

Table of Contents

Preface		v
1. Introduction		
1.1	REXX background	1
1.2	REXX characteristics	1
1.3	REXX applications	2
1.4	REXX's future	3
2. REXX Facilities		
2.1	Introduction	4
2.2	REXX structure	8
2.2.1	Token	8
2.2.2	Literal string	8
2.2.3	Hexadecimal literal string	9
2.2.4	Binary literal string	9
2.2.5	Symbol	10
2.2.6	Numeric symbol	10
2.2.7	Variables	11
2.2.8	Operator	12
2.2.9	Prefix operators	13
2.2.10	Arithmetic operators	14
2.2.11	Abuttal operator	14
2.2.12	Comparative operators	14
2.2.13	Logical operators	15
2.2.14	Implicit concatenation	16
2.2.15	Other syntax elements	16
2.3	REXX expressions	17

2.3.1	Arithmetic expression	19
2.3.2	Comparative expression	18
2.3.3	Logical (Boolean) expression	19
2.4	Statements	19
2.5	Invocation of a REXX exec	21
2.6	Comparison with CLISTS	21

3. REXX Processing Elements

3.1	Introduction	22
3.2	Statements	22
3.3	Routines	24
3.3.1	Subroutine	25
3.3.2	Procedure	25
3.3.3	Function	26
3.3.4	Search order	27
3.3.5	Invocation	28
3.3.6	Parameters (arguments)	29
3.3.7	Return	32
3.3.8	Communication between execs	32
3.4	Types of instruction	35
3.4.1	Conditional control	35
3.4.2	Loop control	35
3.5	Control processing flow	39
3.6	Assignment	39
3.7	Stack (queue) processing	40
3.7.1	Stack as terminal input buffer	40
3.7.2	Stack used for general data storage	41
3.7.3	Stack used for passing data between execs	41
3.7.4	File buffer	43
3.8	Parsing	43
3.9	Debugging	43
3.10	Miscellaneous instructions	43
3.11	Commands	44
3.12	Special variables	44
3.12.1	RC - Return code	45
3.12.2	RESULT - Function result	45
3.12.3	SIGL - Source line number	45

4. REXX Instructions

4.1	Introduction	47
4.2	Instruction descriptions	48
4.2.1	ADDRESS - Set environment	48
4.2.2	ARG - Fetch argument	51
4.2.3	CALL - Invoke routine	52
4.2.4	DO - Define start of DO-block	54
4.2.5	DROP - Free variable	56
4.2.6	EXIT - Terminate exec	57
4.2.7	IF - Conditional execution	58
4.2.8	INTERPRET - Interpret statement	59
4.2.9	ITERATE - Terminate the current cycle in the DO-loop	60

4.2.10	LEAVE - Terminate DO-loop	61
4.2.11	NOP - No-operation	62
4.2.12	NUMERIC - Define numeric formats	62
4.2.13	OPTIONS - Pass special parameters to the language processor	63
4.2.14	PARSE - Assign data	64
4.2.15	PROCEDURE - Define internal procedure	66
4.2.16	PULL - Fetch data element from the head of the stack	67
4.2.17	PUSH - Set data element at the head of the stack	68
4.2.18	QUEUE - Set data element at the tail of the stack	68
4.2.19	RETURN - Return from routine	69
4.2.20	SAY - Display	69
4.2.21	SELECT - Select one condition from a series of conditions	70
4.2.22	SIGNAL - Enable (or disable) an exception condition, or cause control to be passed to a routine (or label)	71
4.2.23	TRACE - Set debugging options	74
4.2.24	UPPER - Transform lowercase characters into uppercase	76
4.2.25	= - Assign	76
4.2.26	Command	77
4.3	Examples	77
5. Exception Handling		
5.1	Introduction	80
5.2	Condition traps	81
5.2.1	ERROR condition	81
5.2.2	FAILURE condition	81
5.2.3	HALT condition	82
5.2.4	NOVALUE condition	82
5.2.5	SYNTAX condition	83
5.2.6	Stacked conditions	83
5.3	Explicit invocation	86
6. String Processing		
6.1	Introduction	87
6.2	Word parsing	88
6.3	Delimiter parsing	90
6.4	Positional parsing	91
6.5	Dynamic parsing	92
6.6	Argument parsing	92
6.7	Worked examples	93
7. Numbers and Arithmetic		
7.1	Introduction	95
7.2	Number	95
7.3	Arithmetic operators	97
7.3.1	Power (exponentiation)	97
7.3.2	Multiplication	97
7.3.3	Division	97
7.3.4	Modulo (integer) division	98
7.3.5	Remainder division	98
7.3.6	Numeric comparison	98

7.4	Precision and representation	99
7.4.1	Computational base	100
7.4.2	Retention of significance	100
7.4.3	Arithmetic exceptions	101
7.5	Sequence of operations	101
7.6	Numeric formatting	102

8. Input/Output

8.1	Introduction	103
8.2	Data buffering	104
8.3	Opening and closing the data files	104
8.4	Terminal operations	104
8.5	File-mode operations	105
8.5.1	File positioning	106
8.5.2	Write operation	106
8.5.3	Delete data operation	106
8.5.4	Read operation	107
8.5.5	Update operation	107

9. Debugging

9.1	Introduction	108
9.2	Exception conditions	109
9.3	Tracing options	110
9.3.1	ALL - Display all expressions before execution	110
9.3.2	COMMANDS - Display all commands before execution	111
9.3.3	ERROR - Display all commands which return an error condition	111
9.3.4	FAILURE - Display all commands which return a negative error condition	112
9.3.5	INTERMEDIATE - Display all expressions (with intermediate results) before being executed	112
9.3.6	LABELS - Display all labels as they are reached	113
9.3.7	RESULTS - Display all expressions (with end results) before being executed	113
9.3.8	SYNTAX - Check syntax without processing the statements	114
9.3.9	n - Suppress trace intervention	114
9.4	Trace output	116
9.4.1	Trace data prefixes	116
9.4.2	Trace intermediate data prefixes	116
9.4.3	Trace output example	117
9.5	Interactive debug	118
9.5.1	Interactive debugging example	119
9.6	Parametric debug	120
9.7	Errors with host commands and functions	120

10. REXX Built-in Functions

10.1	Introduction	122
10.1.1	Function-like (direct) invocation	123
10.1.2	Call invocation	123

10.1.3	Internal functions	124
10.1.4	Function side-effects	124
10.1.5	Invocation errors	124
10.1.6	Internal data format	125
10.2	Function definitions	125
10.2.1	ABBREV - Test whether string is an abbreviation	127
10.2.2	ABS - Return absolute value	127
10.2.3	ADDRESS - Return name of current environment	128
10.2.4	ARG - Return argument	128
10.2.5	BITAND - Logical And	129
10.2.6	BITOR - Logical Or	129
10.2.7	BITXOR - Logical Exclusive-Or	130
10.2.8	B2X - Convert binary data to hexadecimal	131
10.2.9	CENTRE (CENTER) - Centralise data	131
10.2.10	COMPARE - Compare	132
10.2.11	CONDITION - Return condition	132
10.2.12	COPIES - Duplicate data	133
10.2.13	C2D - Convert character data to decimal	133
10.2.14	C2X - Convert character data to hexadecimal	134
10.2.15	DATATYPE - Determine data type	135
10.2.16	DATE - Return current date	136
10.2.17	DELSTR - Delete substring	137
10.2.18	DELWORD - Delete words	138
10.2.19	DIGITS - Return the NUMERIC DIGITS setting	138
10.2.20	D2C - Convert decimal data to character	139
10.2.21	D2X - Convert hexadecimal data to character	139
10.2.22	ERRORTEXT - Return message text	140
10.2.23	FIND - Search for word	140
10.2.24	FORM - Determine NUMERIC FORM setting	141
10.2.25	FORMAT - Format numeric value	141
10.2.26	FUZZ - Determine NUMERIC FUZZ setting	142
10.2.27	INDEX - Search for substring	142
10.2.28	INSERT - Insert substring	142
10.2.29	JUSTIFY - Justify string of words	143
10.2.30	LASTPOS - Determine last position of phrase	144
10.2.31	LEFT - Left-align string	145
10.2.32	LENGTH - Determine length of string	145
10.2.33	LINESIZE - Return the (maximum) width of a terminal line ...	146
10.2.34	MAX - Determine the maximum of a series of numeric values	146
10.2.35	MIN - Determine the minimum of a series of numeric values	146
10.2.36	OVERLAY - Overlay part of a string with a phrase	147
10.2.37	POS - Search for substring	148
10.2.38	QUEUED - Determine the number of entries in the queue	148
10.2.39	RANDOM - Generate a (pseudo-)random number	149
10.2.40	REVERSE - Reverse the sequence of data	150
10.2.41	RIGHT - Right-align string	150
10.2.42	SIGN - Determine numeric sign	151
10.2.43	SOURCELINE - Return "program line"	151

10.2.44	SPACE - Insert fill-character between words	152
10.2.45	STRIP - Remove padding-characters at the start or end of a string	152
10.2.46	SUBSTR - Extract substring	153
10.2.47	SUBWORD - Extract series of words from word-string	154
10.2.48	SYMBOL - Determine the status of a symbol	155
10.2.49	TIME - Return the current time-of-day	155
10.2.50	TRACE - Return (and set) current trace mode	157
10.2.51	TRANSLATE - Translate	159
10.2.52	TRUNC - Truncate numeric value	160
10.2.53	USERID - Return Userid	161
10.2.54	VALUE - Return the content of a symbol	161
10.2.55	VERIFY - Test whether only characters in a phrase are present in string	162
10.2.56	WORD - Fetch word	163
10.2.57	WORDINDEX - Determine the character position of a word in a string of words	163
10.2.58	WORDLENGTH - Determine word length	164
10.2.59	WORDPOS - Determine word-number of word in word-string	164
10.2.60	WORDS - Determine number of words in word-string	165
10.2.61	XRANGE - Define a range of hexadecimal values	165
10.2.62	X2B - Convert hexadecimal to binary	166
10.2.63	X2C - Convert hexadecimal to character	166
10.2.64	X2D - Convert hexadecimal to decimal	167
10.3	Examples	167

11. Host REXX Commands

11.1	Introduction	170
11.2	Host REXX command definitions	170
11.2.1	DELSTACK - Delete stack	171
11.2.2	DROPBUF - Release buffer	171
11.2.3	EXECIO - Perform input/output operation	171
11.2.4	EXECUTIL - Specify execution environment for REXX program	174
11.2.5	HE - Halt execution	175
11.2.6	HI - Halt interpretation	176
11.2.7	HT - Halt typing	176
11.2.8	MAKEBUF - Create new buffer in the stack	176
11.2.9	NEWSTACK - Create a new stack	176
11.2.10	QBUF - Query buffer	177
11.2.11	QELEM - Query elements	177
11.2.12	QSTACK - Query stack	177
11.2.13	RT - Resume typing	178
11.2.14	SUBCOM - Confirm the host environment	178
11.2.15	TE - Trace end	178
11.2.16	TS - Trace start	178
11.3	Example	179

12. MVS Command Functions

12.1	Introduction	180
------	--------------------	-----

12.2	MVS command function calls.....	180
12.2.1	GETMSG - Retrieve a console session message.....	182
12.2.2	LISTDSI - List (obtain) dataset information	184
12.2.3	MSG - Set (interrogate) CLIST CONTROL MSG option.....	186
12.2.4	OUTTRAP - Trap TSO display output	187
12.2.5	PROMPT - Set (interrogate) CLIST CONTROL PROMPT option.....	189
12.2.6	SETLANG - Set language	189
12.2.7	STORAGE - Set (interrogate) main storage content.....	190
12.2.8	SYSDSN - Request dataset status	191
12.2.9	SYSVAR - Fetch TSO system variable	192
12.3	Invocation of other TSO commands	193
12.4	Example.....	194
 13. REXX Invocation		
13.1	Introduction.....	195
13.2	Invocation from TSO/ISPF	195
13.3	Invocation from batch TSO	197
13.4	Invocation from MVS	198
13.5	Linkage to the host (MVS-TSO) environment.....	199
13.6	Linkage to programs	199
13.6.1	LINK (ATTACH).....	200
13.6.2	LINKMVS (ATTACHMVS)	201
13.6.3	LINKPGM (ATTACHPGM)	202
13.7	Interface with ISPEXEC (ISPF Dialog Manager).....	203
13.8	Interface with ISREDIT (ISPF/PDF Edit macro).....	204
13.8.1	Macro example.....	204
13.9	Interface with DB2 (Database2)	204
13.9.1	DB2 invocation example.....	205
13.10	Interface with QMF (Query Management Facility).....	205
13.10.1	DSQCIX - QMF Callable Interface.....	205
13.10.2	DSQCCI - QMF Command Interface	207
13.11	Interface from programs with REXX	208
 14. System Interfaces		
14.1	Introduction.....	209
14.2	General conditions.....	210
14.2.1	C/370 restrictions	211
14.2.2	Other program language restrictions.....	211
14.3	Invocation of a REXX exec.....	212
14.3.1	Interface from programs to batch REXX (IRXJCL service)	212
14.3.2	Invocation of a REXX exec using the TSO Service Facility (IJKFTSR service)	212
14.4	Interface from programs to REXX processor (IRXEXEC)	215
14.4.1	Sample program.....	218
14.4.2	Invocation of an in-store exec	219
14.5	Program access to REXX variables (IRXEXCOM service)	222
14.5.1	Sample programs.....	224
14.5.2	Sample retrieval program.....	228
14.6	Stack processing (IRXSTK service).....	229
14.6.1	Sample programs.....	230

14.7	Command interface	233
14.7.1	Entry conditions	233
14.7.2	Parameter list	234
14.7.3	Exit conditions	235
14.7.4	Examples	235
14.7.5	Command registration interface	238
14.7.6	Command router	240
14.8	Function interface	241
14.8.1	Entry conditions	241
14.8.2	Exit conditions	241
14.8.3	Sample function program	241
14.9	Function package	244
14.9.1	Function directory	245
14.10	Load routine - IRXLOAD service	247
14.11	Initialisation routine - IRXINIT service	249
14.11.1	Sample program	250
14.12	Get result - IRXRLT service	251
14.12.1	Sample program	252
14.13	Control blocks	253
14.13.1	Argument List	253
14.13.2	EFPL (External Function Parameter List)	253
14.13.3	ENVBLOCK (Environment Block)	254
14.13.4	EVALBLOCK (Evaluation Block)	255
14.13.5	EXECBLK (Exec Block)	256
14.13.6	INSTBLK (In-storage Control Block)	257
14.13.7	SHVBLOCK (Shared Variable (Request) Block)	258
14.13.8	VEEP (Vector of External Entry Points)	260
14.14	Examples	262
14.14.1	Assembler implementation	262
14.14.2	C/370 implementation	263
14.14.3	COBOL implementation	267
14.14.4	PL/I implementation	269
 15. Worked Examples		
15.1	Introduction	271
15.2	Worked example 1	272
15.3	Worked example 2	275
15.4	Worked example 3	277
15.5	Worked example 4	284
15.6	Worked example 5	290
 Appendix A. Syntax Notation		
Appendix B. Glossary		
Index		