

UNI-REXX: REXX FOR THE UNIX & VMS ENVIRONMENTS

3RD ANNUAL REXX SYMPOSIUM
ANNAPOLIS, MARYLAND
MAY 5, 1992

ED SPIRE
THE WORKSTATION GROUP
ROSEMONT, ILLINOIS

PRESENTATION OUTLINE

PART 1. "MARKETING" REXX FOR UNIX

THE UNIX MARKET
MIS AND UNIX
MACRO LANGUAGES FOR UNIX
OUR EXPERIENCE SO FAR
PERL
UNI-REXX
RECENT WORK ON UNI-REXX
PLANS TO IMPROVE ACCEPTANCE
OTHER PLANS

PART 2. OTHER REXX RELATED ITEMS

COMBINED REXX AND C DEVELOPMENT PROJECT
RELATIONSHIP TO PUBLIC DOMAIN REXX'S

"MARKETING" REXX

- ... NOT IN THE "GRUBBY" SENSE OF "HAWKING" REXX
 - ... BUT IN THE SENSE OF "MAKING REXX VALUABLE", THROUGH
 - IMPROVEMENT,
 - APPLICATION,
 - EDUCATION,
 - ETC.
-

THE UNIX COMMUNITY

ORIGINALLY ACADEMIC AND ENGINEERING ORIENTED USAGE

FIRST COMMERCIALIZATION INVOLVED SMALL BUSINESS APPLICATIONS
(I.E., XENIX)

RISC PRICE/PERFORMANCE IMPROVEMENTS ARE ATTRACTING TRADITIONAL
COMMERCIAL MIS APPLICATIONS

LATELY, THE INTEL PRICE/PERFORMANCE CURVE IS APPROACHING THAT OF THE
LOW END RISC SYSTEMS, FURTHER ACCELERATING WIDER INTEREST IN UNIX.

TRADITIONAL COMMERCIAL MIS ORGANIZATIONS AND UNIX

RISC PRICE/PERFORMANCE HAS BECOME UNAVOIDABLY ATTRACTIVE

UNIX IS THE ONLY CURRENTLY AVAILABLE OS FOR THESE PLATFORMS.
(OS/2 AND WINDOWS/NT WILL CHANGE THIS SITUATION)

UNIX MAY CONTINUE TO BE THE ONLY TRULY PORTABLE ENVIRONMENT
FOR THOSE WHO SEEK THE FLEXIBILITY OF OPEN SYSTEMS.
(OS/2 AND WINDOWS/NT ARE PROPRIETARY TECHNOLOGY)

THE UNIX LEARNING CURVE IS LARGE FOR EXISTING COMMERCIAL MIS STAFFERS

THESE FACTORS ARE THE BASIS FOR TWG'S PRODUCT LINE OF MAINFRAME UTILITY
SOFTWARE FOR UNIX.

MACRO LANGUAGES FOR UNIX

UNIX INCLUDES MANY "STANDARD" UTILITIES:

- A LARGE NUMBER OF COMMANDS THAT PROVIDE INFORMATION, ACCESS, AND CONTROL AT A VERY LOW LEVEL
- REUSABLE "FILTERS" USED VIA "PIPES"
- I/O REDIRECTION

... MAKING FOR A VERY FLEXIBLE (ALBETT DAUNTING) ENVIRONMENT.

GIVEN THIS LEVEL OF COMPLEXITY, THERE IS CERTAINLY A NEED FOR MACRO FACILITIES. AS AN EXAMPLE, TAKE THE CASE OF ROUTINE DISK SPACE SPACE MANAGEMENT AT THE END-USER LEVEL.

STANDARD UNIX REALLY ONLY PROVIDES THE 'DU' COMMAND, WHICH PROVIDES VERY LOW LEVEL DATA...

(OUTPUT OF UNIX COMMAND "DU /USR/EXPORT/HOME/ETS" FOLLOWS...)

```
1668 /USR/EXPORT/HOME/ETS/PDR/REXX1
56 /USR/EXPORT/HOME/ETS/PDR/REXX2/TRIP
5 /USR/EXPORT/HOME/ETS/PDR/REXX2/CODE
1316 /USR/EXPORT/HOME/ETS/PDR/REXX2
3018 /USR/EXPORT/HOME/ETS/PDR
197 /USR/EXPORT/HOME/ETS/UTIL
5 /USR/EXPORT/HOME/ETS/LOCALTERM/A
2 /USR/EXPORT/HOME/ETS/LOCALTERM/D
9 /USR/EXPORT/HOME/ETS/LOCALTERM/H
69 /USR/EXPORT/HOME/ETS/LOCALTERM/I
5 /USR/EXPORT/HOME/ETS/LOCALTERM/J
3 /USR/EXPORT/HOME/ETS/LOCALTERM/S
2 /USR/EXPORT/HOME/ETS/LOCALTERM/U
5 /USR/EXPORT/HOME/ETS/LOCALTERM/V
13 /USR/EXPORT/HOME/ETS/LOCALTERM/W
114 /USR/EXPORT/HOME/ETS/LOCALTERM
1 /USR/EXPORT/HOME/ETS/.WASTEBASKET
432 /USR/EXPORT/HOME/ETS/REXX
42 /USR/EXPORT/HOME/ETS/XEDIT
24 /USR/EXPORT/HOME/ETS/TERMINFO
3 /USR/EXPORT/HOME/ETS/RXF
133 /USR/EXPORT/HOME/ETS/LANG/SC1.0/MAN/MAN1
398 /USR/EXPORT/HOME/ETS/LANG/SC1.0/MAN/MAN3
4 /USR/EXPORT/HOME/ETS/LANG/SC1.0/MAN/MAN5
541 /USR/EXPORT/HOME/ETS/LANG/SC1.0/MAN
9 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/ARPA
9 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/DEBUG
14 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/HSFS
3 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/LOFS
12 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/LWP
29 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/MON
29 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/NET
```

59 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/NETINET
9 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/NETTLI
15 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/NFS
158 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/PIXRECT
13 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/PROTOCOLS
28 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/RFS
56 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/RPC
58 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/RPCSVCS
14 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/SBUSDEV
20 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/SUN
9 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/SUN3
3 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/SUN3X
3 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/SUN4
3 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/SUN4C
59 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/SUNDEV
153 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/SUNTOOL
142 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/SUNWINDOW
210 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/SYS
3 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/TFS
6 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/UFS
3 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC/VM
1430 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE/CC
1436 /USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE
1001 /USR/EXPORT/HOME/ETS/LANG/SC1.0/CG87
1003 /USR/EXPORT/HOME/ETS/LANG/SC1.0/CG89
12 /USR/EXPORT/HOME/ETS/LANG/SC1.0/MISALIGN
52 /USR/EXPORT/HOME/ETS/LANG/SC1.0/README
9793 /USR/EXPORT/HOME/ETS/LANG/SC1.0
133 /USR/EXPORT/HOME/ETS/LANG/MAN/MAN1
398 /USR/EXPORT/HOME/ETS/LANG/MAN/MAN3
4 /USR/EXPORT/HOME/ETS/LANG/MAN/MAN5
541 /USR/EXPORT/HOME/ETS/LANG/MAN
12537 /USR/EXPORT/HOME/ETS/LANG
219 /USR/EXPORT/HOME/ETS/DOC
49 /USR/EXPORT/HOME/ETS/SYMP
16725 /USR/EXPORT/HOME/ETS

MACRO LANGUAGES FOR UNIX

A TYPICAL MACRO PROGRAM WOULD PROVIDE THIS LOW LEVEL INFORMATION IN A MORE MANAGEABLE FORM...

(OUTPUT OF UNIX COMMAND "SPACE /USR/EXPORT/HOME/ETS" FOLLOWS...)

FILESYSTEM	KBYTES	USED	AVAIL	CAPACITY	MOUNTED ON
/DEV/sd0g	186414	141068	26704	84%	/USR
NUM	SIZE(K)	% NODE	%FILESYS	ITEM	
1	3018	18.041	2.139	/USR/EXPORT/HOME/ETS/PDR	
2	197	1.178	0.140	/USR/EXPORT/HOME/ETS/UTIL	
3	114	0.681	0.081	/USR/EXPORT/HOME/ETS/LOCALTERM	
4	1	0.006	0.001	/USR/EXPORT/HOME/ETS/.WASTEBASKET	
5	432	2.582	0.306	/USR/EXPORT/HOME/ETS/REXX	
6	42	0.251	0.030	/USR/EXPORT/HOME/ETS/XEDIT	
7	24	0.143	0.017	/USR/EXPORT/HOME/ETS/TERMINFO	
8	3	0.018	0.002	/USR/EXPORT/HOME/ETS/RXF	
9	12537	74.942	8.887	/USR/EXPORT/HOME/ETS/LANG	
10	219	1.309	0.155	/USR/EXPORT/HOME/ETS/DOC	
11	53	0.317	0.038	/USR/EXPORT/HOME/ETS/SYMP	
12	16729	100.000	11.859	/USR/EXPORT/HOME/ETS	

SELECT NODE NUMBER FOR REDISPLAY (* FOR SAME) AND OPTIONAL LEVEL OR 'X' TO EXIT

1

FILESYSTEM	KBYTES	USED	AVAIL	CAPACITY	MOUNTED ON
/DEV/sd0g	186414	141068	26704	84%	/USR
NUM	SIZE(K)	% NODE	%FILESYS	ITEM	
1	1668	55.268	1.182	/USR/EXPORT/HOME/ETS/PDR/REXX1	
2	1316	43.605	0.933	/USR/EXPORT/HOME/ETS/PDR/REXX2	
3	3018	100.000	2.139	/USR/EXPORT/HOME/ETS/PDR	

SELECT NODE NUMBER FOR REDISPLAY (* FOR SAME) AND OPTIONAL LEVEL OR '0' TO GO BACK UP ONE NODE OR 'X' TO EXIT

0

FILESYSTEM	KBYTES	USED	AVAIL	CAPACITY	MOUNTED ON
/DEV/SD0G	186414	141069	26703	84%	/USR
NUM	SIZE(K)	% NODE	%FILESYS	ITEM	
1	3018	18.041	2.139	/USR/EXPORT/HOME/ETS/PDR	
2	197	1.178	0.140	/USR/EXPORT/HOME/ETS/UTIL	
3	114	0.681	0.081	/USR/EXPORT/HOME/ETS/LOCALTERM	
4	1	0.006	0.001	/USR/EXPORT/HOME/ETS/.WASTEBASKET	
5	432	2.582	0.306	/USR/EXPORT/HOME/ETS/REXX	
6	42	0.251	0.030	/USR/EXPORT/HOME/ETS/XEDIT	
7	24	0.143	0.017	/USR/EXPORT/HOME/ETS/TERMINFO	
8	3	0.018	0.002	/USR/EXPORT/HOME/ETS/RXF	
9	12537	74.942	8.887	/USR/EXPORT/HOME/ETS/LANG	
10	219	1.309	0.155	/USR/EXPORT/HOME/ETS/DOC	
11	53	0.317	0.038	/USR/EXPORT/HOME/ETS/SYMP	
12	16729	100.000	11.859	/USR/EXPORT/HOME/ETS	

SELECT NODE NUMBER FOR REDISPLAY (* FOR SAME) AND OPTIONAL LEVEL
OR 'X' TO EXIT

9

FILESYSTEM	KBYTES	USED	AVAIL	CAPACITY	MOUNTED ON
/DEV/SD0G	186414	141070	26702	84%	/USR
NUM	SIZE(K)	% NODE	%FILESYS	ITEM	
1	9793	78.113	6.942	/USR/EXPORT/HOME/ETS/LANG/SC1.0	
2	541	4.315	0.384	/USR/EXPORT/HOME/ETS/LANG/MAN	
3	12537	100.000	8.887	/USR/EXPORT/HOME/ETS/LANG	

SELECT NODE NUMBER FOR REDISPLAY (* FOR SAME) AND OPTIONAL LEVEL
OR '0' TO GO BACK UP ONE NODE
OR 'X' TO EXIT

* 2

FILESYSTEM	KBYTES	USED	AVAIL	CAPACITY	MOUNTED ON
/DEV/SD0G	186414	141070	26702	84%	/USR
NUM	SIZE(K)	% NODE	%FILESYS	ITEM	
1	541	4.315	0.384	/USR/EXPORT/HOME/ETS/LANG/SC1.0/MAN	
2	1436	11.454	1.018	/USR/EXPORT/HOME/ETS/LANG/SC1.0/INCLUDE	
3	1001	7.984	0.710	/USR/EXPORT/HOME/ETS/LANG/SC1.0/CG87	
4	1003	8.000	0.711	/USR/EXPORT/HOME/ETS/LANG/SC1.0/CG89	
5	12	0.096	0.009	/USR/EXPORT/HOME/ETS/LANG/SC1.0/MISALIGN	
6	52	0.415	0.037	/USR/EXPORT/HOME/ETS/LANG/SC1.0/README	
7	9793	78.113	6.942	/USR/EXPORT/HOME/ETS/LANG/SC1.0	
8	133	1.061	0.094	/USR/EXPORT/HOME/ETS/LANG/MAN/MAN1	
9	398	3.175	0.282	/USR/EXPORT/HOME/ETS/LANG/MAN/MAN3	
10	4	0.032	0.003	/USR/EXPORT/HOME/ETS/LANG/MAN/MAN5	
11	541	4.315	0.384	/USR/EXPORT/HOME/ETS/LANG/MAN	
12	12537	100.000	8.887	/USR/EXPORT/HOME/ETS/LANG	

SELECT NODE NUMBER FOR REDISPLAY (* FOR SAME) AND OPTIONAL LEVEL
OR '0' TO GO BACK UP ONE NODE
OR 'X' TO EXIT

X

MACRO LANGUAGES FOR UNIX

PERL IS FAST BECOMING THE DE FACTO STANDARD MACRO LANGUAGE FOR UNIX, EVEN THOUGH REXX IS CLEARLY SUPERIOR IN MANY WAYS...

- REXX HAS BETTER PROGRAMMING STRUCTURES
- STRINGS AND ARRAYS START AT 0, NOT 1
- NO WAY TO SPECIFY THE DEFAULT VALUE OF AN ASSOCIATIVE ARRAY (STEM)
- RELATIONSHIP BETWEEN OPERATORS, FUNCTIONS, AND PRECEDENCE IS VERY CONFUSING. FOR EXAMPLE:

```
PRINT (1+1) +1
```

PRINTS "2" AND THROWS "3" AWAY AS AN UNUSED SIDE EFFECT!

- SEMICOLONS REQUIRED (JUST LIKE C)
- ARRAYS, LISTS, AND SCALARS BEGIN WITH SPECIAL CHARACTERS (@,\$,...)
- BRACKETS REQUIRED IN COMPOUND INSTRUCTIONS, I.E.,

```
WHILE
  * STMT;          NOT      WHILE
  *;              STMT;
```

- THE SYNTAX HAS FAR TOO MANY SPECIAL CHARACTERS AND IDEOSYNCRACIES.

MACRO LANGUAGES FOR UNIX

REXX-PERL COMPARISON 1

```
/*  
 * BRING IN PARMS, HANDLE DEFAULTS  
 */  
PARSE ARG NODE LEVEL TRACEOPT  
TRACE VALUE TRACEOPT  
IF NODE="" * NODE="."  
THEN DO  
  CALL POPEN 'PWD'  
  PARSE PULL NODE  
  END  
IF LEVEL=""  
THEN LEVEL=1
```

```
#  
# BRING IN PARMS, HANDLE DEFAULTS  
#  
$NODE = SHIFT @ARGV;  
CHOP($NODE = WD  
  IF ($NODE EQ '.' * $NODE EQ ''));  
$LEVEL = SHIFT @ARGV;  
$LEVEL = 1  
  IF ($LEVEL EQ '');
```


MACRO LANGUAGES FOR UNIX

REXX-PERL COMPARISON 2

```
/*
 * FIND APPLICABLE FILE SYSTEM AND SIZE
 */
CALL POPEN 'DF'
DO WHILE QUEUED(>0
  PARSE PULL L
  TEST=WORD(L,DFNAMEWORD.UNAME)
  IF TEST="/"
  THEN DO
    FSNAME=TEST
    FSSIZE=WORD(L,DFSPACEWORD.UNAME)
    IF DFFREWORD.UNAME<>0
    THEN FSSIZE=FSSIZE-WORD(L,DFFREWORD.UNAME)
  END
  ELSE IF LEFT(NODE,LENGTH(TEST))=TEST
  THEN DO
    FSNAME=TEST
    FSSIZE=WORD(L,DFSPACEWORD.UNAME)
    IF DFFREWORD.UNAME<>0
    THEN FSSIZE=FSSIZE-WORD(L,DFFREWORD.UNAME)
    DO WHILE QUEUED(>0
      PARSE PULL I.
    END
  LEAVE
  END
END
IF DFBLOCKS.UNAME THEN FSSIZE=FSSIZE*BLOCKSIZE.UNAME
```

MACRO LANGUAGES FOR UNIX

REXX-PERL COMPARISON 2

```
#
# FIND APPLICABLE FILE SYSTEM AND SIZE
#
OPEN (DF, 'DF *');
WHILE (<DF>)
*
  CHOP;
  @DF = SPLIT;
  $TEST = @DF@DFNAMEWORD*$UNAME*-11;
  IF ($TEST EQ '/')
  *
    $FSNAME = '/';
    $FSSIZE = @DF@DFSPACEWORD*$UNAME*-11;
    $FSSIZE = $FSSIZE - @DF@DFFREWORD*$UNAME*-11
      IF (@DF@DFFREWORD*$UNAME*-11 != 0);
  *
  ELSE
  *
    IF ($NODE = ° /-$TEST/)
    *
      $FSNAME = $TEST;
      $FSSIZE = @DF@DFSPACEWORD*$UNAME*-11;
      $FSSIZE = $FSSIZE - @DF@DFFREWORD*$UNAME*-11
        IF (@DF@DFFREWORD*$UNAME*-11 != 0);
      LAST;
    *
  *
*
CLOSE(<DF>);
$FSSIZE = $FSSIZE * $BLOCKSIZE*$UNAME;
IF ($DFBLOCKS*$UNAME):
```

MACRO LANGUAGES FOR UNIX

REXX-PERL COMPARISON 3

```
/*  
 * DO A DU, BUFFER UP THE LINES, AND GET THE NODE TOTAL  
 */  
CALL POPEN 'DU' NODE  
DO LINE=1 WHILE QUEUED(>)>0  
  PARSE PULL L  
  PARSE VAR L COUNT NAME  
  NODECOUNTS.NAME=COUNT  
  LINES.LINE=L  
  END  
LINE=LINE-1  
NODECOUNT=WORD(LINES.LINE,1)
```

```
#  
# DO A DU, BUFFER UP THE LINES, AND GET THE NODE TOTAL  
#  
OPEN(DU, "DU $NODE #"):  
$LINE = 1;  
WHILE (<DU>)  
*  
  CHOP;  
  @DU = SPLIT;  
  ($COUNT, $NAME) = @DU%0..11;  
  $NODECOUNTS*$NAME* = $COUNT;  
  $LINES%$LINE1 = $_  
  ++$LINE;  
*  
--$LINE;  
$NODECOUNT = SHIFT(@DU);  
CLOSE(<DU>);
```

OUR EXPERIENCE SO FAR

AS OF THIS TIME

APPLICATIONS THAT EMBED REXX ARE WELL RECEIVED BY COMMERCIAL MIS
TRANSITING TO UNIX

- XEDIT

- ISPF

...REXX USAGE WITHIN THESE SPECIFIC ENVIRONMENTS IS HIGHLY PORTABLE

HOWEVER...

REXX HAS NOT BEEN WELL RECEIVED AS A MACRO FACILITY FOR UNIX.

- WHY

- WHAT ARE WE GOING TO DO ABOUT IT

PERL

- FAMILIAR TO CURRENT UNIX SYSTEM ADMINISTRATORS, WHO ARE ALSO SOFTWARE SELECTORS.
 - REGULAR EXPRESSION SUPPORT
 - MANY UNIX SPECIFIC FUNCTIONS BUILT-IN.
 - EASE OF IMPLEMENTATION
 - PORTABILITY
 - FREE, IN SOURCE FORM
 - LEARNING CURVE IS AS STEEP (OR STEEPER) THAN THE TYPICAL UNIX ALTERNATIVES.
 - QUESTIONABLE SUPPORT
-

UNI-REXX

- FAMILIAR TO COMMERCIAL MIS, UNFAMILIAR TO UNIX SYSTEM ADMINISTRATORS
- LITTLE SPECIFIC UNIX SUPPORT:
 - NO REGULAR EXPRESSION SUPPORT
 - VERY FEW UNIX-RELATED FUNCTIONS
- COMMERCIAL PRODUCT, WHEREAS MOST REXX LANGUAGE PROCESSORS ARE BUNDLED INTO THE UNDERLYING OS.

RECENT WORK ON UNI-REXX

THE RECENT IMPROVEMENTS TO UNI-REXX HAVE "PLAYED TO IT'S STRENGTHS", IMPROVING IT'S USEFULNESS FOR EMBEDDED APPLICATIONS

- ADDITIONAL APIS
 - COMPLETION OF STANDARD REXX FACILITIES
 - IMPROVED PERFORMANCE
-

RECENT WORK ON UNI-REXX

ADDITIONAL API'S

- SYMBOL TABLE ACCESS EXITS
 - READ, WRITE, AND DROP
 - TAKEN ONLY WHEN AN UNINITIALIZED VARIABLE IS REFERENCED
 - EXIT CAN SUPPLY/ACCEPT A VALUE OR ALLOW DEFAULT PROCESSING
 - SUPPLIED/ACCEPTED VALUES MAKE NO REXX SYMBOL TABLE ENTRY
 - COMPILER EXITS
 - SUPPORT LANGUAGE EXTENSIONS BY EMBEDDED APPLICATION
 - COULD SUPPORT A PREPROCESSOR
 - MULTI-THREADING SUPPORT
-

RECENT WORK ON UNI-REXX

COMPLETION OF STANDARD REXX FACILITIES

- FULL REXX MATH
- LANGUAGE LEVEL 4.00 FEATURES

NOTE THAT THIS WORK HAS NOT IMPROVED REXX'S POSITION WITH RESPECT TO UNIX

PLANS TO IMPROVE ACCEPTANCE OF REXX FOR UNIX

LOTS MORE BUILT-IN FUNCTIONS

- UNIX SPECIFIC, ALA PERL ETC. (EXPOSE ENTIRE STANDARD C LIBRARY?)
 - REGULAR EXPRESSION SUPPORT
 - PROCESS MANAGEMENT & COMMUNICATION
 - USER INTERFACES: CURSES FOR SURE, POSSIBLY X AS WELL
 - DATABASE ACCESS
-

PLANS TO IMPROVE ACCEPTANCE OF REXX FOR UNIX

PROCESS MANAGEMENT & COMMUNICATION

WE ARE CURRENTLY EXPERIMENTING WITH FUNCTIONS THAT WILL ALLOW REXX TO CONTROL ONE OR MORE ASYNCHRONOUS PROCESSES VIA STDIN AND STDOUT.

HANDLE=PIPE(COMMAND) - INITIATES AN ASYNCHRONOUS PROCESS, WITH BOTH STDIN AND STDOUT PIPED BACK TO THE PARENT.

RC=PIPEIN(HANDLE,DATA) - RECEIVES RAW DATA FROM A PROCESS'S STDOUT

RC=PIPEOUT(HANDLE,DATA) - SENDS RAW DATA TO A PROCESS'S STDIN

RC=PIPESEL(HANDLE1,HANDLE2,HANDLE3) - BLOCKS UNTIL DATA IS AVAILABLE ON A CONTROLLED PROCESS'S STDOUT, OR ON THE PARENT TASK'S STDIN.

"RAW" DATA CAN BE A BIT CUMBERSOME (CONTROL CHARACTERS ARE PRESENT,) BUT THIS APPROACH ALLOWS FULL CONTROL OF ANY APPLICATION WITHOUT IT'S INCLUSION OF A "REXX MESSAGE PORT" ETC.

CURRENT IMPLEMENTATION USES SOCKETS. PRODUCTION QUALITY WILL REQUIRE PSEUDO-TTY'S INSTEAD.

PLANS TO IMPROVE ACCEPTANCE OF REXX FOR UNIX
BASE TECHNOLOGY FREE TO ACADEMIA

- EXECUTABLE ONLY
 - NO SOURCE LEVEL PORTABILITY
 - NO EMBEDDED USE (GIVEN CURRENT TECHNOLOGY)
 - NO SUPPORT, VERY LIMITED DOCUMENTATION
-

OTHER PLANS

IMPROVED PROGRAM DOCUMENTATION (I.E., PROGRAM LISTING FACILITIES)

IPC FOR THE REXX API'S

- EXPOSE APIs TO INVOKED UNIX COMMANDS
- EXTERNALIZE EMBEDDED LANGUAGE PROCESSING

REXX SHELL

- BETTER INTEGRATE THE EXTERNAL DATA QUEUE & UNIX COMMAND PROCESSING
- ALLOW REXX CONTROL OVER THE PERSISTENT SHELL ENVIRONMENT

ADDITIONAL EMBEDDED APPLICATIONS

LOTS OF WORK IN THE ABOVE PLANS, MORE THAN WE CAN FUND IN THE NEAR TERM.
EFFECTIVENESS OF THESE EFFORTS AT INCREASING ACCEPTANCE IS UNSURE.
PRIORITIZATION WILL BE AN ISSUE, COMMENTS ARE WELCOME.

OTHER ITEMS OF INTEREST FROM TWG

COMBINED REXX AND C DEVELOPMENT PROJECT FOR UNI-SPF
RELATIONSHIP WITH PUBLIC DOMAIN REXX IMPLEMENTATIONS

COMBINED REXX AND C DEVELOPMENT PROJECT FOR UNI-SPF

GOAL - BUILDING UPON OUR BASE OF UNI-REXX AND UNI-XEDIT,
ADD DIALOG MANAGEMENT FACILITIES, "PDF", AND THE SPF EDITOR.

WORKS BEGINS ON DIALOG MANAGEMENT (IN "C") IN APRIL, 1991.

PROTOTYPE DIALOG MANAGEMENT AVAILABLE 3Q91.

INTENSIVE PERIOD OF WORK DURING 4Q91:

-- "PDF" CREATED USING DIALOG MANAGEMENT & REXX

-- XEDIT TURNED INTO THE SPF EDITOR WITH REXX MACROS AND SUPPORTING
"C" CODE AS REQUIRED.

PRODUCTION RELEASE 1Q92.

COMBINED REXX AND C DEVELOPMENT PROJECT FOR UNI-SPF

SPF PROJECT CONSUMED 12 MAN-MONTHS OF C CODING
4 MAN-MONTHS OF REXX CODING

PRODUCED 829K BYTES OF C CODE
612K BYTES OF REXX CODE

HIGHLY PRODUCTIVE REXX PROGRAMMING ENVIRONMENT WAS INVALUABLE TO
RAPID PROTOTYPING AND QUICK DELIVERY.

CERTAIN REXX CODE SEGMENTS EXHIBIT PERFORMANCE PROBLEMS
(MITIGATED BY RISC PRICE/PERFORMANCE CHARACTERISTICS.)

MOST REXX CODE WILL PROBABLY BE RE-WRITTEN IN C EVENTUALLY.

RELATIONSHIP WITH PUBLIC DOMAIN REXX IMPLEMENTATIONS

ADDITIONAL IMPROVEMENTS ARE EXPECTED IN THE ACCEPTANCE OF REXX FOR UNIX

COMMERCIAL IMPLEMENTATION WILL HAVE IT'S ADVANTAGES:

- COMPLETENESS
- STABILITY
- PERFORMANCE
- DOCUMENTATION
- SUPPORT

TWG & IX WILL COOPERATE IN ESTABLISHING STANDARDS WHERE POSSIBLE

- API INTERFACE DEFINITIONS
- BUILT-IN FUNCTION DEFINITIONS

ANY OTHER POSITION WOULD BE FUTILE, ANYWAY!

COMMENTS, PLEASE.

THANKS!