## Introduction

Every Rexx program that wants to use one or more Rexx Utility Functions (distributed with the Regina Rexx package) must begin with the following two statements:

**CALL RxFuncAdd 'SysLoadFuncs','regutil','SysLoadFuncs'  
CALL 'SysLoadFuncs'**

After that, the following functions may be used. ***This is a SUBSET of defined functions***, just those considered useful by me! When a CALLed routine returns a result, it is in the variable “result”. While the examples are shown as CALL instructions, they can also be coded as ***functions***; e.g.,

**SAY function(operands) -or- IF function(operands)=something THEN …**

NOTE that anyplace “stem” is referenced, the stem variable must be specified *without a period*. Also, stem.0 must contain (or will contain) the number of stem entries. Also, for convenience, the function names below are not shown in quotes, but generally you should call them enclosed in single quotes for clarity (they are external functions), though this is not technically required.

An improved version of the CALLs listed above would be the following (thanks to David Serls):

**IF RxFuncAdd( 'SysLoadFuncs','regutil','SysLoadFuncs' )<>0 THEN DO**

**SAY RxFuncErrmsg()**

**EXIT 16**

**END**

**ELSE CALL 'SysLoadFuncs'**

## File System Routines

**CALL SysFileTree filespec, stem [, options] [, tattrib]**

Find all files matching the *filespec* and write their information in *stem* (no period) in alphabetical order.

Default presentation looks like this:

**3/06/18 7:32p 3 A---- C:\Users\fbrac\OneDrive\Documents\Rexx\data1.txt**

Note that months with one digit start with a blank. The “3” above is filesize and is followed by the attributes. The filename starts in column 38.

*filespec* – may include wildcard characters \* (0 or more chars) and ? (1 char)

*options* – how directories are searched and how output is presented

F for files only (no directories)

D for directories only (no files)

B for both files and directories

S for also searching subdirectories

O for only reporting the full path of the file (filename starts in col 1)

T for returning the timestamp as yyyy/mm/dd/hh/mm (filename starts in col 37)

L for returning the timestamp as yyyy-mm-dd hh:mm:ss (filename starts in col 40)

H for widening the filesize column from 10 to 16 digits

I to find files irrespective of case

RECOMMENDATION: “FL” or “FO” depending on whether or not you need timestamp  
 There being no SORT option (as the DIR command has…), you may wish to follow this

command with CALL SysStemSort stem,’D’ for example to sort with newest first.

*tattrib* – specify the file attributes to be matched as a 5-character field of +’s and –‘s as noted below,

or an asterisk to mean all, ignoring the bit. Example: ‘\*-\*+\*’ for read-only non-directories.

Attribute ‘+’ match files ‘-’ match files

A with more than one hard link with exactly one hard link;

D with execute permission without execute permission;

H without read permission with read permission;

R without write permission with write permission;

S with owner id less than 10 with owner greater than or equal to 10.

NOTE: Chip Davis has pointed out that a similar result can be achieved without the Rexx Utilities by using a standard Rexx instruction similar to the following:

ADDRESS SYSTEM 'DIR /B/O:N' filespec WITH OUTPUT STEM stemname.

The final period is required, as you are identifying a stem. “stemname.0” will contain the line count.

**CALL SysGetFileDateTime filename [, which]**

Returns the timestamp associated with the *filename* as yyyy-mmm-dd hh:mm:ss.

*which* – **modify** for the last modification date [default];   
 **access** – for the last access date;

**create** for creation date

**CALL SysFileDelete filename**

Delete *filename* in the current directory, if not fully qualified.

**CALL SysCopyObject from, to**

Copy the *from* file to the *to* filename. If *to* is not fully qualified, the current directory is used.

**CALL SysMoveObject from, to**

Rename the *from* file to the *to* filename. If *to* is not fully qualified, the current directory is used. If *to* is in a different directory, the operation is effectively a COPY and ERASE.

**CALL SysFileSearch target, filename, stem [, option]**

Search *filename* for *target* (case-insensitive by default) and put the matching lines in *stem* (no period).

*option* – **C** for case-sensitive and/or

**N** to prepend each output line with the line number in *filename*; format: n:data (no leading 0s)

## System Routines

**CALL SysSleep n**

Sleep n seconds, where n can be fractional (e.g., 2.5).

**CALL SysDumpVariables [filename]**

Dump or display (the default) all variable names and their values (including stems).

## Console I/O Routines

**CALL SysCls**

Clear the screen.

**CALL SysCurPos [row], [col]**

Set the cursor to row:col and return the *former* position as “row col”. If called without operands, simply returns the former position. You can process this return with the following statement:

PARSE VALUE result WITH $row $col

**CALL SysTextScreenSize**

Returns screen size as “rows cols”. Because of the screen buffer, “cols” will be very large and useless.

**CALL RxMessageBox text [, title] [, button] [,icon]**

Display a message box on the screen and wait for the user to click a button in response. The message box is placed in a fixed position on the mid-right of the screen each time.

*text* – the message text

*title* – text for the title bar; default “Error!”

*button* – one of the following words, which self-define the button options presented:

**ok** [default], **yesno**, **yesnocancel**, **okcancel**, **retrycancel**, **abortretryignore**

*icon* – one of the following words:

**question** (question mark in a blue circle)

**exclamation** (exclamation mark in a yellow triangle)

**information** or **asterisk** ("i" in a blue circle)

**hand** or **stop** ("X" in a red circle) [default]

The return code (“result”) is one of the following representing the button clicked:

1 = OK

2 = Cancel

3 = Abort

4 = Retry

5 = Ignore

6 = Yes

7 = No

## Stem Manipulation Routines

**CALL SysStemSort stem, [, order] [, sensitivity] [, startpos] [, endpos] [, firstcol] [, lastcol]**

Sort everything in a stem as pure ASCII.

*stem* – stem name *without the period* and stem.0 *must* contain the number of elements

*order* – ascending [default] or descending

*sensitivity* – case sensitive or insensitive {don’t know default]

*startpos*, *endpos* – elements to sort [default all]

firstcol, lastcol – columns to sort [default all]

**CALL RegMultiStemSort [order] [, sensitivity] [, firstcol] [, lastcol] stem1, stem2 [, stem 3, …]**

Similar to SysStemSort, but while sorting the first stem, it keeps the others in sync. ***Remember*** that stem.0 must be set to the number of entries, and that it must be set for ***all*** the stems used! Also, for this version of the stem sort, ***the stem names must be enclosed in quotes***. Example:

CALL RegMultiStemSort ,,,,'show.title','show.rating','show.genre'

(Untested: Whether or not unique stem names, like title., rating., genre., still required quotes.)

**CALL RegStemRead filename, stem [, minlen] [, maxlen]**

Read the contents of a file into a stem.

*minlen*, *maxlen* – if specified, the minimum and/or maximum element lengths are returned

**CALL RegStemWrite filename, stem**

Write the elements of stem to a file, as above.

NOTE: You can’t specify the console as an output file, so if that is your desire, you need to write your own routine (or use mine: SubDisplayStem, available on request!)

**CALL RegStemSearch needle, haystack [, start] [, flags]**

Search for *needle* in stem *haystack*.

*needle* – what to search for

*haystack* - stem name *without the period* and stem.0 *must* contain the number of elements

*start* – starting index position [default 1]

*flags* – any combination of **C** (case-sensitive search as opposed to the default of insensitive);   
 **E** (exact match required as opposed to substring match); and/or **S**, indicating that the  
 stem is already sorted, which speeds up the sort.

**CALL RegStemDoOver (stem., index)**

Return 1 (True) if stem.index exists, else 0 (False), simulating the missing function in Regina Rexx, *DO stem OVER n*. Oddly, ***the period must be present in the stem specification!*** Note that while the manual indicates that you can specify a third operand, newstem, it *does not work*.

**CALL SysStemCopy (from, to [,fromindex, toindex, count [insert]])**

Copies *count* elements of *stem* from *to* to starting at *fromindex* to the *toindex* target in *to*. If the last parameter *insert* is “I” (for Insert), then elements are inserted; otherwise they are replaced. Note that using defaults, if *to* is larger than *from*, elements beyond the length of *to* will be deleted in *from.*

**CALL SysStemDelete (stem, index [, count])**

Delete *count* elements from *stem* starting at *index*. Default is a *count* of 1 element.

**CALL SysStemInsert (stem, index, value)**

Insert *value* into *stem* at the *index* position, shifting further entries up one.

## Not Included

The following functions are not included because I didn’t find them useful:

* SysCreateShadow (because the function is not implemented for NT)
* SysFileSystemType (not useful)
* SysMkDir (seems unlikely to be used)
* SysRmDir (seems unlikely to be used)
* SysSearchPath (possibly useful though)
* SysTempFileName (seems of limited value)
* SysSetFileDateTime (possibly useful though)
* SysIni (because I wrote my own …)
* SysBootDrive (limited value)
* SysWinVer / SysOS2Ver / SysLinVer / SysVersion (dull)
* SysUtilVersion (seems unlikely to be used)
* SysDriveInfo (seems of limited value unless you are trying to report this number)
* SysDriveMap (same)
* SysGetErrorText (limited value)
* SysSetPriority (limited value)
* SysQueryProcess (limited value)
* SysSwitchSession (limited value)
* SysSystemDirectory (limited value)
* SysVolumeLabel (limited value)
* SysWaitNamedPipe (limited value)
* SysCurState (limited value)
* SysGetKey (awkward and probably not necessary)
* SysTextScreenRead (limited value)
* All the Macro-Space Manipulation Routines (not implemented)
* All the Semiphore Routines (because I don’t see the value …)
* All the Character Set Conversion Routines (because I haven’t evaluated them yet)
* All the Trigonometric Routines (because I have no need for these …)

Fred Brack

[FredRBrack@gmail.com](mailto:FredRBrack@gmail.com)

Feb 21, 2020; Last Updated August 12, 2020