

THOR_Rexx

COLLABORATORS

	<i>TITLE :</i> THOR_Rexx		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		June 15, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	THOR_Rexx	1
1.1	THOR_REXX	1
1.2	THOR_REXX/--background--	2
1.3	THOR_REXX/BEEP	3
1.4	THOR_REXX/CHANGEWINDOW	3
1.5	THOR_REXX/CLOSEPROGRESS	4
1.6	THOR_REXX/CURRENTMSG	5
1.7	THOR_REXX/CURRENTSYSTEM	6
1.8	THOR_REXX/GETCLIP	7
1.9	THOR_REXX/GETCOMMANDINFO	7
1.10	THOR_REXX/GETFILELIST	9
1.11	THOR_REXX/GETGLOBALCONFIG	10
1.12	THOR_REXX/GETMESSAGEARRAY	11
1.13	THOR_REXX/GETMSGLISTSELECTED	12
1.14	THOR_REXX/GETSELECTEDEVENT	13
1.15	THOR_REXX/ISCONFERENCELIST	14
1.16	THOR_REXX/ISICONIFIED	15
1.17	THOR_REXX/ISMESSAGELIST	15
1.18	THOR_REXX/ISSTARTUPWINDOW	16
1.19	THOR_REXX/LEAVEBBS	16
1.20	THOR_REXX/LOCKGUI	17
1.21	THOR_REXX/NOP	17
1.22	THOR_REXX/OPENPROGRESS	18
1.23	THOR_REXX/PACKEVENTS	19
1.24	THOR_REXX/PUTCLIP	20
1.25	THOR_REXX/QUIT	20
1.26	THOR_REXX/REQUESTFILE	21
1.27	THOR_REXX/REQUESTINTEGER	22
1.28	THOR_REXX/REQUESTLIST	23
1.29	THOR_REXX/REQUESTNOTIFY	25

1.30 THOR_REXX/REQUESTSTRING	26
1.31 THOR_REXX/RESCAN	28
1.32 THOR_REXX/SAVEMESSAGE	28
1.33 THOR_REXX/SHOWCONFERENCE	29
1.34 THOR_REXX/SHOWPICTURE	30
1.35 THOR_REXX/SHOWTEXT	31
1.36 THOR_REXX/STARTEDITOR	32
1.37 THOR_REXX/THORTOFRONT	33
1.38 THOR_REXX/UNLOCKGUI	33
1.39 THOR_REXX/UPDATECONFWINDOW	34
1.40 THOR_REXX/UPDATEPROGRESS	34
1.41 THOR_REXX/VERSION	36
1.42 THOR_REXX/WBTOFRONT	36

Chapter 1

THOR_Rexx

1.1 THOR_REXX

--background--

BEEP

CHANGEWINDOW

CLOSEPROGRESS

CURRENTMSG

CURRENTSYSTEM

GETCLIP

GETCOMMANDINFO

GETFILELIST

GETGLOBALCONFIG

GETMESSAGEARRAY

GETMSGLISTSELECTED

GETSELECTEDEVENT

ISCONFERENCELIST

ISICONIFIED

ISMESSAGELIST

ISSTARTUPWINDOW

LEAVEBBS

LOCKGUI

NOF
OPENPROGRESS
PACKEVENTS
PUTCLIP
QUIT
REQUESTFILE
REQUESTINTEGER
REQUESTLIST
REQUESTNOTIFY
REQUESTSTRING
RESCAN
SAVEMESSAGE
SHOWCONFERENCE
SHOWPICTURE
SHOWTEXT
STARTEDITOR
THORTOFRONT
UNLOCKGUI
UPDATECONFWINDOW
UPDATEPROGRESS
VERSION
WBTOFRONT

1.2 THOR_REXX/--background--

NOTES

All function will set the THOR.LASTERROR variable on error. It will contain a description of why the function returned an error code.

All funtions will return a error level of 30 on failure. The reason for the failure is described in THOR.LASTERROR.

1.3 THOR_REXX/BEEP

NAME

BEEP -- Will flash the current THOR screen.

SYNOPSIS

BEEP

FUNCTION

This function will simply flash the screen THOR is running on. On OS 2.1 and higher, this flash is configurable in the Sound program.

INPUTS

RESULT

0 - success

EXAMPLE

NOTES

New for THOR 2.0

BUGS

SEE ALSO

1.4 THOR_REXX/CHANGEWINDOW

NAME

CHANGEWINDOW - change the size and position of windows

SYNOPSIS

CHANGEWINDOW WINDOW/A,LEFTEDGE/N,TOPEdge/N,WIDTH/N,HEIGHT/N

FUNCTION

This function will change the size and position of the given window identifier to the values you supply.

INPUTS

This function takes the following arguments:

WINDOW - This is a string that will identify which window you will change. Currently, these identifiers are defined:

MAINWINDOW - The main reader window

CONFWINDOW - The window with the list of conferences

NEWFILESWINDOW - The window with the list of new files

MSGLISTWINDOW - The window listing the messages in each conference

DOCKWINDOW - The function dock window (can not be resized)

LEFTEDGE - The new left edge position of the window
TOPEDGE - The new top edge position of the window
WIDTH - The new width of the window
HEIGHT - The new height of the window

RESULT

0 - Success
5 - Window was not open on the screen
30 - Failure

EXAMPLE**NOTES**

If one of the LEFTEDGE, TOPEDGE, WIDTH or HEIGHT arguments is missing, THOR will not change the already existing value for the missing argument(s).

BUGS**SEE ALSO**

1.5 THOR_REXX/CLOSEPROGRESS

NAME

CLOSEPROGRESS -- Close a progress bar window

SYNOPSIS

CLOSEPROGRESS REQ/N/A

FUNCTION

Close a progress bar window that was opened with
OPENPROGRESS

.

INPUTS

REQ/N/A - The number returned in result from
OPENPROGRESS

must

be supplied to this function.

RESULT

0 - success
30 - failure

EXAMPLE

See

UPDATEPROGRESS
for detailed example.

NOTES

New for THOR 2.0

BUGS

SEE ALSO

```
OPENPROGRESS
/
UPDATEPROGRESS
```

1.6 THOR_REXX/CURRENTMSG

NAME

CURRENTMSG -- return information about current message

SYNOPSIS

```
CURRENTMSG STEM/A
```

FUNCTION

Will return in stem variables information about the current message. Currently, the following information will be returned:

```
<stem>.BBSNAME    - The System/BBS this message is on.
<stem>.CONFNAME   - The conference name this message is in.
<stem>.MSGNR      - The message number of the message. This is the
                   internal message number in the THOR database.
```

Further information about a message can be obtained with READBRMESSAGE.

INPUTS

STEM - Base name of the stem variables to put data in. Must be in uppercase.

RESULT

```
0 - success.
30 - failure.
```

EXAMPLE

```
CURRENTMSG stem MSG
if(rc ~= 0) then
do
    say THOR.LASTEROR
    exit
end

say 'System/BBS name of message:' MSG.BBSNAME
say 'Conf name of message:' MSG.CONFNAME
say 'Message number:' MSG.MSGNR

exit
```

NOTES

New for THOR 2.0.

BUGS

SEE ALSO
READBRMESSAGE

1.7 THOR_REXX/CURRENTSYSTEM

NAME

CURRENTSYSTEM -- Get current System and Conference.

SYNOPSIS

CURRENTSYSTEM STEM/A

FUNCTION

This function will return the current System which is open or the selected System in Startup Window. If there's a current conference, this will also be returned.

Currently, the following information will be returned:

<stem>.BBSNAME - The current BBS/System.
<stem>.CONFNAME - The current conference. Note: Will be empty ("") if there is no current conference.

INPUTS

STEM - Base name of the stem variables to put data in. Must be in uppercase.

RESULT

0 - success.
1 - success, but no System was open.
30 - failure.

EXAMPLE

```
CURRENTSYSTEM stem CURRENT

if(rc = 30) then
do
    say THOR.LASTEROR
    exit
end
say CURRENT.BBSNAME

if(CURRENT.CONFNAME ~= "") then
    say CURRENT.CONFNAME

exit
```

NOTES

New for THOR 1.22.

Renamed from CURRENTBBS to CURRENTSYSTEM. The old CURRENTBBS will still work for backwards compatibility, but all script authors are advised to use CURRENTSYSTEM instead.

BUGS

SEE ALSO

CURRENTMSG

1.8 THOR_REXX/GETCLIP

NAME

GETCLIP -- Get a string from the system clipboard.

SYNOPSIS

GETCLIP UNIT/N,MAXCHARS/N

FUNCTION

This function will return in result a string currently in the supplied clipboard unit.

INPUTS

UNIT/N - The clipboard unit to get the string from.

Can be in the range of 0-255, although 0 is the unit supported with pasting etc. Defaults to unit 0.

If the unit supplied is out of the allowed range, 0 will be used.

MAXCHARS/N - Maximum number of characters to get from the clipboard. Defaults to 1000.

RESULT

0 - success

5 - No characters were read from the clipboard.

30 - failure

EXAMPLE

NOTES

New for THOR 2.0

BUGS

SEE ALSO

PUTCLIP

1.9 THOR_REXX/GETCOMMANDINFO

NAME

GETCOMMANDINFO -- Return information on available commands.

SYNOPSIS

GETCOMMANDINFO COMMAND=CMD,STEM/A

FUNCTION

Command to return information on available commands. If the command

parameter is given, the template for the command will be returned in the STEM variable. If the command is not given, all commands and templates will be returned. The data will be put into the following variables:

```
<stem>.COUNT      - Number of commands.  
<stem>.n           - Name of command 'n'.  
<stem>.n.TEMPLATE  - Template for command 'n'.
```

INPUTS

COMMAND - Command to get template for. If this parameter isn't given, all commands with templates will be returned.
STEM - Base name of the stem variables to put data in. Must be in uppercase.

RESULT

```
0 - success  
30 - failure
```

EXAMPLE

```
GETCOMMANDINFO command  
    REQUESTLIST  
    stem TEMP  
if(rc ~= 0) then  
do  
    say THOR.LASTEROR  
    exit  
end  
  
say 'Template for  
    REQUESTLIST  
    command:' TEMP  
  
GETCOMMANDINFO stem INFO  
if(rc ~= 0) then  
do  
    say THOR.LASTEROR  
    exit  
end  
  
say "The number of ARexx commands in THOR:" INFO.COUNT  
  
do i=1 to INFO.COUNT  
    say "Command :" INFO.i  
    say "Template:" INFO.i.TEMPLATE  
end  
exit
```

NOTES

BUGS

SEE ALSO

GETCOMMANDINFO

1.10 THOR_REXX/GETFILELIST

NAME

GETFILELIST -- Get selected/all entries in the file database win

SYNOPSIS

GETFILELIST OUTSTEM,ALL,CLIPBOARD=CLIP/S,FILENAME=FILE

FUNCTION

This function is meant be used when the file database window is open. The selected or all files can be returned in stem variables, or saved to disk or to the clipboard.

INPUTS

OUTSTEM - If present, this is the base name of the stem variables the selected/all files in the current list in the file database window will be put into.

The stem variables will be filled in like this:

<outstem>.COUNT - Number of entries selected/total

<outstem>.FILENAME.<n> - <n> are numbers from 1 to

<stem>.COUNT and will contain the filename.

<outstem>.FILENR.<n> - <n> are numbers from 1 to

<stem>.COUNT and will contain the number the file has in the database.

ALL - If present, all files in the current list will be returned.

CLIPBOARD/S - If present, a filelist of the selected/all files will be saved to the clipboard as they appear in the list.

Can be used in conjunction with FILENAME and OUTSTEM.

FILENAME - If present, a filelist of the selected/all files will be saved to disk using the supplied filename.

If the file already exist, this list will be appended at the end of the previous file.

Can be used in conjunction with CLIPBOARD and OUTSTEM.

RESULT

0 - success

5 - The file database window was not open

6 - The area list was the current list or there were no selected files.

30 - failure

EXAMPLE

```
GETFILELIST outstem LIST
if(rc ~= 0) then
do
  if(rc = 5) then say 'Filedatabase window not open.'
  if(rc = 6) then say 'No files selected.'
  if(rc = 30) then say THOR.LASTERROR
  exit
end
```

```

say 'You selected ' || LIST.COUNT || ' files'
do i=1 to LIST.COUNT
  say 'Nr:' || LIST.FILENR.i || ' Filename: ' || LIST.FILENAME.i
end

```

NOTES

New for THOR 2.0.

Note the difference in output format between using OUTSTEM and saving to clipboard/disk.

Further information about the files returned in OUTSTEM can be obtained with READBRFILE.

BUGS

SEE ALSO

READBRFILE

1.11 THOR_REXX/GETGLOBALCONFIG

NAME

GETGLOBALCONFIG - Obtain the global configuration for THOR

SYNOPSIS

GETGLOBALCONFIG STEM/A

FUNCTION

Function to obtain the global configuration from THOR. All the data will be put in the supplied stem variable.

Currently the following data will be returned:

<stem>.SAVEDIR	- The default directory for saving messages
<stem>.EDITORNAME	- The name of the external editor, if any
<stem>.EDITORTYPE	- INTERNAL or EXTERNAL
<stem>.QUOTECHAR	- The custom quote character(s)
<stem>.QUOTETYPE	- Quote type: 0 for Default, 1 for None, 2 for def+space, 3 for Custom, 4 for Initials.
<stem>.MSGFONTNAME	- Name of font for the message window
<stem>.LISTFONTNAME	- Name of font for the lists
<stem>.GADGETFONTNAME	- Name of font for the gadgets
<stem>.FSEFONTNAME	- Name of font for the internal editor
<stem>.PUBSCREENNAME	- Name of the public screen THOR is running on

INPUTS

STEM - Base name of the stem variable to put config in. Must be in uppercase. Required.

RESULT

0 - Success.
30 - Failure.

EXAMPLE

```
GETGLOBALCONFIG stem CFG
if(rc ~= 0) then
do
    say THOR.LASTEROR
    exit
end

say "THOR Global Config:"
say " Savedir.....:" GFG.SAVEDIR
say " Editor.....:" GFG.EDITORNAME
say " EditorType....:" GFG.EDITORTYPE
say " QuoteChar.....:" GFG.QUOTECHAR
say " QuoteType.....:" GFG.QUOTETYPE
say " Msgfont.....:" GFG.MSGFONTNAME
say " ListFont.....:" GFG.LISTFONTNAME
say " GadgetFont....:" GFG.GADGETFONTNAME
say " FSEFont.....:" GFG.FSEFONTNAME
say " PubScreenName.:" GFG.PUBSCREENNAME
say " Con Output....:" GFG.CONOUTPUT

exit
```

NOTES

This function does not return the same data as the GETGLOBALDATA function does. Some of the data returned from this function in THOR 1.22 can only be obtained with the GETGLOBALDATA function now.

BUGS

SEE ALSO

GETGLOBALDATA

1.12 THOR_REXX/GETMESSAGEARRAY

NAME

GETMESSAGEARRAY -- Get the message numbers valid for the current read mode in the given conference.

SYNOPSIS

GETMESSAGEARRAY CONFNAME=CONF/A,STEM/A,LOGINSTATE=LS/S

FUNCTION

There are currently 3 possible read modes in THOR: All, Unread and Search. THOR keeps all message numbers for all available messages in the present read mode internally. This command will return the message numbers for the current read mode. The message numbers will be sorted based on the read mode and the sort-by method selected in the Dock window. This will be in the same order as messages shown in the message list window. If the conference is not the current one, THOR will, in most cases, have to build the array for this command from scratch.

INPUTS

CONFNAME - The name of the conference on the current BBS.

STEM - If present, this is the base name of the stem variables the message numbers will be put into.

The stem variables will be filled in like this:

```
<stem>.COUNT - Number of entries
<outstem>.<n> - <n> are numbers from 1 to <stem>.COUNT
                and will contain a message number
```

LOGINSTATE - Will return the unread messages that were available when entering the current BBS, regardless of any changes that might have happened to the number of unread messages in the conference after this time.

RESULT

```
0 - success
5 - no message numbers to return
30 - failure
```

EXAMPLE

```
address(thorport)

GETMESSAGEARRAY 'comp.sys.amiga.programmer' array
if(rc ~= 0) then
do
    if(rc ~= 30) then exit
    say THOR.LASTEROR
    exit
end

do n=1 to array.count
    say "Msg nr: "array.n
end
say "Total: "array.count

exit
```

NOTES

New for THOR 2.0

BUGS

SEE ALSO

1.13 THOR_REXX/GETMSGLISTSELECTED

NAME

GETMSGLISTSELECTED -- Get the message numbers of the selected entries in the message list window for the current conference.

SYNOPSIS

```
GETMSGLISTSELECTED STEM/A
```


FUNCTION

From THOR 2.05, it is possible to multiselect messages in the message list window. This command will return the message numbers of the selected messages in the message list window. The message numbers will be returned in the order they appear in the message list window.

INPUTS

STEM - This is the base name of the stem variables the message numbers will be put into.

The stem variables will be filled in like this:

<stem>.COUNT - Number of entries
<outstem>.<n> - <n> are numbers from 1 to <stem>.COUNT
and will contain a message number

RESULT

0 - success
3 - no message numbers to return
5 - the message list window was not open
30 - failure

EXAMPLE

```
address(thorport)

GETMSGLISTSELECTED array
if(rc ~= 0) then
do
    if(rc ~= 30) then exit
    say THOR.LASTEROR
    exit
end

do n=1 to array.count
    say "Msg nr selected: "array.n
end
say "Total: "array.count

exit
```

NOTES

New for THOR 2.1

BUGS

SEE ALSO

1.14 THOR_REXX/GETSELECTEDEVENT

NAME

GETSELECTEDEVENT -- Will return the event number of
of the selected event in the Event
window.

SYNOPSIS

GETSELECTEDEVENT

FUNCTION

This function will return the number of the current event highlighted in the Event window. This number is the number the event has in the event database.

INPUTS

RESULT

0 - success
5 - event window is not open
30 - failure

EXAMPLE

NOTES

New for THOR 2.0

BUGS

SEE ALSO

READBREVENT

1.15 THOR_REXX/ISCONFERENCELIST

NAME

ISCONFERENCELIST -- Test if the conference list window is open or not.

SYNOPSIS

ISCONFERENCELIST

FUNCTION

This function will test if the conference list window is open or not.

INPUTS

RESULT

0 - the conference list window is open
5 - the conference list window is NOT open

EXAMPLE

NOTES

New for THOR 2.11

BUGS

SEE ALSO

1.16 THOR_REXX/ISICONIFIED

NAME
ISICONIFIED -- Test if THOR is in an iconified state

SYNOPSIS
ISICONIFIED

FUNCTION
This function will test if THOR is in an iconified state

INPUTS

RESULT
0 - THOR is iconfied
5 - THOR is NOT iconified

EXAMPLE

NOTES
New for THOR 2.11

BUGS

SEE ALSO

THORTOFRONT

1.17 THOR_REXX/ISMESSAGELIST

NAME
ISMESSAGELIST -- Test if the message list window is open or not.

SYNOPSIS
ISMESSAGELIST

FUNCTION
This function will test if the message list window is open or not.

INPUTS

RESULT
0 - the message list window is open
5 - the message list window is NOT open

EXAMPLE

NOTES
New for THOR 2.11

BUGS

SEE ALSO

1.18 THOR_REXX/ISSTARTUPWINDOW

NAME

ISSTARTUPWINDOW -- Test if the Startup window is open

SYNOPSIS

ISSTARTUPWINDOW

FUNCTION

This function will test if the Startup window is open

INPUTS

RESULT

0 - the Startup window is open, no system is entered
5 - the Startup window is NOT open

EXAMPLE

NOTES

New for THOR 2.11

BUGS

SEE ALSO

THORTOFRONT

1.19 THOR_REXX/LEAVEBBS

NAME

LEAVEBBS - Will leave the current BBS (if any) and return to the Startup window.

SYNOPSIS

LEAVEBBS

FUNCTION

This function will simply close all open windows on a BBS and return you to the Startup window. It will not do anything if you are already in the Startup window.
Events on the closing BBS will be packed if the bbs type defines a packer.

INPUTS

RESULT

0 - success
5 - The user was already in the Startup window.
30 - failure

EXAMPLE

NOTES

BUGS

SEE ALSO

1.20 THOR_REXX/LOCKGUI

NAME

LOCKGUI -- Lock the GUI in THOR

SYNOPSIS

LOCKGUI

FUNCTION

This function will disable all user input. All windows will be locked with waitpointers.

INPUTS

RESULT

0 - success

EXAMPLE

NOTES

New for THOR 2.0

Normally, this function isn't needed in a ARexx script.

Be sure to match this with an `

UNLOCKGUI

` (use of ARexx `on error`

is strongly recommended)!

BUGS

SEE ALSO

UNLOCKGUI

1.21 THOR_REXX/NOP

NAME

NOP -- A do-nothing command.

SYNOPSIS

NOP

FUNCTION

The NOP (NO-oPeration) instruction is provided to control the binding of ELSE clauses in compound IF

statements.

INPUTS

RESULT

0 - success

EXAMPLE

```

if i = j then                /* First (outer) IF */
  if j = k then a = 0       /* Inner IF */
  else NOP                  /* Binds to inner IF */
else a = a + 1              /* Binds to outer IF */

```

NOTES

BUGS

SEE ALSO

1.22 THOR_REXX/OPENPROGRESS

NAME

OPENPROGRESS -- Open a window with a progressbar

SYNOPSIS

```

OPENPROGRESS TITLE/A, TOTAL/N, ABORTTEXT=AT
              PROGRESSTEXT=PT, PROGRESSCHARWIDTH=PCW/N

```

FUNCTION

This function will open a window with a progressbar, an optional abort-gadget (with selectable text) and a text information field.

INPUTS

TITLE/A - Required title text for the progress window
TOTAL/N - Total number of levels in the progressbar. Defaults to 100.
ABORTTEXT - The text in abort gadget. '_' in front of a character will make that character the keyboard shortcut. If this argument is not supplied, no gadget will be created and the user can not abort the progress window.
PROGRESSTEXT - Text to be shown over the progress bar.
PROGRESSCHARWIDTH - Optional width of progressbar in characters. Will default to the number of characters supplied in PROGRESSTEXT.

RESULT

0 - success, in which case result will contain a number that *must* be supplied to
UPDATEPROGRESS
and
CLOSEPROGRESS
in the REQ
parameter.
30 - failure

EXAMPLE

See
 UPDATEPROGRESS
 for detailed example.

NOTES

New for THOR 2.0

Make sure the number in result remains unchanged and pass this number to

 CLOSEPROGRESS
 and
 UPDATEPROGRESS
 without modification.

If modified, nasty things will surely happen.

The use of ARexx 'on error' is strongly recommended.

BUGS

SEE ALSO

 UPDATEPROGRESS
 ,
 CLOSEPROGRESS

1.23 THOR_REXX/PACKEVENTS

NAME

PACKEVENTS - Make event package for a BBS

SYNOPSIS

PACKEVENTS BBSNAME=BBS/A

FUNCTION

This function will make a new event package to be uploaded to a BBS. If the BBS given do not support event packages (eg .REP files for QWK), this function will return without doing anything.

If events like ReplyMsg or EnterMsg are added to a BBS with the WRITEBREVENT command, this command *must* be used to make sure the event package is updated with the new messages!

INPUTS

This function takes only one argument:

BBSNAME - The name of the BBS to make an event package for

RESULT

0 - success
30 - failure

EXAMPLE

NOTES

BUGS

SEE ALSO
WRITEBREVENT

1.24 THOR_REXX/PUTCLIP

NAME

PUTCLIP -- Send a file or string to the system clipboard.

SYNOPSIS

PUTCLIP UNIT/N, FILE, STRING

FUNCTION

This function will take a file and/or a string and put it into the system clipboard.

INPUTS

UNIT/N - The clipboard unit to put the string/file to.
Can be in the range of 0-255, although 0 is the unit supported with pasting etc. Defaults to unit 0.
If the unit supplied is out of the allowed range, 0 will be used.
FILE - A filename of a file you want to put into the clipboard.
STRING - A string you want to put into the clipboard.

RESULT

0 - success
30 - failure

EXAMPLE

NOTES

New for THOR 2.0

BUGS

SEE ALSO

GETCLIP

1.25 THOR_REXX/QUIT

NAME

QUIT -- Will quit THOR

SYNOPSIS

QUIT FORCE/S

FUNCTION

Will quit the addressed THOR.
The FORCE keyword is optional and will close all internal editors without asking. Use the FORCE option with caution since messages etc. in the editors will not be saved before closing.
Without FORCE, THOR will ask the user to close all editors first.

INPUTS

FORCE - "Hostile" exit of THOR, without any confirmation.

RESULT

0 - success

EXAMPLE**NOTES****BUGS****SEE ALSO**

1.26 THOR_REXX/REQUESTFILE

NAME

REQUESTFILE - For requesting a file from the user

SYNOPSIS

REQUESTFILE TITLETEXT=TITLE/A, INITIALDRAWER=ID/A, INITIALFILE=IF
FULLPATH=FP/S, FILEPATTERN=PAT

FUNCTION

This function will open a standard ASL filerequester and let the user select a file from this requester.
The filename can be returned with full path or just the filename.

INPUTS

REQUESTFILE takes the following arguments:

TITLETEXT - The text that is to be placed in the titlebar of the requester.

INITIALDRAWER - The initial drawer on disk that will come up in the requester

INITIALFILE - The initial file that will be in the file field of the requester

FULLPATH - Is set, this function will return the full path to the selected file. If not present, only the filename will be returned.

FILEPATTERN - This is a standard AmigaDOS filepattern which will be used to filter out files that are to be shown in the requester.

"~#?" can be used if you only want directories to be shown in the requester.

RESULT

```
0 - Success. Result will contain the filename selected
5 - User canceled the requester
30 - Failure
```

EXAMPLE

```
REQUESTFILE TITLE 'Select a file:' ID 'Work:dloads' FP PAT '~#?'
if(rc ~= 0) then
do
  say THOR.LASTEROR
  exit
end

say result

exit
```

NOTES

No checking is made if the user do not select a file but still selects OK in the requester.

"~#?" can be used if you only want directories to be shown in the requester.

BUGS

SEE ALSO

1.27 THOR_REXX/REQUESTINTEGER

NAME

REQUESTINTEGER -- Will open an number-entry requester

SYNOPSIS

```
REQUESTINTEGER MIN/A/N,MAX/A/N,INITIALINTEGER=INIT/N,
TITLETEXT=TITLE/A,BUTTONTEXT=BT/A,VAR
```

FUNCTION

Will let the user enter a number within the given range. No other characters than a number will be accepted.

INPUTS

```
MIN/A/N - Minimum number allowed
MAX/A/N - Maximum number allowed
INITIALINTEGER=INIT/N - Initial number to be displayed
TITLETEXT=TITLE/A - Title of the requester
BUTTONTEXT - The text to be in the gadgets and the
  shortcut can be set with '_' in front if the
  key that is to act as a shortcut. The text
  for each gadget is separated with a '|', and
```

it's possible to define 1 or more gadgets,
although anything else than 2 gadgets doesn't
make much sense.

VAR - Optional variable to put the entered number
into. If not supplied, result will be used.

RESULT

0 - success
5 - user cancelled the requester
30 - failure

EXAMPLE

```
REQUESTINTEGER MIN 12 MAX 100 INIT 34 TITLE 'Enter a number:' BT 'Ok| ←
Cancel'' var NUMBER
if(rc ~= 0) then
do
  say THOR.LASTEROR
  exit
end

say "Number entered : " NUMBER

exit
```

NOTES

BUGS

SEE ALSO

1.28 THOR_REXX/REQUESTLIST

NAME

REQUESTLIST - Select from a supplied list of entries

SYNOPSIS

```
REQUESTLIST INSTEM/A,OUTSTEM,TITLE,DRAGSELECT/S,MULTISELECT/S,
SIZEGADGET/S
```

FUNCTION

Function for selecting entries from the list given in the STEM
variable. Drag and multiselection is possible.
The window will be scaled to fit all supplied entries in their
full length and will not be smaller than the width of the TITLE
text.

INPUTS

INSTEM - Base name of the stem variables the tag data is in.
Must be supplied or this function will fail. The stem
variables must be filled like this:

```
<instem>.COUNT - Number of entries
<instem>.<n> - <n> are numbers from 1 to <stem>.COUNT and
must contain a string. All entries upto
```

<stem>.COUNT must be filled with a valid string and no newline characters is allowed.

OUTSTEM - Base name of the stem variables the selected entries in list will be put in. Must be supplied or this function will fail. The stem variables will be filled like this:

```
<outstem>.COUNT - Number of entries
<outstem>.<n> - <n> are numbers from 1 to <stem>.COUNT and
will contain a string. All entries upto
<stem>.COUNT will be filled with a valid
string.
```

TITLE - Optional window title. If not given, a default (localized) window title will be supplied.

DRAGSELECT - If present, the listview will support dragselection of entries in the list. Implies MULTISELECT.

MULTISELECT - If present, the listview will support shift-key mouse selection of several entries.

SIZEGADGET - If present, the window will contain a size gadget and a zoom gadget.

MAXHEIGHT - The maximum height the window will have in pixels. If not present, maxheight will be 1/2 of the visual screen height.

SEARCHGADGET - If present, a search gadget with an search option will be available. This argument will currently only be present when DRAGSELECT and/or MULTISELECT is given.

RESULT

```
0 - Success
5 - User canceled the window
30 - Failure
```

Note: Even with multiselect, there can be a selected entry, which will be marked in bold and underline. On success, result will contain the string selected.

OUTSTEM will not be used if the MULTISELECT or DRAGSELECT argument is not given. Result will be used on in these single-select cases.

EXAMPLE

```
drop LIST.          /* Important! */

LIST.COUNT = 3
LIST.1 = 'First Entry'
LIST.2 = 'Second Entry'
LIST.3 = 'Third Entry'

/* Multi and dragselect from the list, no sizegadget */

REQUESTLIST instem LIST outstem SELECTED MULTISELECT DRAGSELECT title ' " ←
  Select entries:"'
if(rc ~= 0) then
```

```

do
  say THOR.LASTEROR
  exit
end

say 'You selected: ' || SELECTED.COUNT
do i=1 to SELECTED.COUNT
  say SELECTED.i
end

```

NOTES

This function is completely changed from REQUESTLIST in THOR 1.25, and not compatible in *any* way.

This function is well suited for use in combination with GETBBSLIST or some of the other list functions in BBSREAD. The resulting stem variable from e.g. GETBBSLIST can be used directly as INSTEM for this function.

Changed for 2.1:

Previously, when using multi/dragselection, the white single-selected entry would not be present in the outstem variable, only in result. Starting from this version, it will be in both.

The SEARCHGADGET argument is new for THOR 2.11.

BUGS

SEE ALSO

GETBBSLIST, GETCONFLIST, GETFAREALIST

1.29 THOR_REXX/REQUESTNOTIFY

NAME

REQUESTNOTIFY - will open an information requester

SYNOPSIS

REQUESTNOTIFY TEXT/A,BUTTONTEXT=BT/A

FUNCTION

This function will open a requester with some information text within, specified in the TEXT argument. Useful for informing the user about possible errors, etc.

INPUTS

The function takes the following arguments:

TEXT - This is the text that is to be shown to the user in the requester.
 TEXT can, currently, contain upto 2000 characters and will be processed like the signatures in THOR. This means that variables like \$ver and \$time can be used and will be translated. Note that the 2000 limit is the limit *after* any \$ver and \$time variables has been expanded, so the practical limit

is more like 1950 characters.
 '\n' (backslash n) can be used to add a newline.

BUTTONTEXT - The text to be in the gadgets and the shortcut can be set with '_' in front if the key that is to act as a shortcut. The text for each gadget is separated with a '|', and it's possible to define 1 or more gadgets.

RESULT

0 - Success, in which case the value in result will be 1 (TRUE) for leftmost (positive) response, then each consecutive response will return 1 more, the rightmost (false) response will return 0 (FALSE) in result.
 30 - Failure

EXAMPLE

```
REQUESTNOTIFY TEXT "Do you feel fine?" BT "_YES!|_MAYBE|_NO"
if(rc ~= 0) then
do
  say THOR.LASTEROR
  exit
end

if(result = 0) then say 'Sorry to hear that!'
if(result = 1) then say 'I feel fine too!'
if(result = 2) then say "Ok, I'm not sure either.."
```

exit

NOTES

Do not use this function if you intend to show large amounts of text.

```
REQUESTLIST
or
SHOWTEXT
is more suitable for this purpose.
```

BUGS

SEE ALSO

```
REQUESTLIST
,
SHOWTEXT
```

1.30 THOR_REXX/REQUESTSTRING

NAME

REQUESTSTRING - For requesting a string from the user

SYNOPSIS

REQUESTSTRING TITLETEXT=TITLE/A,BUTTONTEXT=BT,INITIALSTRING=ID

BODYTEXT=BODY,MAXCHARS/N/A,INVISIBLE=QUIET/S

FUNCTION

This function will open a string requester that will accept any string input you give it, upto MAXCHARS number of characters.

INPUTS

REQUESTSTRING takes the following arguments:

TITLETEXT - The text that is to be placed in the titlebar of the requester.

BUTTONTEXT - The text to be in the gadgets and the shortcut can be set with '_' in front if the key that is to act as a shortcut. The text for each gadget is separated with a '|', and it's possible to define 1 or more gadgets. If you use more than 2 gadgets, the THORRC variable should be checked to find out what gadget that was selected.

INITIALSTRING - If present, the supplied string will be put in the string gadget initially.

BODYTEXT - The optional text that is to appear above the string gadget in the requester.
'\n' (backslash n) can be used to add a newline.

MAXCHARS - This is the maximum number of characters the user is allowed to enter into the gadget.
MAXCHARS can be no more than 1000.
Do not set MAXCHARS to a number higher than you need, it will just be a waste of memory.
Default is 100 characters.

INVISIBLE - When this keyword is present, letters entered will not be echoed, only dots will appear for typed letters. Use this on passwords etc.

RESULT

0 - Success. Result will contain the string entered
5 - User canceled the requester or didn't enter any text
30 - Failure

On success, this function will set the THORRC variable to 1 for leftmost (positive) response, then each consecutive response will return 1 more, the rightmost (false) response will return 0 in THORRC.

EXAMPLE

```
REQUESTSTRING TITLE "Enter something:" BT "_Ok|_Cancel" ID "Some text ←
"
if(rc ~= 0) then
do
say THOR.LASTEROR
```

```
        exit
    end

    say result

    exit
```

NOTES

THORRC is new for THOR 2.1.

BUGS**SEE ALSO**

1.31 THOR_REXX/RESCAN

NAME

RESCAN - Scan for mail packets in the download directory

SYNOPSIS

RESCAN

FUNCTION

This function will, if the Startup window is open, scan for new mail packets (grabs) in the configured download directory and show them with a '*' in the Startup window.

INPUTS**RESULT**

0 - Success

EXAMPLE**NOTES**

Currently, no error codes will be returned to arexx upon failure in the scan for grabs. However, an informative error message will be shown to the user in a requester if this command fails for one reason or another.

BUGS**SEE ALSO**

1.32 THOR_REXX/SAVEMESSAGE

NAME

SAVEMESSAGE - Save a message in THOR to disk or to the clipboard

SYNOPSIS

SAVEMESSAGE BBSNAME=BBS, CONFNAME=CONF, MSGNUMBER=MSGNR/N, CURRENT/S,
FILENAME=FILE, CLIPBOARD=CLIP/S, NOHEADER/S, NOANSI/S

OVERWRITE/S

FUNCTION

Function to save a message to disk or to the clipboard. The message can be saved with or without ANSI codes and a header field.

INPUTS

BBSNAME - Name of the BBS
CONFNAME - Name of the conference
MSGNUMBER - Message number in database
CURRENT - Save current message
FILENAME - File name to save message as
CLIPBOARD - Save message to clipboard
NOHEADER - Don't save any header fields with the message
NOANSI - Strip all ANSI codes from the message
OVERWRITE - Will overwrite any existing file with the same filename instead of appending to the end of the same file. Use this argument with caution.

RESULT

0 - Success
5 - Message is deleted.
30 - Failure

EXAMPLE

NOTES

The NOANSI argument is new for THOR 2.0.

FILENAME and CLIPBOARD can both be used with the effect that the message will be saved both to the clipboard and to a file.
CURRENT will override BBSNAME, CONFNAME and MSGNUMBER.

The OVERWRITE argument is new for THOR 2.1.

BUGS

SEE ALSO

1.33 THOR_REXX/SHOWCONFERENCE

NAME

SHOWCONFERENCE -- Show the given conference, as if it was selected from within THOR.

SYNOPSIS

SHOWCONFERENCE CONFNAME=CONF/A

FUNCTION

This function will show the given conference, just as if the conference was selected for reading directly from within THOR. The read mode currently in use will be used, and the conference must exist on the current BBS.

INPUTS

CONFNAME - Name of the conference on the current BBS to show.

RESULT

0 - success
5 - The ReplyMsg/EnterMsg window was open, in which case this function can't be executed.
30 - failure

EXAMPLE

NOTES

New for THOR 2.0

This command can't be used while the ReplyMsg/EnterMsg window is open. This window uses the current conference pointer and this pointer must not be changed while this window is open.

The conference enter ARexx script will *NOT* be run as usual when this function is used. This is to avoid loops where a script calls this function, which again starts a new script etc.

BUGS

SEE ALSO

1.34 THOR_REXX/SHOWPICTURE

NAME

SHOWTEXT

-- View a picture using datatypes in the message window.

SYNOPSIS

SHOWTEXT

FILENAME=FILE/A,CLEAR/S

FUNCTION

This function will put the supplied picture into the message list area using datatypes.

INPUTS

FILENAME - Filename of the picture to view

CLEAR - This option will clear the contents of the message list area first so that only the picture will be present.

RESULT

0 - success
5 - the main window was not open
30 - failure

EXAMPLE

NOTES

New for THOR 2.1

This function requires AmigaOS 3.0 or higher and datatypes.library installed and working properly.

Running THOR in 16 colors or more is recommended.

BUGS

SEE ALSO

1.35 THOR_REXX/SHOWTEXT

NAME

SHOWTEXT -- Show a textfile in the message window.

SYNOPSIS

SHOWTEXT FILENAME=FILE/A,CURRENTMSGTEXT=CMT/S,APPEND/S

FUNCTION

This function will put the supplied text file into the message list area. The text file will be processed just like any ordinary message text. ANSI codes, Style Tags, etc. will be shown according to your configuration settings.

INPUTS

FILENAME - Filename of text file to show

CURRENTMSGTEXT - This switch will cause THOR to quote the text currently visible in the message text area, and not the text on disk when replying to the message.
Note: This option will only work if no other message text is shown in the message text area before the message is replied to.

APPEND - If there is already existing text in the message window, the new text will just be appended and will not overwrite the already existing text or gfx.

RESULT

0 - success
30 - failure

EXAMPLE

NOTES

New for THOR 2.0

Calling this function with a binary file as FILENAME might cause problems.

APPEND is new for THOR 2.21.

BUGS

SEE ALSO

1.36 THOR_REXX/STARTEDITOR

NAME

STARTEDITOR -- Start the configured editor in THOR on a file

SYNOPSIS

STARTEDITOR FILENAME=FILE/A

FUNCTION

This function will start the configured editor (internal or external) on the given file.

INPUTS

FILENAME - Filename to start the editor with. Must be supplied or this function will fail. Note: This must be the complete path to the file. Can be a non-existing file as long as the editor supports this.

RESULT

0 - success
30 - failure

EXAMPLE

```
STARTEDITOR "ram:test.file"
if(rc ~= 0) then
do

    REQUESTNOTIFY
    TEXT "'THOR.LASTEROR'" BT "_Ok"
    exit
end

exit
```

NOTES

New for THOR 2.0

If the configured editor is the internal editor in THOR, the ARexx port for the started editor will be available as normal.

Not all external editors accept a non-existing file as argument upon startup. CED and TurboText does, but Edge doesn't.

The editor will be started async as a separate process from THOR.

The line length in the internal editor will be 80 characters.

BUGS

SEE ALSO

1.37 THOR_REXX/THORTOFRONT

NAME

THORTOFRONT -- Will bring the THOR screen to front.

SYNOPSIS

THORTOFRONT

FUNCTION

Will bring the screen the addressed THOR is using to the front.

INPUTS

RESULT

0 - success

EXAMPLE

NOTES

If THOR is iconfied, THOR will be uniconified and brought to front.

BUGS

SEE ALSO

1.38 THOR_REXX/UNLOCKGUI

NAME

UNLOCKGUI -- Unlock the GUI in THOR after a call to LOCKGUI

SYNOPSIS

LOCKGUI

FUNCTION

This function will enable all user input after a previous call to

LOCKGUI

.

INPUTS

RESULT

0 - success

EXAMPLE

NOTES

New for THOR 2.0

Safe to call even if
LOCKGUI
has never been called.

BUGS

SEE ALSO

LOCKGUI

1.39 THOR_REXX/UPDATECONFWINDOW

NAME

UPDATECONFWINDOW -- Updates the conference window

SYNOPSIS

UPDATECONFWINDOW

FUNCTION

This function will update the conferences in the conference window so all information is up-to-date, with correct number of messages, conference flags and names. Use this function if you have added or deleted messages from a conference.

INPUTS

RESULT

0 - success
5 - Conference window was not open.
30 - failure

EXAMPLE

NOTES

New for THOR 2.0

BUGS

SEE ALSO

1.40 THOR_REXX/UPDATEPROGRESS

NAME

UPDATEPROGRESS -- Update a progressbar opened with
OPENPROGRESS

SYNOPSIS

UPDATEPROGRESS REQ/N/A,CURRENT/N,TOTAL/N,PROGRESSTEXT=PT

FUNCTION

This function will update a progressbar window.

INPUTS

REQ/N/A - The number returned in result from
OPENPROGRESS

CURRENT/N - Change the current level of the progressbar

TOTAL/N - The progressbar will be reset to it's initial state if
this argument is provided with a total value.

PROGRESSTEXT - Text to be shown over the progress bar.

RESULT

0 - success

5 - The user has pressed the abort gadget and your ARexx script
should close the progress window with
CLOSEPROGRESS
and
abort. It is your responsibility to check this and stop
your ARexx script.

30 - failure

EXAMPLE

```

OPENPROGRESS
  TITLE "Progress Test" AT "_Abort" PCW 30
if(rc = 0) then
do
  window = result      /* Important !!! */

do i=1 to BBSLIST.COUNT
  /* Display bbsname and reset progressbar.*/
  UPDATEPROGRESS REQ window TOTAL 50 CURRENT 1 PT "'BBSLIST.i'"
  do n=1 to 50
    /* Update progressbar 50 times for each bbs */
    UPDATEPROGRESS REQ window CURRENT n
    /* Check if user aborted */
    if(rc ~= 0) then do

CLOSEPROGRESS
  REQ window
  exit
    end
  end
end
end
end

```

NOTES

New for THOR 2.0

BUGS

SEE ALSO

```

OPENPROGRESS
,
CLOSEPROGRESS

```

1.41 THOR_REXX/VERSION

NAME

VERSION -- Get version information from THOR

SYNOPSIS

VERSION STEM/A

FUNCTION

This function will return version information from THOR.
Currently, the following information will be returned:

<stem>.THOR - The release version of THOR
<stem>.BBSREAD - The version of the bbsread.library
<stem>.UTGUI - The version of the utgui.library
<stem>.FSE - The version of the THOR internal editor
<stem>.REQTOOLS - The version of the reqtools.library

INPUTS

STEM - Base name of the stem variables to put data in. Must be in
uppercase.

RESULT

0 - success
30 - failure

EXAMPLE

```
VERSION stem VER

say "THOR           : " VER.THOR
say "bbsread.library : " VER.BBSREAD
say "utgui.library  : " VER.UTGUI
say "FSE           : " VER.FSE
say "reqtools.library : " VER.REQTOOLS

exit
```

NOTES

New for THOR 1.22.

BUGS

SEE ALSO

1.42 THOR_REXX/WBTOFRONT

NAME

WBTOFRONT -- Will bring the Workbench screen to front.

SYNOPSIS

WBTOFRONT

FUNCTION

Will bring the Workbench screen to front. To bring the THOR screen to front again, use

THORTOFRONT

.

INPUTS

RESULT

0 - success

30 - failure

EXAMPLE

NOTES

New for THOR 2.0

BUGS

SEE ALSO

THORTOFRONT
