

Rexx Language Association 2008 Symposium

An Experimental INCLUDE facility for Rexx

Rationale, Objectives, Development and Results

Les Koehler
3 May 2008

Table of Contents

Abstract

Rationale

Objectives

Development - Skeleton

Development - PARSE ARGS

Development - INCLUDE

Development - INCLUDE LOGIC

Development - SYNTAX

Development - Example of Syntax trap

Results - Sizes

Results - Added

Results - Includes

Results - Performance

Summary and Conclusion

Abstract

I will present an experimental INCLUDE facility intended to increase code reuse and improve maintainability.

Although written for ooRexx, the code can easily be adapted to any Rexx interpreter that implements the ANSI standard.

The presentation will include the code itself, execution measurements before and after as well as size comparisons.

Rationale

The Rexx Compiler for the mainframe provides /*%INCLUDE to allow the programmer to readily reuse existing code.

An examination of the four Excel Automation programs presented last year revealed that they used many blocks of code that were identical for each program.

These repetitive blocks of code tend to overwhelm the unique function of the code being written and make reusability impossible.

Furthermore, if a better technique was discovered while writing a later program, that code had to be manually copied to the other programs, which is burdensome and error prone.

Objectives

- **No preprocessing required**
- **Minimum overhead**
- **No special file extension required**
- **Recursion allowed**
- **Integrate with error trapping**
- **Allow arguments to be passed**
- **Optional tracking of included files**
- **All variables exposed**
- **KISS**

Development - Skeleton

```
Arg monyy '('opts
/* Opts: Flagname, keyword, min abbrev, default, text */
opt.1= 'PRINT?   PRINT    1 0 Print the worksheet'
opt.2= 'QUIT?    QUIT     1 0 Quit Excel worksheet'
opt.3= 'TIME?    TIME     1 1 Calculate elapsed time'
opt.4= 'KILLQ?   KILLQ    1 0 Kill Excel quietly'
opt.5= 'VISIBLE? VISIBLE  1 1 Make worksheet visible'
opt.6= 'INCLUDES? INCLUDES 1 0 Show Includes'
opt.7= 'PDF?     PDF      2 1 Convert PDF to PRT'
opt.0=7
address cmd
config_file='c:\WaterMillVillageReportsConfig.txt' /* User specified paths */
props = .Properties~load(config_file)             /* Get the file */
file_path = props~getProperty("prt_file_path")    /* Get our paths */
!incpath = props~getProperty("!incpath")
parse value exception1() with report_exception   /* Get SYNTAX handler */
signal on any name protectall                    /* Trap EVERYTHING */
call parse_args                                 /* Sets ERR? MON YY and flags */
call include 'ExcelSetup'
/* Main code goes here */
call include 'autofit'
```

```
call include_logic 'justify_and_draw_boxes','B:D','#,##0.00'  
call include_logic 'filter_and_insert_headers'  
call include_logic 'SaveAndSetupPrint',.7,.5,.5,.5,0,0  
call include_logic 'finish_up'  
exit
```

Development - PARSE_ARGS

```
PARSE_ARGS:  
exposes='REPORT_EXCEPTION !INCTRAK. !INCFEILE. !INCIX ,  
  !INCLUDES? !INCLINE.',  
  '!INCLINES. UNKNOWNNS !INCPATH'  
me='Cash12ToExcel'      /*
```

Development - INCLUDE

```
INCLUDE:  
Parse Value !inctrak.0+1 '!inc' ,  
  With !incix !inctrak.!incix ,  
  1 !inctrak.0 .  
Parse Arg !incfile.!incix, !incl.!incix, ,  
  !inc2.!incix, !inc3.!incix, !inc4.!incix, ,  
  !inc5.!incix, !inc6.!incix, !inc7,!incix, ,  
  !inc8.!incix  
  
If includes? Then Do  
  Say Copies(' ',!incix) 'enter' !incix ,  
  !inctrak.!incix !incfile.!incix  
End  
  
!incstream.!incix = .stream~new(!incpath ,  
  ||!incfile.!incix'.rex')  
!incarray.!incix = !incstream.!incix~charin( , ,  
  !incstream.!incix~chars)~makearray /* Read the file */  
!incstream.!incix~close  
Do !i.!incix = 1 To !incarray.!incix~items  
/* Process each array element */  
  !incline.!incix = !incarray.!incix[!i.!incix]  
  If !incline.!incix='' Then Iterate  
  
  !inctest=left(Strip(!incline.!incix,'L'),2)  
  If !inctest='/'||'*' Then Iterate  
  If !inctest='-'||'-' Then Iterate
```

```

    Interpret !incline.!incix
End
If includes? Then Do
    Say Copies(' ',!incix) 'exit ' !incix ,
        !inctrak.!incix !incfile.!incix
End
Parse Value !inctrak.0-1 '!inc' ,
    With !incix !inctrak.!incix 1 !inctrak.0 .
Return

```

Development - INCLUDE_LOGIC

```

INCLUDE_LOGIC:
Parse Value !inctrak.0+1 '!incl' With !incix !inctrak.!incix 1 !inctrak.0 .
Parse Arg !incfile.!incix,!incl,!inc2,!inc3,!inc4,!inc5,!inc6,!inc7,!inc8
Parse Arg !incfile.!incix,!incl.!incix,!inc2.!incix,!inc3.!incix, ,
    !inc4.!incix,!inc5.!incix,!inc6.!incix,!inc7.!incix,!inc8.!incix
If Symbol('INCLUDES?')='VAR' Then Do
    If includes? Then Do
        Say Copies(' ',!incix) 'enter' !incix !inctrak.!incix !incfile.!incix
    End
End
Else delayed?=1
    /* Fastest method in ooRexx to read in an entire file and convert it to an array */
    !incstream.!incix = .stream~new(!incpath||!incfile.!incix'.rex')
    !incarray.!incix = !incstream.!incix~charin(!,!incstream.!incix~chars)~makearray
    !incstream.!incix~close
    !inclines.!incix=''
    Do !i.!incix = 1 To !incarray.!incix~items /* Process each array element (rows in our text file) */
        !incline.!incix = !incarray.!incix[!i.!incix]
        If !incline.!incix='' Then Iterate
        If Strip(Substr(!incline.!incix,1,2))='/'||'*' Then Iterate
        If Strip(Substr(!incline.!incix,1,2))='-||'-' Then Iterate

        !inclines.!incix=!inclines.!incix||!incline.!incix;'
    End
    If !inclines.!incix\='' Then Interpret !inclines.!incix
    Else Say 'Nothing found in' !incstream.!incix 'to interpret by INCLUDE_LOGIC'
    If delayed? & includes? Then Do
        Say Copies(' ',!incix) 'enter' !incix !inctrak.!incix !incfile.!incix
        delayed?=0
    End
    If includes? Then Do
        Say Copies(' ',!incix) 'exit ' !incix !inctrak.!incix !incfile.!incix
    End
    Parse Value !inctrak.0-1 '!incl' With !incix !inctrak.!incix 1 !inctrak.0 .

```

Return

Development - SYNTAX

```
SYNTAX:
PROTECTALL:
lastline=Sourceline()                /* SHOW_SOURCE needs these */
_sigl=sigl
Interpret report_exception
Call include_logic 'show_source'
If !incix>1 Then Do
  Say 'Include active:'
  Do i=!incix-1 To 0 By -1
  Select
    When !inctrak.i='!inc' Then Do
      Say 'INCLUDE had' Translate(!incfile.i) 'running at index' i
      Say 'Statement executing:'
      Say !incline.i
    End
    When !inctrak.i='!incl' Then Do
      Say 'INCLUDE_LOGIC had' Translate(!incfile.i) 'running at index' i
      Say 'Statement executing:'
      Say !inclines.i
    End
  Otherwise Nop
End
End
End
/* If we have an error, we want to be sure to quit Excel, don't it will be an orphaned process */
If Symbol('XLOBJ')='VAR' Then xlobj~Quit
Exit
```

Development - Example of SYNTAX trap

```
*****RxException*****
CONDITION =NOVALUE
DESCRIPTION=A1
INSTRUCTION=SIGNAL
PROPAGATED =0
RC =RC
SIGL =181
SOURCE =WindowsNT COMMAND C:\THE\__tmprun.the
SOURCELINE =If !inclines.!incix\='' Then Interpret !inclines.!incix
```

```

STATUS      =OFF
*****RxException*****
C:\THE\__tmprun.the:
=====
Source lines.....
 170: !incarray.!incix = !incstream.!incix~charin(,!incstream.!incix~chars)~makearray
 171: !incstream.!incix~close
 172: !inclines.!incix=''
 173: Do !i.!incix = 1 To !incarray.!incix~items /* Process each array element (rows in our text file) */
 174: !incline.!incix = !incarray.!incix[!i.!incix]
 175: If !incline.!incix='' Then Iterate
 176: If Strip(Substr(!incline.!incix,1,2))='/'||'*' Then Iterate
 177: If Strip(Substr(!incline.!incix,1,2))='-||'-' Then Iterate
 178:
 179: !inclines.!incix=!inclines.!incix||!incline.!incix;'
 180: End
>>181: If !inclines.!incix\='' Then Interpret !inclines.!incix
 182: Else Say 'Nothing found in' !incstream.!incix 'to interpret by INCLUDE_LOGIC'
 183: If delayed? & includes? Then Do
 184: Say Copies(' ',!incix) 'enter' !incix !inctrak.!incix !incfile.!incix
 185: delayed?=0
 186: End
 187: If includes? Then Do
 188: Say Copies(' ',!incix) 'exit ' !incix !inctrak.!incix !incfile.!incix
 189: End
 190: Parse Value !inctrak.0-1 '!incl' With !incix !inctrak.!incix 1 !inctrak.0 .
 191: Return
 192:
=====
Include active:
INCLUDE_LOGIC had PROCESS_PRT_ARGS_AND_OPTIONS running at index 1
Statement executing:
Parse Value hoadate(monyy) With err? mon yy rest;log?=1;Do k=a1 To opt.0 /* Initialize the option flags */; Parse Var op

```

Results - Sizes

	Delin- quency	Cash12	Balance Sheet	Income/ Expense	Total
Bytes - Old	14570	10511	17120	11902	54103
Bytes - New	8125	4981	11220	6251	30577
Difference	6445	5530	5900	5651	23526
Percentage	44.23	52.61	34.46	47.48	43.48

Lines - Old	417	261	475	298	1451
Lines - New	257	126	328	159	870
Difference	160	135	147	139	581
Percentage	38.37	51.72	30.95	46.64	40.04

Results - Added

ADDED	Bytes	Lines
Include	542	15
include_logic	619	16
parse_args	255	7
TOTAL:	1416	38

Results - Includes

INCLUDES	Bytes	Lines				
ExcelSetup	1972	46	1	1	1	1
autofit	734	18	1	1	1	1
justify_and_draw_boxes	1356	37	1	1	1	1
insert_headers	732	17	1	1	1	1
SaveAndSetupPrint	1363	27	1	1	1	1
finish_up	638	20	1	1	1	1
process_args_and_options	1039	29	1	1	1	1
error_or_unknown	339	14	1	1	1	1
get_prev_file	788	17	1	0	1	0
show_syntax	647	19	1	1	1	1
TOTAL:	9608	244				

Net Lines Saved:	206	189	206	189
Added function/cmt lines:	46	54	59	50

Results - Performance

	Backup	Include	Delta	Percent
Low	37.672	38.172	0.500	0.01327
High	38.031	38.797	0.766	0.02014
Difference	0.359	0.625	0.266	
Median	37.829	38.500	0.671	0.01774
Average	37.858	38.511	0.653	0.01725

Summary and Conclusion

This INCLUDE facility has been used for two additional programs and has met all its Objectives.

Although some code is added, the additions are overshadowed by the lines saved.

A required change to the four existing programs was readily made by adding an option and making just one change to INCLUDED source.

Although there is a performance impact, it is minor for single use imbeds.

With apologies to Mike Cowlshaw, I find that this INCLUDE facility is productive and easy to use, allowing the programmer to readily modularize his code, similar in nature to the ::ROUTINE directive in ooRexx.