****Rexx Math Bumper Pack, 616****Rexx newsgroup forum, 531****Rexx Server Pages (RSPs), 287–288****Rexx Sockets, 505****Rexx Speech library, 412–414****Rexx Text Editor (RexxEd), 183, 387, 399, 617****REXX Tools and Techniques (Nirmal), 513****Rexx2Nrx package, 617****Rexx-aware editors, 183****Rexx/CGI library (cgi-lib.rxxx), 274–276, 615****Rexx/CURL package, 616****Rexx/Curses package, 616****Rexx/DB2 package, 250–253****Rexx/DW package, 256–257, 264–266, 616****RexxEd (Rexx Text Editor), 183, 387, 399, 617****Rexx/gd library, 266–271, 617****Rexx/imc interpreter**

- advantages of, 316, 318, 345–346
- C-language I/O functions, 351, 354–356
- definition of, 12, 312, 345
- documentation for, 346
- environmental functions, 348, 352–354
- example using, 352–356
- extra features in, 348–351
- installing, 346–348
- miscellaneous functions, 350
- packages and tools supported by, 351–352
- platforms supported by, 345
- SAA interface functions, 350
- stack functions, 350
- standards supported by, 345
- transcendental mathematical functions, 349

Rexx/ISAM package, 617**Rexxlets, 434****RexxMail package, 617****RexxRE package, 617****rexXMLFini function, RexxXML, 294****Rexx/SQL package**

- alternatives to, 250–253
- binding variables, 247, 249–250
- cursor processing, 245–247
- database connections, 233, 234–236, 248–250
- databases supported by, 229–230, 248–250
- definition of, 111, 617
- downloading, 231–232
- environmental control, 233
- environmental functions, 233
- error trapping in, 237–239
- features of, 230
- functions in, list of, 233, 597–606
- installing, 232
- issuing SQL statements, 233, 250
- tables, creating and loading, 239–241
- tables, selecting results from, 241–244, 245–247
- tables, updating, 243–244
- transactions, 233

RexxTags package, 617**Rexx/Tk package**

- definition of, 256, 258, 617
- downloading and installing, 258–259
- example using, 260–264
- functions in, list of, 607–613
- resources for, 264
- scripting with, 259, 264

Rexx/Trans package, 617**RexxUtil package, 206–207, 387, 472, 617****REXX/VM Reference, 494, 593****REXX/VM User's Guide, 511****Rexx/Wrapper package, 617****RexxXML library**

- applying stylesheet to document, 299
- definition of, 111, 292, 617
- downloading and installing, 295
- example using, 299–302
- features of, 292
- functions for, list of, 293–294
- licensing, 295
- loading, 296
- processing XML documents, 296–297, 298
- updating XML documents, 297
- validating documents against schemas, 298

RexxXML Usage and Reference, 299

rexxXMLInit function, RexxXML, 294

right function, 90, 122, 563

rmdir function, BRexx, 368

rool interpreter

- advantages of, 317, 319, 447–448
- definition of, 13, 310, 312, 447
- documentation for, 447, 449
- downloading and installing, 448–449
- object-oriented programming with, 452–456
- support for, 448
- tools for, 450–451
- Windows GUI functions, 448

::routine directive, Open Object Rexx, 470

routines. See functions; subroutines

.rs object, Open Object Rexx, 471

r4Sh function, HHNS WorkBench, 278

RSPs (Rexx Server Pages), 287–288

RT command

- OS/TSO Rexx, 502
- VM Rexx, 498

RTOS (real-time operating system), 430

RxAcc package, 617

RxBlowFish package, 617

RxCalibur package, 617

RxComm Serial Add-on package, 388, 617

RxCreate function, REXX Dialog, 389, 407

RXDDE package, 617

RxDlgDropFuncs function, REXX Dialog, 389

RxDlgIDE package, 617

RxDlgIDE (REXX Dialog IDE), 390, 400

RxDlgLoadFuncs function, REXX Dialog, 389

RxErr function, REXX Dialog, 389, 407

RxFile function, REXX Dialog, 389

rxfuncadd function

- Regina, 585–586
- Reginald, 398
- REXX Dialog, 389
- Rexx/DW, 265
- Rexx/gd, 267
- Rexx/Tk, 261
- SAA, 341, 350

rxfuncdrop function

- Regina, 586
- Reginald, 398
- SAA, 341, 350

rxfuncerrmsg function

- Regina, 586
- Reginald, 398
- SAA, 341

rxfuncquery function

- Regina, 586–587
- Reginald, 398
- SAA, 341, 350

RxInfo function, REXX Dialog, 389

rxJava package, 617

RxMakeShortcut function, REXX Dialog, 389

RxMsg function, REXX Dialog, 389, 408

RxProject package, 618

RxQuery function, REXX Dialog, 389

rxqueue executable, Regina, 342

rxqueue function

- Regina, 168, 338, 342, 587
- Reginald, 396, 399

RxRSync package, 618

RxRunRamScript function, REXX Dialog, 389

RxRunScript function, REXX Dialog, 389

RxSay function, REXX Dialog, 408

RxSet function, REXX Dialog, 389

RxSock package, 111, 388, 618

rxstack executable, Regina, 342

RxWav package, 618

S

SAA API

- definition of, 324
- Regina, 336, 343
- Reginald, 386
- Rexx/imc, 350

SAA (Systems Application Architecture) standard, 193, 309, 532

say instruction

- debugging with, 133–135
- default environment, determining, 636
- definition of, 45, 75, 544
- NetRexx, 633
- system information strings returned by, 635

sayn instruction, Rexx/imc, 350

scheduled tasks, 210

schema validation, with RexxXML, 294, 298

scoping, 123–130

screen interfaces, affecting portability, 203

Script Launcher, 399, 618

“Scripting: Higher Level Programming for the 21st Century” (Ousterhout), 11

scripting language

- definition of, 3–4
- performance of, 5–6, 11
- Rexx as, 308

- SearchPath **function, Reginald, 391**
- seek **function**
 BRexx, 365
 Regina, 339, 587
- select **instruction**
 as CASE construct, 34, 37
 definition of, 40–41, 544–545
 example using, 43–46
 NetRexx, 521, 633
 Rexx/imc, 350
- self **reference, rool, 453**
- self **variable, Open Object Rexx, 471**
- semicolon (;), **line separation character, 59, 173**
- SENTRIES **command, CMS, 499**
- Set **class**
 Open Object Rexx, 468, 620
 rool, 456
- SET **command, CMS, 499**
- SETLANG **function, OS/TSO Rexx, 501**
- SHARE **users group, 531**
- shared **variable, rool, 453**
- shell **language scripts, running Rexx as, 324**
- shell **scripts. See command procedures**
- SHLIB_PATH **variable, 340**
- show **function, Regina, 338, 587–588**
- sigl **variable, 151, 197**
- sign **function, 103–104, 563**
- signal **instruction**
 compared to call instruction, 152–154
 definition of, 47, 49–50, 545
 error trapping with, 144–146, 147–148
 mainframe Rexx, 504
 NetRexx, 633
- signal off **instruction, 148**
- signal on **instruction, 148**
- signatures, **NetRexx, 520**
- significant **digits. See numeric instruction**
- simple **symbols (variable names), 25, 54, 171–172**
- simple **variable name, 25**
- sin **function**
 BRexx, 364
 Rexx/imc, 349
- single **quotes ('...'), enclosing character strings, 23, 26**
- sinh **function, BRexx, 364**
- skewed **tree, 63**
- SLAC (Stanford Linear Accelerator Laboratory), **274**
- slacfnok **function, CGI/Rexx, 275**
- slash
 / (division operator), 27
 // (remainder division operator), 27
- sleep **function**
 Regina, 338, 588
 Reginald, 399
- smart **phones, 423. See also handheld platforms**
- Socket **class, rool, 456**
- SOCKET **function, VM Rexx, 498**
- SORT **command, CMS, 510**
- Sort **statement, Reginald, 396**
- source **special name, NetRexx, 522, 629**
- SourceForge.net **Web site, 258**
- sourceline **function, 150, 153, 196, 197, 563**
- space **function, 92, 563–564**
- spacing **and indentation, 172–173**
- sparse **arrays, 53**
- special **characters, I/O and, 74–75**
- special **methods, NetRexx, 523, 630**
- special **names, NetRexx, 522–523, 629**
- special **variables**
 for Mod_Rexx, 626–627
 for Open Object Rexx, 471
 rc variable, 147, 151, 197, 211, 213
 result variable, 41, 151, 197
 sigl variable, 151, 197
- Speech **Function Library, 387, 412–414, 618**
- SpeechClose **function, Rexx Speech, 414**
- SpeechOpen **function, Rexx Speech, 413**
- SpeechPitch **function, Rexx Speech, 414**
- SpeechSpeak **function, Rexx Speech, 413**
- SpeechSpeed **function, Rexx Speech, 414**
- SpeechVoiceDlg **function, Rexx Speech, 414**
- SpeechVolume **function, Rexx Speech, 414**
- split **function, rool, 454**
- SQL **Communications Area (SQLCA), 230, 238**
- SQL **statements. See also Rexx/SQL package**
 binding variables, 247, 249–250
 issuing, 233, 250, 251
 support for, 230
 tables, creating and loading, 239–241
 tables, selecting results from, 241–244, 245–247
 tables, updating, 243–244
- SQLCA (SQL **Communications Area), 230, 238**
- SqlClose **function, Rexx/SQL, 233, 245, 597**
- SqlCommand **function, Rexx/SQL, 233, 597–598**
- SqlCommit **function, Rexx/SQL, 233, 598**
- SqlConnect **function, Rexx/SQL, 233, 598–599**
- SqlDefault **function, Rexx/SQL, 233, 599**
- SqlDescribe **function, Rexx/SQL, 233, 245, 599–600**
- SqlDisconnect **function, Rexx/SQL, 233, 600**
- SqlDispose **function, Rexx/SQL, 233, 601**
- SqlDropFuncs **function, Rexx/SQL, 601**

- SqlExecute **function, Rexx/SQL, 233, 601**
- SqlFetch **function, Rexx/SQL, 233, 245, 602**
- SqlGetData **function, Rexx/SQL, 233, 602–603**
- SqlGetInfo **function, Rexx/SQL, 233, 603**
- SQLite driver, for Reginald, 387, 392**
- SqlLoadFuncs **function, Rexx/SQL, 603–604**
- SqlOpen **function, Rexx/SQL, 233, 245, 604**
- SqlPrepare **function, Rexx/SQL, 233, 245, 604**
- SqlRollback **function, Rexx/SQL, 233, 605**
- SqlVariable **function, Rexx/SQL, 233, 605–606**
- sqrt **function**
 - BRexx, 364
 - Rexx/imc, 349
- squareRoot **function, rool, 454**
- S/Rexx (Treehouse Software Inc.), 323**
- stack**
 - affecting portability, 205
 - BRexx functions for, 363
 - buffers and, 167–168
 - for command I/O, 225–226, 342
 - definition of, 159–162
 - example using, 162–166
 - instructions affecting, 160
 - maximum size of, 162
 - multiple, in Reginald, 396
 - number of items in, 160
 - object-oriented, in Open Object Rexx, 481–485
 - portability of, 166–168
 - Regina functions for, 342
 - Reginald functions for, 396, 398–399
 - Rexx/imc functions for, 350
 - superstacks, in Regina, 333
- Stack **class, rool, 456**
- standard input, 68**
- standard output, 68, 75**
- standards**
 - history of, 192–195
 - list of, 8, 532
 - for mainframe Rexx, 503–504
 - portability and, 191, 195
- Standord Linear Accelerator Laboratory (SLAC), 274**
- state **function**
 - Regina, 338, 588
 - Reginald, 391
- statements. See instructions**
- static **variable, rool, 453**
- .stderr **object, Open Object Rexx, 471**
- .stdin **object, Open Object Rexx, 471**
- .stdout **object, Open Object Rexx, 471**
- stem **class, Open Object Rexx, 469, 621**
- stem variables (array names), 55**
- stemdelete **function, Reginald, 399**
- steminsert **function, Reginald, 399**
- storage **function**
 - BRexx, 363
 - OS/TSO Rexx, 501
 - Regina, 338, 589
 - VM Rexx, 498
- Stream **class, Open Object Rexx, 469, 621**
- stream **function**
 - BRexx, 365
 - definition of, 71–72, 196, 564
 - portability and, 205
 - Regina, 339, 589
 - Rexx/imc, 351
 - VM Rexx, 497
- stream instance, 477**
- streams, I/O, 67–68, 216, 217**
- strict comparison operators, 29–30, 187**
- strictly equal operator (==), 29**
- strictly greater than operator (>), 29**
- strictly greater than or equal to operator (>=), 29**
- strictly less than operator (<), 29**
- strictly less than or equal to operator (<=), 29**
- strictly not equal operator (\== or \==), 29**
- strictly not greater than operator (\> or \>), 29**
- strictly not less than operator (\<< or \<<), 29**
- String **class, Open Object Rexx, 469, 621**
- string comparisons, 28**
- string delimiters, 26**
- string manipulation, 79–80, 337**
- string processing, 79**
- strings**
 - bit string functions, 96–97
 - concatenating, 79–80
 - example using, 85–89, 93–96
 - functions for, list of, 89–90
 - in literals, 23
 - parsing, 79–80, 81–85
 - word-oriented functions for, 92–96
- strip **function, 87–88, 90, 564**
- striphtml **function, CGI/Rexx, 275**
- structured programming, 7, 33–34, 178–179**
- study question answers, 637–655**
- style. See programming style**
- stylesheets. See XSLT (Extensible Stylesheet Language Transformations)**
- subclasses, Open Object Rexx, 466–467**

SUBCOM command, OS/TSO Rexx, 502
subroutines. See also call instruction
 calling, 112
 definition of, 41
 error handling using, 144–146
 example using, 43–46
 label for, 42–43
 passing parameters to, 116–117, 128, 185
 placement of, 113–115, 123, 177
 recursive, 121–123
 result of, 112–113
 scope of variables in, 123–128
 types of, 41
subscripts, 181
substr function, 88–89, 90, 565
subtraction operator (-), 27
subword function, 93, 565
super special method, NetRexx, 523, 630
super special name, NetRexx, 522, 629
super variable, Open Object Rexx, 471
superclasses, Open Object Rexx, 466–467
superstacks, Regina support for, 333
Supplier class, Open Object Rexx, 469, 621
suspect function, CGI/Rexx, 275
Symbian OS platforms, 9, 423, 425, 428–429
symbol function, 112, 565
symbolic pointers, 63
symbols. See also labels
 compound (array names), 54–55
 simple (variable names), 25, 54
SYNTAX condition, 144, 148
SYSCPUS function, OS/TSO Rexx, 501
SYSDSN function, OS/TSO Rexx, 501
System class, rool, 456
system function, Rexx/imc, 348, 352
system functions
 BRexx, 363
 Reginald, 398
 Rexx/imc, 348, 352
system information strings, 635
SystemPropertyMap class, rool, 456
systems administration, using Rexx for, 10
Systems Application Architecture Common Programming Reference, 494
Systems Application Architecture (SAA) standard, 193, 309, 532
systems programming languages, 11
SYSVAR function, OS/TSO Rexx, 501

T

Table class
 Open Object Rexx, 468, 620
 rool, 456
tables (arrays). See arrays
tables (database)
 creating and loading, 239–241
 selecting results from, 241–244, 245–247
 updating, 243–244
tablet PC, 422, 423. See also handheld platforms
Tags function, HHNS WorkBench, 278
tail of compound symbol, 55
Talkabout Network, 531
tan function
 BRexx, 364
 Rexx/imc, 349
tanh function, BRexx, 364
TblHdr function, HHNS WorkBench, 278
Tcl Developer Exchange Web site, 258
Tcl/Tk in a Nutshell (Raines, Tranter), 264
Tcl/Tk scripting language, GUI package using. See Rexx/Tk package
TCP/IP communications, Rexx for Palm OS, 442
TE command
 OS/TSO Rexx, 502
 VM Rexx, 498
Tek-Tips Rexx Forum, 532
template, parsing by, 82–85
terminate method, rool, 453
testing. See debugging
text processing, 10. See also string manipulation; strings
textual analysis, 93–96
THE (The Hessling Editor), 183, 618
this special method, NetRexx, 523, 630
this special name, NetRexx, 523, 629
thread-safe interpreter, 281–288, 333
tilde, double (~~), method invocation, 466, 470
tilde (~), method invocation, 453, 465, 466, 470
time function
 definition of, 196, 566
 mainframe Rexx, 504
tiny Linux, 430
Tk, GUI package using. See Rexx/Tk package
TK package, 111
TkActivate function, Rexx/Tk, 607
TkAdd function, Rexx/Tk, 262–263, 607

- TkAfter **function, Rexx/Tk, 607**
- TkBbox **function, Rexx/Tk, 607**
- TkButton **function, Rexx/Tk, 607**
- TkCanvas **functions, Rexx/Tk, 607–608**
- TkCget **function, Rexx/Tk, 608**
- TkCheckButton **function, Rexx/Tk, 608**
- TkChooseColor **function, Rexx/Tk, 608**
- TkChooseDirectory **function, Rexx/Tk, 608**
- TkComboBox **functions, Rexx/Tk, 612**
- TkConfig **function, Rexx/Tk, 608**
- TkCurSelection **function, Rexx/Tk, 608**
- TkDelete **function, Rexx/Tk, 608**
- TkDestroy **function, Rexx/Tk, 608**
- TkDropFuncs **function, Rexx/Tk, 263, 611**
- TkEntry **function, Rexx/Tk, 608**
- TkError **function, Rexx/Tk, 258, 608**
- TkEvent **function, Rexx/Tk, 608**
- TkFocus **function, Rexx/Tk, 608**
- TkFont **functions, Rexx/Tk, 608–609**
- TkFrame **function, Rexx/Tk, 609**
- TkGet **function, Rexx/Tk, 609**
- TkGetOpenFile **function, Rexx/Tk, 263, 609**
- TkGetSaveFile **function, Rexx/Tk, 609**
- TkGrab **function, Rexx/Tk, 609**
- TkGrid **functions, Rexx/Tk, 609**
- TkImageBitmap **function, Rexx/Tk, 609**
- TkImagePhoto **function, Rexx/Tk, 609**
- TkIndex **function, Rexx/Tk, 609**
- TkInsert **function, Rexx/Tk, 609**
- TkItemConfig **function, Rexx/Tk, 609**
- TkLabel **function, Rexx/Tk, 609**
- TkListbox **function, Rexx/Tk, 609**
- TkLoadFuncs **function, Rexx/Tk, 261, 611**
- TkLower **function, Rexx/Tk, 610**
- TkMCListbox **functions, Rexx/Tk, 612–613**
- TkMenu **functions, Rexx/Tk, 610**
- TkMessageBox **function, Rexx/Tk, 263, 610**
- TkNearest **function, Rexx/Tk, 610**
- TkPack **function, Rexx/Tk, 610**
- TkPopup **function, Rexx/Tk, 610**
- TkRadioButton **function, Rexx/Tk, 610**
- TkRaise **function, Rexx/Tk, 610**
- TkScale **function, Rexx/Tk, 610**
- TkScan **function, Rexx/Tk, 610**
- TkScrollbar **function, Rexx/Tk, 610**
- TkSee **function, Rexx/Tk, 610**
- TkSelection **function, Rexx/Tk, 610**
- TkSet **function, Rexx/Tk, 610**
- TkSetFileType **function, Rexx/Tk, 610**
- TkTcl **function, Rexx/Tk, 610**
- TkText **function, Rexx/Tk, 610**
- TkTextTagBind **function, Rexx/Tk, 611**
- TkTextTagConfig **function, Rexx/Tk, 611**
- TkTopLevel **function, Rexx/Tk, 611**
- TkTree **functions, Rexx/Tk, 611–612**
- TkVar **function, Rexx/Tk, 611**
- TkVariable **function, Rexx/Tk, 611**
- TkWait **function, Rexx/Tk, 611**
- TkWinInfo **function, Rexx/Tk, 611**
- TkWm **function, Rexx/Tk, 611**
- TkXView **function, Rexx/Tk, 611**
- TkYView **function, Rexx/Tk, 611**
- tokenized scripts, **Open Object Rexx, 473**
- tokenizers, **324**
- tools, **list of, 615–618**
- TopHat tools, **r4 and roo! interpreters, 451**
- topower **function, Rexx/imc, 349**
- trace **function, 139–140, 196, 197, 566–567**
- trace **instruction**
 - definition of, **135–139, 140–142, 545–546**
 - NetRexx, **521, 633**
- trace **panel, Reginald, 395**
- trace **special name, NetRexx, 523, 629**
- trailing **comments, 22**
- transactions, **Rexx/SQL, 233**
- transcendental **mathematical functions. See mathematical functions**
- transient **streams, 67**
- translate **function, 90–92, 95, 567**
- Tranter, **Jeff (Tcl/Tk in a Nutshell), 264**
- Tree **class, roo!, 456**
- Treehouse Software Inc., **S/Rexx, 323**
- trees, **63–64**
- trim **function, Regina, 337, 589**
- TRL-1 **standard, 193**
- TRL-1 (*The Rexx Language: A Practical Approach to Programming*) (Cowlshaw), **193**
- TRL-2 **standard, 193–195**
- TRL-2 (*The Rexx Language*) (Cowlshaw), **8, 193, 310, 532**
- .true **object, Open Object Rexx, 471**
- trunc **function, 103–104, 567–568**
- TS **command**
 - OS/TSO Rexx, **502**
 - VM Rexx, **498**
- TSO/E Rexx, **500–503**
- TSO/E REXX **Reference, 500, 593**
- TSO/E REXX **User's Guide, 511**
- Twocows Inc. **Web site, 360**
- typeless **variables, 4, 25**

U

u2a function, BRexx, 368
uname function
 Regina, 338, 590
 Reginald, 398
unbalanced tree, 63
underflow error, 28, 102
underscore (_), in variable names, 171
Unicode support, Open Object Rexx, 473
Uniform Resource Identifier (URI), 436
uninitialized variables, 26
uni-Rexx (The Workstation Group), 275, 323
universal languages, 8–9
Unix platforms, 8
unixerror function
 Regina, 338, 590
 Reginald, 398
unstructured programming, 47–50, 179
upper camel case, 171
upper function
 Regina, 337, 590–591
 rool, 454
upper instruction, VM Rexx, 496–497
uppercase. See case sensitivity
URI (Uniform Resource Identifier), 436
URLs, retrieving data from, with RexxXML, 294
Use Arg function, Reginald, 396
use arg instruction, Open Object Rexx, 466
use instruction, Open Object Rexx, 471
USER condition, 472
user groups, 531
user-defined functions, 110
userid function
 mainframe Rexx, 595
 Regina, 338, 591
 Reginald, 398
 Rexx/imc, 348, 352
 VM Rexx, 497
Using Mailslots with Reginald, 392
Using Reginald to Access the Internet, 392
Using Reginald with a Common Gateway Interface (CGI), 392

V

>V> in trace file, 138, 139
value function
 definition of, 568
 Reginald, 395, 416–418
ValueIn function, Reginald, 391

ValueOut function, Reginald, 391
vardump function, BRexx, 363
variable management, 4
variable names (simple symbols), 25, 54, 171–172
variables
 assigning, 25, 26
 binding, for SQL, 247, 249–250
 data type of, 23, 25
 declaration of, 23, 25, 172, 182
 definition of, 25
 exposed, 124–128
 global, 127–128
 placeholder variable (.), 116
 scope of, 123–128
 typeless, 4, 25
 uninitialized, 26
 uninitialized, 115
variables (attributes), Open Object Rexx, 465
vector class reference operator ({}), 454
Vector class, rool, 456
verify function, 88, 90–92, 568
version special name, NetRexx, 523, 629
vertical applications, r4 and rool interpreters, 451
vertical bar, double (||), concatenation operator, 30–31, 80
vertical bar (|), logical OR operator, 30
VisPro Rexx interface, 257
Vlachoudis, Vasilis (inventor of BRexx), 312, 359
VM GUI interface, 505
VM platforms, 8
VM Rexx
 ANSI-standard “not” sign, 496
 CMS commands, 499
 CMS immediate commands, 498
 comment on first line of script, 494
 compilers, 498–499
 enabling buffer functions for, 573
 file types, 496
 functions, 497–498
 instructions, 496–497
 online help, 495
 OS commands, 496
VSAMIO interface, 505
VSE platforms, 8
VSE simulation interface, 505
VX*Rexx interface, 257

W

W32 Funcs package, 618
WAVV Forum, 532

Web forums, 531–532

Web servers, programming

- Apache Web server, 281–288
- with CGI, 273–281
- methods for, 273

Web sites

- Amiga Forum, 532
- AROS, 323
- Bochs emulator, 424
- BRexx interpreter, 360–361
- Code Comments community, 531
- DBForums, 531
- embedded programming, 430
- handheld devices, 430
- Henri Henault & Sons Web site, 277
- IBM DeveloperWorks, 288
- IBM Rexx Family, 533
- IBM Rexx Language, 533
- IBM Rexx manuals, 533
- Mod_Rexx interface, 282, 288
- MVS Forums, 532
- MVS Help forum, 532
- NetRexx interpreter, 517
- Open Object Rexx interpreter, 460, 462
- PocketConsole emulator, 424
- PocketDOS emulator, 427
- Quercus Systems, 323
- r4 interpreter, 448
- Regina interpreter, 14
- Regina Rexx language project, 531
- Reginald interpreter, 386
- Reginald Rexx Forum, 532
- Rexx for Palm OS interpreter, 434–435, 532
- Rexx home page, 532
- Rexx LA, 531
- Rexx/DW package, 257, 265
- Rexx/gd library, 267
- Rexx/imc interpreter, 346, 352
- Rexx/Tk package, 256, 258
- roo! interpreter, 448
- SHARE users group, 531
- SLAC (Stanford Linear Accelerator Laboratory), 274
- SQLite, 392
- Talkabout Network, 531
- Tek-Tips Rexx Forum, 532
- Treehouse Software Inc., 323
- WAVV Forum, 532
- The Workstation Group, 323
- X-Master, 435
- XTM emulator, 428

webify function, CGI/Rexx, 275

Wegina package, 618

wherex function, BRexx, 368

wherey function, BRexx, 368

white space, 23, 172–173

whole numbers (integers), 26

WideCharacterVector class, roo!, 456

widgerts, Rexx/DW, 256, 264

WindowObject class, Open Object Rexx, 622

Windows CE platforms. See also handheld platforms

BRexx functions for, 367–368

definition of, 9, 423

Windows DLLs, Reginald, 395–396

Windows GUI functions. See also GUI packages

r4 and roo! interpreters, 448

Reginald, 388–390, 404–412

Windows Internet API, Reginald, 388

Windows platforms

definition of, 8

installing Regina interpreter on, 15–16

I/O redirection on, 76

multiple interpreters running on, 326–327

Open Object Rexx classes for, 621–622

Open Object Rexx support for, 472–473

Reginald support for, 385–386

return codes for OS commands, 154

Windows registry, accessing with Reginald, 395, 416–418

Windows Scripting Host (WSH), 473

WindowsClipboard class, Open Object Rexx, 622

WindowsEventLog class, Open Object Rexx, 622

WindowsManager class, Open Object Rexx, 622

WindowsProgramManager class, Open Object Rexx, 621

WindowsRegistry class, Open Object Rexx, 621

windowtitle function, BRexx, 368

word function, 93, 569

wordindex function, 93, 569

wordlength function, 93, 95, 569

word-oriented functions, 92–96

wordpos function, 93, 95, 569

words

definition of, 92

parsing by, 82–83

words function, 93, 95, 570

The Workstation Group, uni-Rexx, 275, 323

The World of Scripting Languages (Barron), 11

wraplines function, CGI/Rexx, 275

write function, BRexx, 365

write position, 68

- writetech function, Regina, 339, 591**
writeln function, Regina, 339, 591
WSH (Windows Scripting Host), 473
WWWAddCookie function, Mod_Rexx, 623
WWWARGS special variable, Mod_Rexx, 626
WWWAUTH_TYPE special variable, Mod_Rexx, 626
WWWConnRecAborted function, Mod_Rexx, 625
WWWConstruct_URL function, Mod_Rexx, 623
WWWCONTENT_LENGTH special variable, Mod_Rexx, 626
WWWCONTENT_TYPE special variable, Mod_Rexx, 626
WWWCOOKIES special variable, Mod_Rexx, 626
WWWDEFAULT_TYPE special variable, Mod_Rexx, 626
WWWEscape_Path function, Mod_Rexx, 623
WWWFILENAME special variable, Mod_Rexx, 626
WWWFNAMETEMPLATE special variable, Mod_Rexx, 626
WWWGATEWAY_INTERFACE special variable, Mod_Rexx, 626
WWWGetArgs function, Mod_Rexx, 286, 623
WWWGetCookies function, Mod_Rexx, 623
WWWGetVersion function, Mod_Rexx, 286, 623
WWWHOSTNAME special variable, Mod_Rexx, 626
WWWHTTP_time function, Mod_Rexx, 623
WWWHTTP_USER_ACCEPT special variable, Mod_Rexx, 626
WWWHTTP_USER_AGENT special variable, Mod_Rexx, 626
WWWInternal_Redirect function, Mod_Rexx, 623
WWWIS_MAIN_REQUEST special variable, Mod_Rexx, 626
WWWLogError function, Mod_Rexx, 623
WWWLogInfo function, Mod_Rexx, 623
WWWLogWarning function, Mod_Rexx, 624
WWWPATH_INFO special variable, Mod_Rexx, 626
WWWPATH_TRANSLATED special variable, Mod_Rexx, 626
WWWPOST_STRING special variable, Mod_Rexx, 626
WWWQUERY_STRING special variable, Mod_Rexx, 627
WWWREMOTE_ADDR special variable, Mod_Rexx, 627
WWWREMOTE_HOST special variable, Mod_Rexx, 627
WWWREMOTE_IDENT special variable, Mod_Rexx, 627
WWWREMOTE_USER special variable, Mod_Rexx, 627
WWWReqRec functions, Mod_Rexx, 624–625
WWWREQUEST_METHOD special variable, Mod_Rexx, 627
WWWSPCOMPILER special variable, Mod_Rexx, 627
WWWRun_Sub_Req function, Mod_Rexx, 624
WWWSCRIPT_NAME special variable, Mod_Rexx, 627
WWWSendHTTPHeader function, Mod_Rexx, 286, 624
WWWSERVER_NAME special variable, Mod_Rexx, 627
WWWSERVER_PORT special variable, Mod_Rexx, 627
WWWSERVER_PROTOCOL special variable, Mod_Rexx, 627
WWWSERVER_ROOT special variable, Mod_Rexx, 627
WWWSERVER_SOFTWARE special variable, Mod_Rexx, 627
WWWSetHeaderValue function, Mod_Rexx, 624
WWWSTrvRec functions, Mod_Rexx, 625
WWWSub_Req_Lookup_File function, Mod_Rexx, 624
WWWSub_Req_Lookup_URI function, Mod_Rexx, 624
WWWUNPARSEDURI special variable, Mod_Rexx, 627
WWWURI special variable, Mod_Rexx, 627
- ## X
- x2b function, 97, 105, 570**
x2c function, 105, 570–571
x2d function, 105, 571
XEDIT command, CMS, 499
XEDIT editor, 183, 505
XEDIT macros, VM REXX, 496
X-Master, 435
XML (Extensible Markup Language), 291–292. See also REXXXML library
xmlAddAttribute function, REXXXML, 293, 297
xmlAddComment function, REXXXML, 293, 297
xmlAddElement function, REXXXML, 293, 297
xmlAddNode function, REXXXML, 293, 297
xmlAddPI function, REXXXML, 293, 297
xmlAddText function, REXXXML, 293, 297
xmlApplyStylesheet function, REXXXML, 294, 299
xmlCompileExpression function, REXXXML, 294
xmlCopyNode function, REXXXML, 293, 297
xmlDropFuncs function, REXXXML, 293
xmlDumpSchema function, REXXXML, 294, 298
xmlError function, REXXXML, 293, 298, 301
xmlEvalExpression function, REXXXML, 294, 302
xmlExpandNode function, REXXXML, 293, 297
xmlFindNode function, REXXXML, 294, 298, 302
xmlFree function, REXXXML, 293
xmlFreeContext function, REXXXML, 294
xmlFreeDoc function, REXXXML, 293, 296
xmlFreeExpression function, REXXXML, 294
xmlFreeSchema function, REXXXML, 294, 298
xmlFreeStylesheet function, REXXXML, 294, 299
XMLGenie! tool, r4 and rool interpreters, 451
xmlGet function, REXXXML, 294
xmlLoadFuncs function, REXXXML, 293, 296
xmlNewContext function, REXXXML, 294
xmlNewDoc function, REXXXML, 293

xmlNewHTML function, REXXXML

- `xmlNewHTML` **function, REXXXML, 293**
- `xmlNodeContent` **function, REXXXML, 293**
- `xmlNodesetAdd` **function, REXXXML, 294**
- `xmlNodesetCount` **function, REXXXML, 294**
- `xmlNodesetItem` **function, REXXXML, 294, 298**
- `xmlOutputMethod` **function, REXXXML, 294, 299**
- `xmlParseHTML` **function, REXXXML, 293, 301**
- `xmlParseSchema` **function, REXXXML, 294, 298**
- `xmlParseXML` **function, REXXXML, 293, 296, 298**
- `xmlParseXSLT` **function, REXXXML, 294, 299**
- `xmlPost` **function, REXXXML, 294**
- `xmlRemoveAttribute` **function, REXXXML, 293, 297**
- `xmlRemoveContent` **function, REXXXML, 293, 297**
- `xmlRemoveNode` **function, REXXXML, 293, 297**
- `xmlSaveDoc` **function, REXXXML, 293, 296**
- `xmlSetContext` **function, REXXXML, 294**
- `xmlValidateDoc` **function, REXXXML, 294, 298**
- `xmlVersion` **function, REXXXML, 293, 301**
- X/Open CLI, 230, 232**
- XPath, 292, 298**
- `xrange` **function, 90, 96–97, 570**
- XSLT (Extensible Stylesheet Language Transformations), 292, 294, 299**
- XTM emulator, 424, 428–429**