in

COLLABORATORS

	TITLE : in		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

in

1.1 Table of Contents

Textread release 1.13
Version 38.82

User's guide

Released 3 August 1995

If you don't wish to read the whole document, just hava a quick browse in the History section, to see what have changed since previous releases.

1996-07-06

This release is called 1.14, but it is not finished. Because I have failed - big time - to release new upgrades, I have decided to release the source code in the hope that somebody is willing to take over Textread. I still consider this program to be very good, I use it every day myself. Sure, some things are lacking, but the fact is that it has everything I need. That, and the fact that it is written completly in assembler, are the reasons why I don't have any motivation to continue to improve it.

So instead of letting more time pass, my hope is that somebody out there can clean up the source, fix the bugs and release a new version. Or perhaps just keep the good parts and begin from scratch (may I suggest another language than assembler?).

To those who have sent be bugreports, ideas and comments and especially those of you who have paid the shareware fee... What can I say? I'm sorry. If is wasn't for you I would have stopped developing this program years ago.

A big "thank you!" must also go to Olivier ANH for the frensh translation of the documentation. Unfortunately, I lost your address and have been unable to contact you. I hope your work was not in vain, but can be used in a futute version of Textread.

Have a wonderful life, and may God bless you all.
Martin Blom

PS. If somebody decides to continue the program, would you be so kind to contact me first? Thank you! DS.

1.2 Preface

Preface
~~~~~~~

The programs in this package are copyrighted by Martin Blom, but freely distributable as long as only a nominal fee is charged for copying and distribution.

If you, the user, finds this program worth using, please consider donating a small amount of money to the authur, see the Shareware? section.

Textread must be distributed in this archive in unmodified form. Nothing may be added, or removed from it. I take no responsibility for any trouble Textread may cause. Use it at your own risk. If you, for some strange reason, wish to include Textread in your own software package, please contact me first:

Martin Blom
Alsättersgatan 15A:24
S-584 35 Linköping
SWEDEN

E-mail:

lcs@lysator.liu.se

1.3 Introduction

Introduction
~~~~~~~~~~~~

The first official release of Textread, 1.03, was released in November

1992. I wrote it for two reasons:
 1) I wanted to learn how to code system friendly programs under Release 2
 2) I was not satisfied with the speed of the current more clones.

Speed was the most important issue, memory had lower priority. This is still the case, even if I now have added system-friendly routines to my own hardware-banging blitter and direct screen memory accessing text routines, and have optimized memory usage.

Release 2.04 (V37) is required, and the program was designed with "Amiga User Interface Style Guide" beside the keyboard, so I think you will feel comfortable with the layout. In addition, Textread will take advantage of asl.library V38, XPK and powerpacker.library if present.

1.4 Is Textread Shareware?

```
                          Is Textread Shareware?
                          ~~~~~~~~~~~~~~~~~~~~~~
```

The first release of Textread was Freeware. As time went by, I found my self sitting in front of Trash'm-One, torturing myself coding this boring program. And I figured that if somebody else will enjoy my work, he or she might find it worth some money. So I made release 1.10 Shareware.

However, I have never liked Shareware, especially when it comes to shitty programs like Textread. So beginning with release 1.13, I do not demand a donation. But if you use it, and wish to see futher updates, please consider paying.

The suggested donation is 40 SEK, 20 DM or US$ 10. If you know anything about Sweden, you know that 20 DM is about 100 SEK. So if you want to save money, send SEK. My address can be found in the Preface section.

1.5 Usage

```
                                  Usage
                                  ~~~~~
```

```
 The following files should be included in this package:

     Textread (dir)
          fonts (dir)
               IBM (dir)
                 11                              8
            IBM.font
          ENVARC (dir)
               Textread (dir)
                 Textread.PC.prefs               Textread.PC.prefs.info
                 Textread.prefs                  Textread.prefs.info
       fonts.info                      Textread
       Textread.doc                    Textread.doc.info
```

```
       Textread.guide                   Textread.guide.info
       TextRead.info
  FILE_ID.DIZ                      Textread.info
  Textread.readme
```

The size of the executable file is 47632 bytes.

There is no installation script, just copy the files where they belong/where you want them. The IBM font is only provided as an example for the Textread.PC.prefs pref-file, and is not required.

Textread can be started either from Shell or from Workbench.

1.6 Shell

Starting Textread from CLI/Shell

Type Textread ? to get the standard AmigaDOS template:

```
FILES/M,TABS/K/N,UNIT/K/N,SETTINGS/K,NEW/S:
```

If you give wrong arguments, you will get this text:

```
Textread 1.13 ©1992-1995 Martin Blom.
Usage: Textread [[FILES] <files|dirs>] [TABS <n>] [UNIT <n>]
                [SETTINGS <name>] [NEW]
```

You may supply any number of files or paths as arguments to Textread, a new buffer will be created for each of them. If a path was given instead of a file name, or the file didn't exist, a file requester will appear.

The other arguments are options;

TABS <n>
Selects the tab size (on screen, not printer).

UNIT <n>
Selects the unit of printer.device to use when printing. Useful if you've got a printer.device that supports multiple printers.

SETTINGS <name>
Selects a settings file to use. Textread first looks for this keyword, then in 'ENVARC:Textread/Textread.prefs'. If neither of these files could be opened, the build-in defaults will be used.

NEW
If a copy of Textread was already loaded before, it will be told to open a new buffer with the new files in order to use as little memory as possible. The second copy will then only keep a small code segment in memory, waiting for the buffer to close. Specify this keyword to force a new copy of Textread to be loaded.

Here are some examples:

```
1> Textread Textread.doc TABS 3
```

```
1> Textread TABS 3 FILE Textread.doc
```

These two command do the same thing, they load Textread.doc and show it with tab size set to 3.

```
1> Textread TABS 8 UNIT 1 SETTINGS ENV:Textread.prefs
```

This command sets the tab size to 8, selects the second parallel or serial port for printing and uses the settings file found i ENV:. You will be prompted for a file to view.

```
1> Textread dh1:Docs/cars.txt dh1:Docs
```

First loads dh1:Docs/cars.txt, then creates a new buffer and opens a requester showing the files in dh1:Docs

1.7 Workbench

Starting Textread from Workbench

Couldn’t be simplier. Just double-click on its icon, and you will be prompted for a file name. You can also select Textread’s icon and then doubleclick on another icon to view that file (or, if it was a settings file, to use that settings). If you doubleclicked on a drawer or disk, the file requester will appear and list the files in that directory.

Another way to start Textread is to set the icon’s default tool to ’Textread’. Just select an icon and chose Information from the Workbench Icons menu.

The icon parsing is quite flexible. Suppose you can see Textread, two different prefs files, and a couple of text files on you Workbench. Assume the prefs files are called Textread.PC.prefs and Textread.Amiga.prefs, and you have a couple of MS-DOS text files. You want to show 2 ’normal’ text files and 1 MS-DOS file. Here is what to do: Click one on Textread, press shift and hold it down while pressing Textread.PC.prefs, <MS-DOS text file>, Textread.Amiga.prefs, <first Amiga text file>, and doubleclick on the <second Amiga file>. Textread will first load the PC prefs file, and load the PC text file. After that, a new buffer will be created using your normal prefs, and the first Amiga text will be loaded into this buffer. The last file will be loaded in a new buffer using the same prefs as the previous file. Well... just try it!

Here are the tooltypes supported:

```
 (FILETYPE=PREFS
   Tells Textread that this is a settings file.)

  TABS=<n>
   Selects the tab size (on screen, not printer).

  UNIT=<n>
   Selects the unit of printer.device to use when printing.  Useful if
   you’ve got a printer.device that supports multiple printers.
```

SETTINGS=<name>
Selects a settings file to use. Textread first looks for this keyword, then in 'ENVARC:Textread/Textread.prefs'. If neither of these files could be opened, the build-in defaults will be used.

NEW
If a copy of Textread was already loaded before, it will be told to open a new buffer with the new files in order to use as little memory as possible. The second copy will then only keep a small code segment in memory, waiting for the buffer to close. Specify this keyword to force a new copy of Textread to be loaded.

1.8 Control

Control
~~~~~~~

Once you have started Textread, you have several options via the menus. Below is a full description of each menu. If you press »Help«, you'll get a summary of all the keyboard commands. Press a key to view the text file again.

At the bottom of the screen you will notice a button, an info box and a scroll bar.

The small button labelled Q will quit Textread. The info box shows current line number, percentage and the status of some of the checkmarks in the menu (<W>rap words, Remove <A>NSIs, Remove C<R>s, Don't <F>ake font) The scroll bar is used to scroll the screen (!). And the arrows... guess!

When you have loaded a text, you can edit it with your favourite text editor by pressing <Shift-E>. The editor specified in ENV:Editor will be run and the text will be loaded. However, only the file name is passed to the editor, so if the text was crunched, you editor must be able to load such files.

You can also set and jump to up to ten bookmarks. Set them with <Shift> and a function key, and jump with just a function key. The bookmarks will be cleared when a new file is loaded (if you don't open a new buffer, that is).

Except for the Print options window, all requesters have keyboard short-cuts. They are marked as usual. Escape cancels a requester, and Return accepts. In some requesters the cursor keys (unshifted or shifted) works too. In the Find requester, you can deactivate the string gadget by pressing Shift-Return. Reactivate by pressing the underlined character. Pressing Return in the string gadget starts the search.

Textread has three menus:

Project Menu
Settings Menu
Commands Menu
~~~~~~~

1.9 Project Menu

Project menu

New

Opens a new buffer.

Open...

Will bring up a file requester and ask for a file name. At the moment, multiselect is not possible.

Open new...

This is a combination of 'New' and 'Open...'.

Save...

Will save the file back to disk after confirmation.

Save As...

Will bring up a file requester and ask for a file to save the file as.

Print

Will send your text to the printer using your settings (see below). Unfortunately, this will lock up Textread until the printing is finished. This is unacceptable and I will fix this in a later version. Make sure that 'printer.device' and either 'serial.device' or 'parallel.device' is in DEVS: and you Printer preferences are correct (especially the number of lines). If your (parallel) printer isn't on-line, this command will return immediately.

Print as...

Will open a window with several gadgets. See separate section, Print options.

Reveal

Shows a listview with the file names of all open buffers. Select one, and that buffer will be displayed.

Close

Closes the current buffer, and frees all memory used. If the buffer in question was the last one, Textread will terminate.

Iconify

Will close the screen, open an icon in the Workbench window and put Textread to sleep. When you doubleclick on the icon, the screen will reopen. Starting a second copy of Textread without the 'NEW' keyword will uniconify Textread.

Workbench »
 Open

Tries to open the Workbench screen.

 Close

Tries to close the Workbench screen.

About...

Will open a requester showing the current version, a copyright message and the current text file's name and length in bytes/charters.

Quit Textread

Quits Textread, WITHOUT giving you a second chance to abort.

1.10 Settings Menu

Settings menu

Wrap words?

Turns wordwrap on or off. If on, no words will be cut at the end of each line. This should usually be on, except when reading C sources, maybe...

Use Sysfont?

If set, your preferred screen font will be used to render menus and requesters. If not set, you preferred fixed width font will be used.

Auto-remove ANSIs?

If selected, ANSI codes will always be removed from the document when a new file is loaded.

Auto-remove CRs?

If selected, all return codes will always be removed from the document when a new file is loaded. This is handy when viewing MS-DOS text files.

Don't 'fake' font?

If selected, Textread will not display some (normally invisible) ascii codes as inverted letters. This is handy when viewing MS-DOS text files containing national characters, like å, ä, ö, ü, æ...

System rendering?

If selected, all screen rendering will be done by system calls rather that hardware-banging blitting and direct screen modifications. This is a lot slower on Amiga native bitmap screens, but allows

Textread to run on a foreign (graphic card) screens. It will also let you use any fixed width font, not just a 8$\times$8 pixel font. "Don’t ’fake’ font?" will be checked and disabled when this switch is on.

Check fonts?

Textread’s font requester will open every font to check if it is usable if this switch is selected. Only fonts that can be used will be shown. By turning this switch off, this check will be done after a font has been chosed (and only for that font) which can speed up the font requester several times, especially if some kind of font cache program is running (This is how Textread 1.03

Tab size...

Lets you select a new tab size for the screen (not the printer). The maximum value is 15, but this is far to high. 8 is default and looks often good, especially when you’re viewing assembler sources, but a lower tab size is better when looking at C sources (3 is a good choice!).

Screen mode...

Brings up a requester, asking for a screen mode. Please note that the mode must be an Amiga mode, not smaller than 640*200, and not bigger than 2040*2040. (It might work with some graphics cards, as long the card has a good emulation or a planar mode that looks just like the Amiga standard modes.)

Screen colours »

Select screen colours...

Starts the program ’SYS:Tools/Colors’, which lets you change Textread’s screen colours.

Default colours

Restores the colours to their default.

Text font »

Select text font...

Lets you select a new font to be used. However, only a 8*8 pixel font will be accepted. This is because I prefer speed in front of cool fonts.

Default font

Selects the System default font, specified in Font preferences. If no font is suitable, the ROM Topaz font will be used.

Load settings...

Lets you chose a settings file to use. This means that you can save the preferred tab size, screenmode and so on to disk, and then chose between different configurations. When Textread is started, it first looks in the icon's tool type array for a name of an settings file or, if started from the Shell, it searches the arguments for the file. If not found, it tries to load the file 'Textread.prefs' from the same directory as Textread is in. If this file isn't present the build-in settings will be used.

Save settings

Saves your current setting using the same name the setting file was loaded as. If no setting file was previously loaded, you settings will be saved as 'Textread.prefs' in the Textread directory.

Save settings as...

Saves your current settings using the name you type or select.

Create icons?

Selects if icons will be saved together with your settings files. The icons is useful if you have different settings. To use a setting, just doubleclick on that settings file and Textread will be run using the wanted settings. However, if you only runs Textread from the Shell, the icons could be waste of diskspace. If so, simply turn this option off.

Copy to all buffers

This will copy the current configuration to all open buffers.

1.11 Commands Menu

Commands menu

Find...

Asks for a string to be searched for, and then searches from current position. If found, the screen will be positioned so the top line contains the string (if it not was too close to the end of the file). If nothing was found, the screen will flash.

If 'Wildcards' in the requester is checked, you may use all Amiga wildcards as a pattern. BEWARE! If wildcards is turned on, searching is MUCH slower!

'Direction' selects if the search should start forwards or backwards.

'Case sensitive' selects how the string entered will be treated. If on, the search routines will look for an exact match of the entered string. But if it's off it doesn't matter what case the string was entered in. That means that if you want to find the word 'assembler', Textread will find 'assembler', 'ASSEMBLER', 'Assembler'... Textread's search routine (non-wildcard) can handle all national charters if you have 'local.library' loaded (WB 2.1). The wildcard search is handled by AmigaOS and is of course localized.

Find next

Searches forwards in text for the string entered in the 'Find...' requester.

Find previous

Searches backwards in text for the string entered in the 'Find...' requester.

Go to line...

Brings up a requester and asks for a line number to jump to.

Go to percent...

Brings up a requester and asks for a percent number to jump to.

Edit text

Starts the editor that the environmental variable $EDITOR points to, given the file name with full path as the only argument.

Remove ANSIs

Will remove ANSI codes from the document.

Remove CRs

Removes all Return codes from the document. This is handy when viewing MS-DOS text files, since MS-DOS uses both Line Feed (LF, ascii #10) and Carriage Return (CR, ascii 13) as an end-of-line mark.

1.12 Print options

Print options
~~~~~~~~~~~~~

When you select 'Print as...' from the Project menu, a window with some gadgets will open. This requester has not been updated from 1.03, nor have the printer routines. I hope to clean this up in the next (?) version. (And so I've said a couple of releases now... :( )

The field buttons
~~~~~~~~~~~~~

Used to control the header of each page. The left filed may contain the name of the file or a title, max 30 charters long. The middle and right filed may contain todays date or page number. If you don’t want any header, simply set all fields to ’Nothing’. If the header is on, you loose 2 lines on each page.

Header style

Selects how the header will be printed. Bold, underlined, or italic.

Title

Max 30 characters of text. See above.

Tab size

Selects the tab size for the printer. Note that all tabs are converted to spaces before printed, so you printer doesn’t need to support different tab sizes.

Wrap words

Selects wordwrap on or off. See Settings menu, ’Wrap words?’ above.

Print

Starts the printing. See Project menu, ’Print’.

Use

Closes the window.

Cancel

Restores the options and closes the window.

1.13 What’s missing?

```
                          What’s missing?
                          ~~~~~~~~~~~~~~~
```

ARexx interface. Don’t cross your fingers.
Localisation. I’m looking into it.
Multithreaded printing, new ’Print as...’ requester. Yea right.
Multiselect in the file requester. An easy one.

I’d also like to see some nice icons. If you have the talent, don’t hesitate to send me some!

And anything YOU want (well, at least I’ll consider it). Comments are welcome. See the Preface section for my address.

1.14 Credits and Acknowledgments

Credits and Acknowledgments
~~~~~~~~~~~~~~~~~~~~~~~~~~~

Textread was programmed all in assembler. Now when I have begun to understand that C thing, I whish I had learned it years ago.

In release 1.10-1.11, the requesters were designed using GadToolBox 2.0c by Jan van den Baard.

From 1.13 and up, the requesters were designed using Designer by Ian OConnor.

Thanks to Nicolas Dade, Carsten Fulde, Che Lalic, Stefan Virebrand, Stefan Stammler, Pontus Munck, Sven Thomas, Stefan Sommerfeld, Mathias Karlsson, Ulrich Sibiller, Dave Dustin and Excel for your help and support!

Special thanks to Stefan Stammler, Mathias Karlsson, Dave Dustin, Jan Thurfors and Bovier Philippe for paying the shareware fee. Yes, it's true: only five have payed. Still, that's an increase of more than 100% since the last release..!

/ Martin Blom

This AmigaGuide document was made by Sven Thomas and Martin Blom.

1.15 Revision History

Revision History
~~~~~~~~~~~~~~~~

```
Release 1.00
------------
        V37.0           First release. (12.9.92)

        V37.01          Minor bugfix. (12.9.92)

        V37.10          Finally  got  the  docs  to  Powerpacker.library!
                        Thanks, David of Iceland!  (12.9.92)

        V37.11          Now accepts a directory instead of a file from the
                        Shell too.

                        When you load a new file, the textposition will be
                        reset the top of file.

                        Screen  will  now  flash if nothing was found when
                        searching.

                        Fixed a bug in the screen print routine.
```

```
                        This ^ led to another bug.  Fixed!  (13.9.92)

Release 1.01
------------
        V37.13          Silly bug removed (division by zero).

                        Another silly bug removed in print options window.
                        Thanks, Mungwall...  (18.9.92)

        V37.14          Decreased the size of settings file.

                        Added arrowgadgets to the scrollbar.

                        Fixed the last (!?) bug...  (20.9.92)

Release 1.02
------------
        V37.15          LAME  bug removed:  IntuiTicks forced the Helppage
                        to disappear!

                        Textread  now checks if your (parallel) printer is
                        online before it tries to print.

                        The  number of lines won't be recalculeated if the
                        user cancels the Tab size-requester.

                        Added bookmarks on F1-F10.  (24.10.92)


Release 1.03 (first public release)
------------
        V37.16          Textread  was  kind of slow when you looked at the
                        end of a 600 kB text.  Fixed...

                        The  number of lines won't be recalculeated if the
                        user  doesn't  change  the tabsize in the Tabsize-
                        requester.

                        <Alt> and cursor up/down moves 100 lines.

                        Fixed bug:  Strange textoutput if big overscan was
                        used.

                        Removed   the   annoying   'copyright'   requester.
                        (7.11.92)

Release 1.1
-----------
        V38.63          (NOTE!  Some things were changed more than once!)

                        MAJOR  code  cleenup.  Result:  shorter and faster
                        (?) code.

                        Some errormessages added.  But more are needed.

                        Now    uses    a    ScreenMode-requester,    so    any
```

bitplanemode is supported. Also rewrote the blit routines to use the ECS-blitter if available.

The icon now shows the filename when Textread is iconified.

Removed the "Verify quit?" menu item.

Now supports XPK libraries. But it still works with powerpacker!

Some cleanup in the menus.

Some minor bigfixes.

Removed the lame helptexts. Mail me if you want them back!

Now shows the status of some checkmarks at the bottom of the screen.

Added the posibillity to remove (legal) ANSI sequences.

NewLook-Menus on OS 3.0+.

Textread doesn't NEED diskfont.library anymore. But the fontrequester will not work, if diskfont isn't present.

Added a hook to the fontrequester, that removes all fonts that are not 8*8 pixels. This may slow down the requester a bit, but only the first time.

Textread now uses rtLockWindow() when the Print Options window is open.

Added possibility not to 'fake' the font, ie. chars $0 to $1f and $80 to $9f will be printed as $0 to $1f and $80 to $9f, and not as inverted $40 to $5f and $c0 to $df. Hmm, well... 8 bit MS-DOS textfiles will now look as they are supposed to do. If you have the right font.

Added the posibillity to remove CR characters. They are used in MS-DOS textfiles togeather with LF as a line terminator. The Amiga only uses LF. This will make MS-DOS textfiles look better.

Removed an Enforcer read hit.

Fixed some argument parsing bugs.

No more flicker in the line number gadget! Real improvement, eh?

Added am AppIcon to the Workbench, and removed it again. :)

Replaced my blitter routines with BltBitMap for compabillity (HA!). The scrolling is slower, but you won’t notice. Trust me.

I FINALLY got my A4000/040... 2 month delayed! Many enforcer hits removed...

Textread doesn’t need neither the powerpacker nor xpkmaster libraries. But if found, they will be used (XPK, pp, dos, in that order.).

Temporary (?) removed the colour requester. Did anybody used thay thing? It was an easy way to make Textread support AGA colours... :)

Made the settings routines work (quick’n dirty). The format is NOT compatible with old prefs files, and NO version check is made, so please delete all old prefs... Sorry. The format will change in next release again.

The ANSI-remover can now also handle the BS character. (For some UNIX man files.)

Reduced flicker when using the scrollbar. Improved the linecounting routine a bit, hopefully reducing memory fragmentation.

Replaced Reqtools with ASL. Why? I don’t know... The binary file size increased by almost 10kB!! :(

Added wildcards to the search routines. Uses AmigaDOS’ MatchPattern and MatchPatternNoCase, so they are SLOW!

Added a tiny Quit button at the bottom of the screen.

Added finescroll. Yes, I know... The implementation sucks. I might try to fix it some day, but I don’t know how I can improve it without adding double buffering..?

Added multibuffer support. You can now have any number of files loaded, limited only by available memory. Hopefully, this will make up for the increased program size.

Added ’NM_COMMANDSTRING’ menu shortcuts if running V39+.

Made ’Escape’ quit Textread when the last buffer was closed.

Added command 'Copy settings', which makes all buffers use the current settings.

"+" and "-" now switches to the next/previous buffer.

If a second copy off Textread is started, a new buffer will be opened in the first copy and the second will terminate. This feature can be tured off by using the NEW keyword or tooltype.

Multiply arguments can now be passed from the Shell or Workbench. No wildcards yet, though...
(9.11.94)

Release 1.11

V38.64 V38 of asl.library is not needed anymore, V37 will do, but you loose the screenmode requester.
(30.11.94)

Release 1.13

V38.82 'Edit text' now runs the editor async.

Textread should now be 100% (well, close) compatible will graphics cards, if 'system rendering' is selected.

Added 'Save' and 'Save As...' menu items.

The file requesters now have a title, just like the other ones, when using RTPatch.

Minor improvements to the requesters, removed redundant code.

Improved the ANSI/ISO sequence remover (It did not remove ANSI/ISO sequences, just some common Escape sequences!).

When 'system rendering' is selected, any fixed-width font may be used, as long as 80$\times$20 characters fits on screen. (Expect some slowdown)

When searching, the found word will be hightlighted (or the line the word is on if wildcard is used).

You can now supply your own icon images for the prefs files and the AppIcon, just create the files "ENV:Textread/def_prefs.info" and "ENV:Textread/def_app.info".

(Re)added 'Go to line' (l, #) and 'Go to %' (g, %)

The default preference file is now located in

'ENVARC:Textread/Textread.prefs'.

Added a palette requester. Well, not quite... All it does is running SYS:Tools/Colors (Exists on your WB 2.0 disk. 3.0+ owners may want to try Ian J. Einman's IanUtils (Fred Fish, or AmiNet:dev/asm/IanUtils.lha)

The prefs files have changed format for the last time, the new format is more flexible.

The key binding has changed a bit.

The 'Reveal' list now also shows buffer numbers.

Textread now supports pipes. Read the docs for notes.

Some bugs removed, some small things improved.

Textread never detaches itself from shell anymore. It was inconsequent, and besides, 'run' was made for a reason.

Added Open/Close Workbench menu items.

If a second copy of Textread is started, without the NEW argument, only a small portion of the code will stay resident in memory, waiting for the new buffer in the old Textread to be closed. Ehmm.. let's just say it saves memory.

System rendering and fontfake is allways on (and can't be turned off) when a non-native screenmode is selected. This will avoid potential (chip) memory trashing.

Fixed the "***Corrupt window pointer!" problem WShell reported. Thanks, Ulrich!

Added a requester to prevent accidental saves.

Discovered what caused a frustrating bug that made my custom text routines do nothing (no screen clear, no text printing, no nothing.) It was Magic Layers (1.1) . So if Textread suddenly (or more specific after you have dragged a window on Textread's screen) stops printing, just iconify it and deiconify, and don't drag any more windows on Textreads screen!

Changed the shareware rules a bit.

Removed a bug that caused Textread to crash if the ENVARC: assign did not exist.

```
$VER: Textread guide 38.82 (3.8.95)
```