

QuickMsg

Axel Greve

COLLABORATORS

	<i>TITLE :</i> QuickMsg		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Axel Greve	August 13, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1 QuickMsg	1
1.1 QuickMsg.rexx	1
1.2 Requirements	2
1.3 Installation	2
1.4 Usage	2
1.5 Examples	3
1.6 Examples 1 & 2	4
1.7 Example 3	4
1.8 Example 4	4
1.9 Example 5	4
1.10 Examples 6 & 7	5
1.11 Example 8	5
1.12 Example 9	6
1.13 Beispiel 10	6
1.14 Hints	7
1.15 The legal stuff	7
1.16 Thanks	7
1.17 History	8
1.18 Author	8

Chapter 1

QuickMsg

1.1 QuickMsg.rexx

QuickMsg.rexx 1.0

Program and documentation by A. Greve

QuickMsg.rexx has been written to display a message in an own window for a selectable time on a (also selectable) PubScreen without having to press a button to continue. So the window doesn't contain any buttons, but the window won't block any in-/outputs and it will disappear after some time anyway. Thus, messages like

»Couldn't find presets, gonna use default settings«
can be displayed without interrupting the calling program and having to confirm this message.

Requirements

Installation

Usage

Examples

Examples 1 & 2

-> D=..., P=...

Example 3

-> FROM filename

Example 4

-> \n (line feed)

Example 5

-> " (double quote)

Examples 6 & 7

-> Batch usage, FILE filename

Example 8

-> M=...

Example 9

```
-> *FROM, *FILE
```

Example 10

```
-> W=..., H=...
```

Hints

The legal stuff

Thanks

History

Author

For representing the text output and as a substitute for double quotes the

symbols » and « are used in this description. These symbols won't appear in the display of course, they are only for symbolizing the left and right margins of the display. To get differentiation from the real double quotes in this description the symbols are used as well.

1.2 Requirements

- * rexxsupport.library v34.9+, part of the ARexx installation
 - * triton.library v1.3+, by Stefan Zeiger, Aminet: dev/gui
 - * tritonrexx.library v37.6+, by Jürgen Kohrmeyer, Aminet: util/rexx
- If the listed libraries are not installed, the text will be displayed in a CLI window.
- * Kickstart/Workbench 2.04+ (see the requirements of the libs above in the corresponding docs).
 - * REXXmast must be running.

1.3 Installation

Copy the program to the path where You keep Your ARexx scripts (usually REXX:).

Copy the documentation to a place where You keep Your docs.

1.4 Usage

The program can only be used from CLI/Shell.

Start the program with "rx [Pathname]QuickMsg[.rexx] [Argument]". Pathname (ending with ":" or "/") and the extension ".rexx" are not required if the script is copied to REXX:. Entering ? as the argument the programm will show a quick help.

You will find some applications at

Examples

, describing the usage of

and the argument passing to QuickMsg in detail.

If You use an "AppLauncher" like ToolManager, the program can be started from a button.

The program can be called in 3 different ways:

- 1.) rx QuickMsg.rexx <Text> [D=<Seconds>] [P=<PubScreen>] [M=<Textformat>] [W=< ←
Width>] [H=<Height>]
- 2.) rx QuickMsg.rexx FILE <Filename> [D=<Seconds>] [P=<PubScreen>] [M=<Textformat ←
>] [W=<Width>] [H=<Height>]
- 3.) rx QuickMsg.rexx FROM <Filename>

<Text> or FILE <Filename> always must be the first argument, the further arguments are optional and their order is unimportant.

Argument meanings:

<Text>	Text to be displayed
FILE <Filename>	Display a text file, arguments from command line
FROM <Filename>	Display a text file, arguments from text file
D=<Integer>	Display time in seconds
P=<String>	Name of PubScreen the window is to be opened on
M=<Integer>	Display mode for text, see Example 8
W=<Integer>	Text width in characters
H=<Integer>	Text height in lines

1.5 Examples

The usage of [Argument] is described best by some examples:

Examples 1 & 2

-> D=..., P=...

Example 3

-> FROM filename

Example 4

-> \n (line feed)

Example 5

-> " (double quote)

Examples 6 & 7

-> Batch usage, FILE filename

Example 8

-> M=...

Example 9

-> *FROM, *FILE

Example 10

-> W=..., H=...

1.6 Examples 1 & 2

```
rx QuickMsg Display this for 10 secs D=10 P=DOPUS.1
```

will show »Display this for 10 secs« for ten seconds on the PubScreen of DirOpus. D=<seconds> determines the displaying time, if this argument is missing the time will be two seconds. P=<PubScreenname> gives the name of the PubScreen. If this argument is missing the window will be opened on the Workbench. PubScreen names are case-sensitive, so You will have to take care of capital and small letters.

```
rx QuickMsg D=4 P=y d=2 p="Test Screen"
```

will show »D=4 P=y« for two seconds on the Workbench screen, if the PubScreen called »Test Screen« doesn't exist. The last occurrence of D=... and P=... will be evaluated and following text is ignored. No notice will be taken of lower/uppercase. The PubScreen name has to be enclosed in double quotes »"« if it contains spaces. Assigning values to D and P has to be done without spaces, so D= 2 or D = 2 or D =2 are ignored.

1.7 Example 3

```
rx QuickMsg FROM Ram:T/test
rx QuickMsg fRoM "Ram Disk:T/test"
```

will take the text to be shown and the settings for display time and the PubScreen not from the command line, but from the given filename. Both examples are pointing to the same file, if the drive RAM: is named as »Ram Disk:«. The filename has to be enclosed in double quotes »"« if it contains spaces. There must be a space between FROM (case-insensitive) and the filename.

Unlike FILE (see

Example 6

) only the file's first line is read and the rest of the command line is ignored.

1.8 Example 4

```
rx QuickMsg a\nb\n
```

To display a multiple-line string a »\n« has to be inserted at the line feed's position. The given example will show »a« in the first line, »b« in the second and an empty third line.

1.9 Example 5

```
rx QuickMsg "a"b'c'
```

As TritonRexx uses double quotes »"« for defining strings, they may not

occur in the text to be displayed. QuickMsg converts them to single quotes »'« to prevent a program abort with error 18 (invalid argument for function). So the program call given above would show »'a'b'c'«.

1.10 Examples 6 & 7

```
.key Arg1/A,Arg2
.bra {
.ket }
.def Arg2 Workbench
Info {Arg1} ALL >T:test
rx QuickMsg FILE T:test D=5 P={Arg2}
```

This example shows the batch usage with passing arguments. If Arg1 equals to RAM: and Arg2 equals to DOPUS.1 the Info command will output some informations about RAM:, redirected to T:test, and QuickMsg will display this file for 5 seconds on the DirOpus screen. FILE follows the same conditions that FROM does, see also

Example 3

In contrast with FROM the whole file is read in with FILE and arguments like D=..., M=... or P=... are read from the command line.

```
.key Arg1/A,Arg2,Arg3
.bra {
.ket }
.def Arg2 Workbench
.def Arg3 5
Echo "Current setting:*n" >t:ShowAsgnTmp{$$}
Assign {Arg1} Exists >>t:ShowAsgnTmp{$$}
rx QuickMsg FILE t:ShowAsgnTmp{$$} P={Arg2} D={Arg3}
Delete t:ShowAsgnTmp{$$} Quiet
```

Another example similar to the one before: An assigned path is passed to the script (S:, f. ex.) and for the time of {Arg3} QuickMsg shows the real physical path it is assigned to. With \$\$ the current CLI's process number is added to the redirection file name, so the resulting name might be »t:ShowAsgnTmp21«. Doing like this the outputs of several CLI's running at the same time may be differentiated.

1.11 Example 8

```
rx QuickMsg Another example M=2
```

shows »Another example« in bold text.

With M=... the text representation can be affected:

M=0 Normal representation

M=1 Highlighted text

M=2 Bold text

M=3 Text with 3D-effect

M=4 Settings for text representation will be taken from title settings of

TritonPrefs

Any other numeric value will result in normal representation. Assigning strings (instead of numbers) will display this as text, so M=abc shows »M=abc«.

1.12 Example 9

To display the keywords FROM and FILE as text they have to be preceded by an *:

```
rx QuickMsg *From now on...
rx QuickMsg *File deleted !!!
```

shows »From now on...« and »File deleted !!!« on the Workbench screen for two seconds each.

1.13 Beispiel 10

If the text output should be limited in width or in height (↔ measured in characters each) the arguments W=<number> and/or H=<number> may be given:

```
rx QuickMsg Text output - width limits W=8
rx QuickMsg Text output cutted off W=6 H=3
```

will show the following messages one after the other for 2 seconds each:

```
»Text out«
»put - wi«
»dth limi«
» ts «

»Text o«
»utput «
»cutted«
```

If the text output might produce results not to be forecasted, like with the arguments FROM (

```
Example 3
) or FILE (
Examples 6 & 7
), the enfor-
```

ced outputting to CLI may be prevented. This would happen if TritonRexx finds out at runtime that the window would be too big. Also see the second pararaph at

Hints
for that.

The first output from the given example is only limited to the width; if the height is getting too big the output would be redirected to CLI nevertheless. The second output from the example is additionally limited to the height and everything that follows is cut off when the height clipping comes in!

If both arguments are given the width is calculated first and height is calculated second.

1.14 Hints

If Triton and TritonRexx are not (correctly) installed on Your system the text will be displayed in a CLI window. Text presentation effects from

Example 8
aren't possible in this case. Also, the displayed text may look a little confusing by line feeds f. ex., so You better shouldn't use this program in this case ;-)

If TritonRexx finds out at runtime that the output window would be too big because the used screen isn't wide or high enough the text will be output to CLI also.

The program related window settings (ENV:Triton/QuickMsg.win.1) will be deleted after quitting QuickMsg. By this it's achieved to open the window centered on the screen at the next program start. Otherwise, the window would be opened on its last position.

Displaying text files (FILE in Example 6) will look much better using a fixed font like "topaz". The reason is that TritonRexx centers strings line-wise. Therefore QuickMsg justifies the length of every text line in the file to the length of the longest line to get a left boundary.

QuickMsg doesn't limit the delay time. If You set it to D=1000 the text is displayed for 16 minutes and 40 seconds! You'd better know what You are doin'...

1.15 The legal stuff

QuickMsg is Freeware. Do with it whatever You want, but You are not allowed to make profit with it. Although I have tested it on my system (A1200, KS 3.0, Bliizzard II w/ 24MB RAM) I am not responsible for GURUs or data loss caused by usage of this program - You use it at Your own risk.

1.16 Thanks

I would like to thank Stefan Zeiger for his triton.library and Jürgen Kohrmeyer for his tritonrexx.library and rexxtricks.library. Without their development this program wouldn't have been possible.

1.17 History

v1.0 26-Sep-2002 * Initial version, first public release

1.18 Author

If You use and like (or hate) this program, please send me a message. I'd like to receive bug reports, suggestions for improvement or just a "hello" at:

eMail Axel.Greve@siemens.com
 (no private eMail, so please keep the Netequette :-)

Snail mail Axel Greve
 Hochfeld 26
 D-22607 Hamburg
 Deutschland

Please send flames, bombs, etc. >NIL:

Happy & successful "QuickMessaging",
Axel

--- Let's keep the Amiga spirit alive !!! ---
