Most

Version 1.53
29 January 1994

Most is a 2.0 only textviewer which I wrote with full workbench support in mind. It supports ARexx, AmigaGuide, locale.library, XPK, AppWindows, AppIcons, Clipboard, Pipes, no limited number of windows and more. Most is GiftWare.

by Uwe Röhm

Documentation for Most V1.53
Date: 29 January 1994
Author: Uwe Röhm

Most is a 2.0 only textviewer which I wrote with full workbench support in mind. First at all an overview of interesting features:

- unlimited number of windows
- each text has its own process
- search function
- highly configurable
- clipboard support
- "click-and-load" function for includes
- font sensitive GUI
- PubScreen, CustomScreen or Workbench
- AppWindow, AppIcon
- ARexx Port
- localization with locale.library
- AmigaGuide support
- XPK support
- pipe support
- as mentioned, kickstart 2.0 is requiered

1 Copyright

Most 1.53 is published under the concept of GiftWare! So if you find Most a useful tool (and of course use it) please make me a present of any kind - and make me **very** happy.

Beside this I keep my Copyright:

You **must** get my aggreement if you wish to publish Most in a commercial product. Especially it is not allowed to use Most for military purposes of any kind which includes using Most in companies which plan or only support military products.

You are allowed to include Most in PD packages which cost less than $10.- ! But I ask you to keep the archive together. You have to include the documentation with each version. Thank you!

2 Background

There are some goals I try to achieve with Most:

- full workbench support
- easy to use with mouse **and** keyboard
- flexibility
- memory friendly :-)

So Most uses a window for **each** text - there is no possibility to use it like Commodore's More in a shell. Of course you can choose another screen to open on or even an own CustomScreen. How much windows are opened by Most and where they are is absolut without interest and only limited by the amount of free (chip-) memory!

As each text is displayed in his own window there is also its own process behind. Cause of the windows are absolut independent of each other. Especially you can send ARexx commands to different windows simultaneous and you are free to open new Most windows even if previously opened windows are "busy" - there is Most's main process which handles all texts and some global ARexx commands (see Chapter 11 [ARexx Port], page 22).

To be memory friendly and to display very huge texts even on a 500KB system Most does buffering. Small texts, which fit completly into the buffer, are read only once and closed than so that you can e.g. edit and save it again. But if the text is larger than the configurated buffer length (yes, of course you can change the buffer size, but read on :-)) Most will do some sort of "paging". So memory usage will be limited by the buffer size you specified (little memory -> small buffer). On the other hand this slowes scrolling and searching down!!! Sorry for this, but if you have enough memory just increase to buffer size to - let's say - 2 MB. Don't be afraid. Of course Most only uses the amount of memory it needs if a text is smaller than the buffer ;-)

There is one exception of the paging method: `XPK` packed files. In the current version Most does **no paging** to `XPK` files but reads the complete file into memory (see Chapter 9 [XPK support], page 20). Perhaps this will change in future, perhaps it will stay...

Oh, I nearly forgot that Most can display all `ANSI` sequences. Without this it wouldn't be a real text viewer. As I don't want to do old work again and again I decided to use the console.device. This enables you to mark lines and copy them with `Amiga C` into the clipboard! I regret that there are also some problems:

- You can only mark text inside the visible text portion.
- Scrolling is quiet annoying on an Amiga without 680x0 (x > 1).
- Don't load binary files. I warned you!!!

3 Configuration

Most does the following configuration steps on startup:

1. internal defaults
2. Environments `Most/Most` and `Most/FileReq` (When used from shell local environments will be read first!)
3.
   Shell-start: parse command line
   WB-start   : parse ToolTypes
4. NEW since Most 1.53: For each text loaded Most will read the project icon (if exists) and use the new tooltypes. You can also include all ARexx commands in the tooltypes which will be executed just after loading the new text! So you can write local ARexx scripts directly into the icons of texts. For a list of the possible commands see Chapter 11 [ARexx Port], page 22.

And now, ladys and gentlemen, here are the options to be used in 2.), 3.) and 4.):

3.1 FILES

Usage A list of filenames to be displayed (only on shell command line!). You can use any AmigaDOS wildcards.

3.2 PUBSCREEN

Usage Name of PubScreen where Most will open its windows on.

3.3 PORTNAME

Usage Custom name of global Chapter 11 [ARexx Port], page 22 of Most. Overrides the default name.

Default *MOSTPORT*

3.4 STARTUP

Usage Name of an ARexx script which will be executed every time Most opens a window. (see Chapter 11 [ARexx Port], page 22).

Remember that you can use directly all ARexx commands as tooltypes in project icons of text files (as a sort of "tooltype script").

3.5 WINDOWX

Shortcut `WX`

Usage X-coordinate of Chapter 6 [The main window], page 17

3.6 WINDOWY

Shortcut `WY`

Usage Y-coordinate of Chapter 6 [The main window], page 17. -1 for centered windows

3.7 WINDOWW

Shortcut `WW`

Usage width of windows

Default 600

3.8 WINDOWH

Shortcut `WH`

Usage height of windows

Default 400

3.9 SCREENMODE

Shortcut `SM`

Usage A valid screenmode Most will use to open its on screen with (which will be a PubScreen named `Most`). You can get a list of the avaible screenmodes of your system if you start `Prefs/ScreenMode`.

> **Attention!** The names of the screenmodes are different for most versions of the monitor files. Remember that you must specify a valid name, which means, that you must spell it exactly as in the monitor files. If in doubt start `Prefs/ScreenMode` and read the listview.

Example

```
SCREENMODE "Pal:HighRes"
```

3.10 FONT

Usage name of a font to be used. The name must include the `.font` postfix. Don't forget it! And be careful with proportional fonts.

Example

```
FONT "helvetica.font"
```

3.11 FONTHEIGHT

Usage Tells Most which font height it should use. Works only together with the Section 3.10 [FONT], page 7 option. If not specified Most uses the system default font's height.

3.12 APPICON

Usage A switch, which lets Most use an AppIcon on workbench. If you start Most with the `APPICON` option and without further file names, Most will only put its AppIcon onto workbench and stay in background. (see Chapter 8 [The AppIcon], page 19)

3.13 ICONX

Usage X-coordinate of Most’s Section 3.12 [APPICON], page 7 on workbench

3.14 ICONY

Usage Y-coordinate of Most’s Section 3.12 [APPICON], page 7 on workbench

3.15 BUFLEN

Usage Very interesting option to justify Most’s buffer size. In general: On small systems use small values, on larger systems try higher buffer sizes. `BUFLEN` defaults to 50 KB (which is quiet small :-)). Texts smaller than the buffer size will be read complete into memory and use only the necessary amount of memory.

Default 50 000 Bytes

3.16 PIPELEN

Usage Most can also read out of a pipe. Normally it would allocate a buffer of Section 3.15 [BUFLEN], page 8/10 Bytes for reading. With this option you can justify the buffer length for pipes by hand.

Example

```
PIPELEN 8192
```

3.17 QUOTES

Usage You can specify a sequence of quote characters. These characters will be used for the “click-and-load” option: If you doubleclick on a word which stands **between** a quote character Most will try to load a text named like this word. This is very useful for programmers as you can view include files very easy!

Default "’<>(){}[]

3.18 DIRS

Usage Specify one or more directories Most will search through if you use the "click-and-load" function (see Section 3.17 [QUOTES], page 8). All directories must be seperated by `|`.

Example

```
DIRS "Include:|TexInputs:|LC:Include/"
```

3.19 ENDS

Usage
Postfixes to be appended on include files with the “click-and-load” function in the given order (see Section 3.18 [DIRS], page 9, see Section 3.17 [QUOTES], page 8).

Example

```
ENDS ".c.h.tex"
```

3.20 STACKSIZE

Usage Adjust the size of the editor's stack. Useful for example for the emacs ("Eight Megabytes And Constantly Swapping"?)...

Example

```
STACKSIZE 50000
```

Default 8192 bytes

4 Keyboard commands

In the current version Most supports all (and more!) keyboard shortcuts well known from More. Future versions will allow custom keyboard commands...

4.1 Scrolling

one line forwards
: `CURSOR DOWN NUMERICPAD 2 RETURN ENTER`

one line backwards
: `CURSOR UP NUMERICPAD 8 DELETE`

one page forwards
: `SHIFT CURSOR DOWN SPACE PAGE DOWN`

one page backwards
: `SHIFT CURSOR UP BACKSPACE PAGE UP b`

Jump to beginning of text
: `HOME <`

Jump to end of text
: `END >`

oen column to the right
: `CURSOR RIGHT NUMERICPAD 6`

one column to the left
: `CURSOR LEFT NUMERICPAD 4`

8 columns to the right
: `SHIFT CURSOR RIGHT`

8 columns to the left
: `SHIFT CURSOR LEFT`

4.2 Searching

search actual searchstring again
: `n`

ignore upper/lower case
: `.`

searching with distinction of upper/lower case

`/`

Open the search requester

`f s`

4.3 Open file

Load new text with `l` (a filerequester shows up).

4.4 Info requester

Beside the usual Chapter 1 [Copyright], page 2 you find here some information about the current text: Its filename, the filelength, the number of lines and the visual number of lines, and also the name of the `XPK` packer (or just "—-" if the file isn't packed). At last there's a message indicating that the current text has been read completly into memory or – if not – the length of the used read buffer.

The Info requester can be opened with `h`.

4.5 Help

If you press `HELP`, the AmigaGuide will be started with the documentation of Most. Of course this requires the AmigaGuide be installed correctly. If Most can't open the amigaguide.library it will open the Section 4.4 [Info requester], page 13.

AmigaGuide searches its documents in the current directory and also in the paths listed in the environment variable *AmigaGuide/Path*.

4.6 Edit file

If you press `e`, the current text will be loaded into an editor. The editor will be started in the background. The namen of the editors must be configurated in the environment variable *EDITOR*.

4.7 Quit file

Close the text window with `ESCAPE`, `CTRL-C` or `q`.

5 Menus

There are the following menus available:

5.1 Project

`New...` Opens a new window and brings up a filerequester where you can choose a text file to be displayed in the new window. If you cancel the filerequester the window will be closed. The position and dimensions of the filerequester can be stored in Section 5.5 [Settings], page 16.

`Open...` Asks you for a new file to be loaded into the current window.

`Edit...` Starts the editor with the current text via RUN. The name of the editor can be configurated in ENV:Editor (see Section 11.11 [Edit], page 25).

`Print` Simply prints the current text to PRT:.

`Info...` Pops up the Section 4.4 [Info requester], page 13 where you find some useful information to the current text like name, length, line count, visible lines, `XPK` packername and a short comment if the text has been loaded completely into memory or if it uses an buffer.

`Quit` Quits the current window (and if it was the last one Most at all)

5.2 Move

Move through the current text as the menus describe.

With the menu “Jump to line...” you can specify a certain line number in a requester where you will jump to.

5.3 Menu Search

`searching...`
Chapter 7 [The search requester], page 18 will be opened with its stringgadget activated. If the search requester is open this menu is disabled.

`forwards/backwards`
: Search string forwards/backwards (only with open search requester)

5.4 Extras

`Refresh page`
: Reset console and redraw page

`Set tabulator width...`
: Enter new numeric value for the tabulator width (per default every 8th character there's a tab).

`Execute ARexx Macro...`
: Asks for an ARexx Macro to be executed (see Chapter 11 [ARexx Port], page 22).

5.5 Settings

`save`

`global Prefs`
: Save actual settings in two environments in *ENV:* and *ENVARC:*. A subdirectory `Most` will be created automatically. All known options from workbench or shell start are saved and also the dimensions and position and the pattern string of the file requester.

`local Prefs`
: Does the same as above, but uses local shell variables. This only works if you started Most from a Shell!

6 The main window

Title
In the title the filename and on the right hand side the current column and line numbers are displayed. Format:

```
          <filename>                                            <column>/<line>
```

Gadgets
Beside the system gadgets there are three further ones which control text scrolling:

You can use the scroller continously by click-and-draging or for scrolling one page forwards/backwards if you click under/above the knob. The knob's position represents the current text position, its height the relation between the visible and the total text length. Use the arrows to scroll one line in the correspondending direction.

Mouse
"Click-And-Load" function.

If you doubleclick into the window Most will look at this position to find a word which is limited by one of the configurated quote characters (see Section 3.17 [QUOTES], page 8). This word (can include spaces!!) will be treated as a filename for which Most searches first in the current directory and than in each further configurated (see Section 3.18 [DIRS], page 9) directory. Most will also try the plain filename and than (if not found) append each prefix given with the Section 3.19 [ENDS], page 10 option. The first found file will be displayed in a new window. If no name matches, the screen will flash (DisplayBeep).

Example:

> Doubleclick on "#include <exec/exec.h>" between "<" and ">".
> the file `Include:Exec/Exec.h` will be loaded, if you
> specifed "DIRS Include:" on startup.

Clipboard

You can easy mark any text in the visible text by click-and-dragging (drag without releasing the mouse button). Copy the marked block with `Amiga-C` into the clipboard. The current version doesn't support paste from the clipboard.

AppWindow

Take any "real" icon (no AppIcon) on workbench and drop it in Most's window. It will be loaded into the current window. If you drop more than one icons Most will open further windows.

7 The search requester

Here you can enter a search string and also manipulate the three search option. Therefor you can use the mouse and/or the keyboard shortcuts indicated by the underlined characters. Additionaly you have the same functionality as in the main window! So no annoying re-activation of Chapter 6 [The main window], page 17.

- All menus are avaible as in the text window. Only the Section 5.3 [Menu Search], page 15 will be disabled and instead “search forwards” and “search backwards” are unghosted.
- Beside the mentioned Requester shortcuts you can use **all** normal Chapter 4 [Keyboard commands], page 12 like scrolling with the cursor keys!!! (Be sure the string gadget isn’t activated)

8 The AppIcon

The AppIcon has two functions

`Doubleclick`
: A filerequester will come up where you can choose a file to be loaded and displayed. If you select “Cancel” Most will show a security requester where you can quit Most.

`Icons`
: Drag-and-drop one or more icons on the AppIcon and they will be loaded.

Comments:

> Most uses his own program icon as its Section 3.12 [APPICON], page 7. Perhaps I will support custom icons in the future but not yet ;-) The coordinates of the AppIcon can be saved in the menu Section 5.5 [Settings], page 16. Use the options Section 3.13 [ICONX], page 8 and Section 3.14 [ICONY], page 8 for exact placement of the AppIcon.

9 XPK support

Main goal of Most’s XPK support is to hide any consequences of packed data from the user. This means, that any packed file (with XPK) will automatically be unpacked from Most if the XPK packages is correct installed. The only hint you get is in the information requester, where the line “XPK-packer:” will show you the used packer (usually you see “—-” there). There is only one thing special with XPK packed data: The file will be loaded completely into memory and no buffering will be done!

If Most finds an encrypted files (with XPK of course), it opens a password requester and prompts you for the password. Be careful, because this is your only chance! If you give the wrong answer, you will see the encrypted file and have to reload it.

On the other hand if you don’t have XPK you can use Most without any difference. The xpkmaster.library is not required by Most to run at all. Of course in this case XPK packed data will not be unpacked :-)

10 Pipe support

As a special feature Most 1.42 supports pipes. This means, that you can type something like “type foo | Most” at your (W)Shell prompt and Most will show the output of the first type command in its window!

As I said this feature is brand new. So I regret that I couldn’t test it for all possibilities. Please report me any failure.

There are two small problems:

1. `XPK` doesn’t work with pipes.
2. Second there is the problem with pipes that you cannot know how much data therein will be... So one effect is, that you cannot scroll back to the start of an file if you scrolled over the bounds of a buffer and the other one is that the size of the buffer will be (maximum Section 3.15 [BUFLEN], page 8 / 10) every time (you can set the pipe’s buffer size with the Section 3.16 [PIPELEN], page 8 option). If the pipe’s data fits completly into the buffer you have no restrictions for scrolling. But if the data is to large - Sorry.

11 ARexx Port

Beside the global ARexx port *MOSTPORT* (or the name you gave it with the option Section 3.3 [PORTNAME], page 5 on startup) of Most's main process each text window has his own private ARexx port. To select a window first send the command Section 11.1 [GetFileList], page 22 to the global port. You will receive a list of the avaible windows and its portnames.

All following ARexx commands can be send to a private window port, but Most's main port does only understand the first 6 commands Section 11.1 [GetFileList], page 22, Section 11.2 [GetScreenName], page 23, Section 11.3 [Quit], page 23, Section 11.6 [OpenNew], page 24, Section 11.5 [Set], page 24 and Section 11.4 [Request], page 23.

In `RC` Most will return an errorcode of

- 0 for no errors
- 5 if it couln'd open a new text with Section 11.7 [Open], page 24 or Section 11.6 [OpenNew], page 24
- 10 if the parameters for a command were wrong
- 20 for an unknown ARexx command

All other commands will return an errorcode of 20 if send to *MOSTPORT*. Sorry, but the ARexx port isn't complete today. But I'm working on it :-)

The following ARexx commands are supported:

11.1 GetFileList

Template GetFileList

Result In *RESULT* you get a list of all available text windows and their port names. *RESULT* looks like this: `{`<window's name> <port name> `}^+`

Example

```
GetFileList
RESULT: "Most.dok MOST.0 Most.doc MOST.1"
```

11.2 GetScreenName

Template GetScreenName

Result *RESULT* contains screen name.

Usage Get the name of the screen, where the current window is on.

Example

```
GetScreenName
RESULT - "Workbench"
```

11.3 Quit

Template Quit

Usage Quits Most and closes **all** windows. Chapter 8 [The AppIcon], page 19 of Most will be deleted – if there is one at all.

Example

```
Quit
```

11.4 Request

Template Request `TITLE/K/A TEXT/K/A GADGETS/K/A`

Resultat *RESULT* contains number of selected gadget:

0	the most right gadget (should be “Cancel”)
1 bis n-1	the other gadgets from left to right

Usage A simple method of asking the user silly questions via requester. Title and text are simple strings. With `GADGETS` you specify the texts of the gadgets (and how much gadgets there are), seperated with '`|`'.

Example

```
Request TITLE "Test" TEXT "All OK?" GADGETS "YES|No"
RESULT - 1 if YES has been selected, else 0
```

11.5 Set

Template Set `CASE/S` `WORD/S` `BACK/S`

Usage Set some internal parameters like

`CASE` ignore upper/lower case on/off

`WORD` search only for whole words on/off

`BACK` search backwards on/off

All parameters will be set if specified, and disabled otherwise.

Example

```
Set CASE ON
Enables casesensitiv searching, but disables the other two options.
```

11.6 OpenNew

Template OpenNew `FILE/K`

Usage Load a new text in a new window. The filename is optional. If you don't specify a name a filerequester opens.

You don't need to select a current window before. This command doesn't return until the new window is closed again!

Example

```
OpenNew FILE Most.dok
```

11.7 Open

Template Open `FILE/K`

Usage Load a new text in the current window. The filename is optional. If you don't specify a name a filerequester opens. This command doesn't return until the window is closed again!

Example

```
Open FILE Most.dok
```

11.8 Close

Template Close

Usage Closes the current window.

11.9 ToFront

Template ToFront

Usage Makes the current visible: It will be made the frontmost window and the screen it's on will be activated and go to front.

11.10 GetWindowDims

Template GetWindowDims

Result In *RESULT* stands `<xpos> <ypos> <breite> <höhe>`

Usage Get the position and dimensions of the current window in *RESULT*

Example

```
GetWindowDims
RESULT - "40 40 600 180"
```

11.11 Edit

Template Edit

Usage Section 11.11 [Edit], page 25 the current text (runs editor).

11.12 Info

Template Info

Usage Opens the Section 4.4 [Info requester], page 13 to the current window.

11.13 FirstPage

Template FirstPage

Usage Show the first page of the current text.

11.14 LastPage

Template LastPage

Usage Show the last page of the current text.

11.15 NextPage

Template NextPage

Usage Show the next page of the current text.

11.16 PrevPage

Template PrevPage

Usage Show the previous page of the current text.

11.17 NextLine

Template NextLine

Usage Scroll up one line.

11.18 PrevLine

Template PrevLine

Usage Scroll down one line.

11.19 NextColumn

Template NextColumn

Usage Scroll left one column.

11.20 PrevColumn

Template PrevColumn

Usage Scroll right one column.

11.21 NextTab

Template NextTab

Usage Scroll left 8 columns.

11.22 PrevTab

Template PrevTab

Usage Scroll right 8 columns.

11.23 Goto

Template Goto `LINE/K/N/A COLUMN/K/N`

Usage Jump to the specified line. Optional you can specify a new first column.

Example

```
Goto LINE 142
```

11.24 Search

Template Search `TEXT/K`

Usage Without argument: Chapter 7 [The search requester], page 18 opens or if allready open the current string will be search again.

With searchstring: Search for the given string.

Example

```
Search TEXT foobar
```

12 Known bugs

- Several control sequences for cursor movements trash the display.
- Most has problems with the pipe of WShell.

13 Ideas - Bug reports

As mentioned before Most isn't complete today. There are some points I will improve in the future. I also have Commodore's Style Guide in mind and will follow its rules the next release better.

I'm absolut interested in any kind of responds - ideas, enhancements requests, criticism or bug reports. Call me, write me, mail me. I will hear you!

My address:

Uwe Roehm
Woerthstr. 18
D - 94032 Passau
Germany

EMail:

bugs@roemer.adsp.sub.org

Fido:

2:2494/22.5

14 Current version

You can download the current version of Most from `ftp.uni-passau.de` (132.231.1.10) in the directory `pub/amiga/local/UweR`. You will also find other interesting software in `pub/amiga/local`...

15 Greetings

I want to thank my beta testers

Karl, obviously never sleeping
David, Uli, Holger, Stefan and Stefan ;-)

A special thanks to Urban for his sugesstions and `XPK`. And a great thank you to everyone who wrote me bug reports, translations or suggestions.

Index

Table of Contents