

TextView

COLLABORATORS

	<i>TITLE :</i> TextView	
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>
WRITTEN BY		April 13, 2022

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	TextView	1
1.1	TextView.guide	1
1.2	introduction	3
1.3	requirements	3
1.4	usage	4
1.5	load	6
1.6	save	7
1.7	crunch	7
1.8	keys	8
1.9	mouse	8
1.10	search	9
1.11	jump	9
1.12	iconify	10
1.13	settings - load & save	10
1.14	settings - gui related	11
1.15	settings - text viewing	14
1.16	settings - info string	16
1.17	Output	18
1.18	html	18
1.19	macros	19
1.20	problems	20
1.21	Reports	20
1.22	changes	20
1.23	distribution	26
1.24	shareware	27
1.25	keyfile	27
1.26	credits	28
1.27	author	29

Chapter 1

TextView

1.1 TextView.guide

=====

January 22, 1999

Torbjörn Andersson
presents

```

      _____
     |         |         |         |         |         |         |         |         |
     | | / _ \ \ / / _ \ \ / / | | / _ \ \ / / | | / _ \ \ / / | | / _ \ \ / /
     | |  _/ > < | | \ v / | |  _/ \ v v /
     | | \ / \ / \ / \ / | | \ / \ / \ / \ /
           Version 1.22

```

=====

1. Contents

Contents	1
Introduction	2
Requirements	3
Usage	4
Load	5
Save	

.....	6
Crunch	7
Keys	8
Mouse	9
Search	10
Jump	11
Iconify	12
Settings - Load & Save	13
Settings - GUI related	14
Settings - Text viewing	15
Settings - Info String	16
Text Output	17
HTML	18
Macros	19
Problems	20
Bug Reports	21
Changes	22
Distribution	23
Shareware	24
Keyfile	25

Credits	26
Author	27

1.2 introduction

2. Introduction

TextView is shareware. Unless you have a keyfile, the About-requester will appear every time you start TextView. Also, two functions will be disabled. Please read more about this under "Shareware" and "Keyfile".

TextView is one of the programs I use most. I made it because I wanted a good and fast textviewer. And TextView is fast. It loads files in a twinkling of an eye, it has custom routines for text output and scrolling, and searching for a string is very fast. I have tried to make it configurable and userfriendly, so I hope it can be used by anybody.

Features:

- Custom routines for text output and scrolling.
- Scrollbar with soft realtime scrolling.
- Can show different fontstyles and colors.
- Proportional fonts are supported.
- Hides AmigaGuide commands.
- Hides HTML tags.
- ~Pipe support.
- Commodity.
- Macro keys.
- Multiple files.
- Files can be shown in hexadecimal form.
- Reads files crunched with PowerPacker or XPK.
- Can crunch and save files.
- Font sensitive GUI.
- Configurable.
- Fast searchroutine.
- Bookmarks.
- ~AppWindow and AppIcon.
- Can be made Resident.
- AmigaOS 1.2/1.3 compatible, AmigaOS 2.0/3.0 aware.

1.3 requirements

3. Requirements

Some functions requires a keyfile. Please read under

"Shareware" for information on how to obtain a keyfile.

A few features require AmigaOS 2.0 or higher, otherwise AmigaOS 1.2 is enough.

Asl.library is required for file, font- and screen mode selecting (part of AmigaOS 2.0+).

Iffparse.library is required if you want to copy blocks to the clipboard (part of AmigaOS 2.0+).

Reqtools.library is required for various functions that require a string or integer input.

Powerpacker.library is required if you want to view a file crunched with PowerPacker, or crunch and save a file.

XPK libraries are required if you want to view a file compressed with XPK.

Arp.library is required on kickstart 1.2/1.3 for command line parsing.

(If you are using kickstart 1.2 or 1.3 you may find my own asl.library useful. It uses reqtools.library and works under KS 1.2/1.3. It is in a package called NewReqLibs, which can be found on Aminet in util/libs.)

1.4 usage

4. Usage

TextView can be started from both CLI and Workbench, and can be made resident.

By typing "TextView ?" you will get this AmigaDOS template:
FILES/M, SETTINGS/K, MACROS/K, PALETTE/K, KEYFILE/K, TABS/K/N,
NORMAL/S, HEX/S, ASCII/S, MOUSEMODE/N, SCROLLBARMODE/N,
DISPLAYBEEP/N, SCREEN/S, WINDOW/S, PUBSCREEN/K, COLORS/N,
FONTNAME/K, FONTSIZE/K/N, CLIPUNIT/K/N, NEW/S, CX_PRIORITY/K/N,
CX_POPKEY/K, CX_POPUP/K:

Most options that you can't change within TextView, will still be saved with the other settings, so you only have to specify it once on the command line.

Explanation of options:

SETTINGS

Specify settings filename with path to use instead of default one.

MACROS

Use this macro file.

PALETTE

Palette file. You can create one with the Workbench 3.0 Palette Preferences editor.

KEYFILE

Keyfile to use.

TABS

Set tabsize.

NORMAL

Force normal view mode. If not NORMAL, HEX or ASCII options are used, hex or ascii mode will automatically be used if the file is executable, an icon, an iff file or has (many) exceptional characters. Which one of hex or ascii that will be used is determined by the mode that was used when the settings file was saved.

HEX

View file in hex mode.

ASCII

View file in ascii mode.

MOUSEMODE

- 0 = Smooth scrolling.
- 1 = Scroll one line.
- 2 = No scrolling.

Add this number for additional option:

- (0 = Continue scrolling until next click.)
- 16 = Scroll only while left mouse button is held down.

SCROLLBARMODE

- 0 = Smooth scrolling.
- 1 = Normal scrolling.
- 2 = Wait until scrollbar is released.

DISPLAYBEEP

- 0 = None of 1 or 2 below.
- 1 = DisplayBeep at end of text.
- 2 = DisplayBeep when searching fails.
- 3 = Both 1 and 2 above.

SCREEN

Use a custom screen.

WINDOW

Open a window on a public screen.

PUBSCREEN (KS 2.0+)

Name of public screen to open window on.

COLORS

Number of colors of screen.

FONTNAME

Name of font for text.

FONTSIZE

Size ditto.

CLIPUNIT

Number of clipboard unit to use.

NEW

Force TextView to open a new window, even if there are other copies of TextView loaded.

CX_PRIORITY (KS 2.0+)

A number from -128 to 127, that sets the priority when TextView registers as a Commodity.

CX_POPKEY (KS 2.0+)

A string that describes a key to use as a hot key for making TextView's window appear.

CX_POPUP (KS 2.0+)

If set to "NO" - start iconified.

1.5 load

5. Load

If you start TextView with no arguments, it will open a file requester. You can specify filenames on the commandline to load directly, or a directory to position the file requester in.

If you want to load a new file, use "Open..." in the Project menu, or "Open New..." if you want to keep the first file.

You can also drag icons over TextView's window, and TextView will load them automatically.

TextView will load the whole file in one turn, which has some disadvantages. It requires that there is a free continuous block of memory as large as the file, and you can't start reading until the whole file is loaded. But it is fast.

If the file was crunched with PowerPacker or XPK, it will automatically be decrunched before it is shown. If it is encrypted you will be asked for a password first.

When the file is loaded, TextView will count the number of lines in it, and at the same time create a small speed up table used when scrolling through the file.

If the file can't be found, '.pp' will be added to the filename, and then TextView will retry to load the file. If it fails, another try will be done with '.xpk'.

If a file is loaded from a pipe device, TextView will first pull the whole file from it in small blocks. Then it will allocate a block of memory as large as the whole file and copy the smaller parts to it. Thus the memory required for this is twice the size of the file, and it will lead to memory fragmentation. But I hope it is better than no pipe support at all.

If there isn't enough free memory to load the whole file, TextView will allow you to read it anyway, by loading parts of it. This doesn't work for crunched files, and not for loading from pipe devices. Also, many features will be disabled.

The loading of a file can be aborted in any way that normally is used to quit TextView.

1.6 save

6. Save

Use "Save As..." in the Project menu to save the current file. TextView will warn you if the file already exists.

"Save All..." will save all loaded files. You will be asked to select a drawer to save all files in. If you select Cancel in the filerequester, you will be asked if you want to overwrite all loaded files instead.

"Save Block As..." saves the lines (if any) that are marked.

1.7 crunch

7. Crunch

Select "Crunch & Save As..." in the Project menu to crunch a loaded file using powerpacker.library and then save it. Select "Crunch & Save All..." to crunch and save all loaded files.

You can't use this function to encrypt or crunch executable files, it only handles "data" files. The efficiency used is always "best". Speedup buffer is automatically adjusted to available memory. One or two asterisks (*) will be shown in the titlebar while crunching if the largest buffer couldn't be allocated, and crunching is then slower. Crunching can be

aborted in any way that normally is used to quit TextView.

1.8 keys

8. Keys

Select "Help" in the Project menu, or simply press the Help-key to see information about the key commands. Press Help again to escape from the Help-file.

By selecting "Information" or pressing Del, you will get information about the files loaded, and some other stuff. An asterisk (*) marks the file that was viewed, you will get back to that one if you press Del again.

The size in bytes of each file is shown in decimal and hexadecimal form, and the address in memory where the file is loaded.

If you press Ctrl-D, TextView will quit with a returncode of 20, which can be used to break scripts.

You can use the arrow keys to view lines that are too long, or use Word Wrap by pressing W. TextView will then format the text by putting line feeds where appropriate. Word Wrap cannot be undone.

Shift-E will start and pass the filename to the editor set in ENV:EDITOR. (Put a line like this in your Startup-Sequence/User-Startup: "Setenv EDITOR C:Ed".)

1.9 mouse

9. Mouse

You can use the mouse to move through the file and marking blocks.

If you hold down the shift key when you press the select button, the line the pointer is over will be marked. Move the pointer to another line, and press select again together with shift. Now a block is marked that can be copied to the clipboard, or printed.

When the select (left) button is pressed, TextView will start to examine the mouse position. When it is moved a little bit away from the middle of the window, the text will start scrolling up or down.

The further the mouse is moved from the middle, the faster the text will scroll. (This can be changed with the

MOUSEMODE option.)

You don't have to keep the button pressed, clicking is enough for TextView to realize that it should start scrolling. Click once again to stop scrolling. (This can be changed with the MOUSEMODE option.)

When the text reaches top or bottom the scrolling will stop, if you move the mouse in the opposite direction it will not scroll back.

1.10 search

10. Search

If you want to search for a string there is a Search menu, but it is easier to just press F or S. Use shift if you want to search for a string case-sensitive. When you have entered the string in the string gadget, the titlebar will read "Searching: <string>...". If the whole file is scanned and the string couldn't be found, the titlebar will read "Not Found". If the string was found, it will be highlighted.

Press N to search for the next occurrence of the string, or P to search for the previous one.

The search function will never move the text to a different horizontal position.

If you search for a string in an AmigaGuide file with "Settings->Rendering->Hide AmigaGuide Commands" set, some occurrences might not be highlighted.

If you search for a string in hex mode, it will be shown at the top of the window, and it will not be highlighted.

1.11 jump

11. Jump

If you press %, G/J or Shift-G/J, TextView will open a string requester for you to enter where to jump to. This will then be shown in the first line of the window.

Jump to "%" means percent of the number of bytes the file contains, not lines. (The percent the titlebar shows is percent of lines, and it counts the bottom line.)

1.12 iconify

12. Iconify

If you select Iconify in the Project menu or start TextView with the argument `CX_POPUP=NO`, and "Settings->Miscellaneous->Workbench Icon" is selected, TextView will place an icon on Workbench.

Double-click on the icon to activate TextView. You can also drop one or more files on the icon, and TextView will load them.

TextView has no built in image for the icon. It searches for "TextView.info" in `ENV:TextView/` and in `PROGDIR:.` If it can't find any icon there, it uses the default icon for tools. `PROGDIR:` is an assign to the directory a program was started from (not available for programs made Resident).

Adding an application icon to Workbench is a new feature of the operating system, you must be running AmigaOS 2.0 or higher for this.

1.13 settings - load & save

13. Settings - Load & Save

The settings menu lets you configure TextView. This section describes choices related to loading and saving.

There are some options in TextView that you only can set from the command line or by using Workbench tooltypes. See "Usage" for an explanation of these.

TextView will search for its configuration file in `ENV:TextView/TextView.prefs`, unless you use the `SETTINGS` argument. When you save the default configuration file, save it in `ENVARC:TextView/TextView.prefs`. From there it will automatically be copied to `ENV:` when you boot.

Note that the configuration file won't be loaded on startup if you don't have a keyfile (see "Keyfile").

Max Buffer Size

Maximum size of buffer in bytes to allocate for file. If the file is larger than this size, only a small buffer will be used to read small parts of the file. If you enter minus one (-1), TextView will always try to allocate a buffer as large as the whole file.

Hold down shift when you select this item, and you will change the size of the small buffer to use for big files instead (default 10,000 bytes).

Loading Size

Bytes to read from file between updates of the progress bar and checking if the user wants to abort the loading. If the file is smaller than this size, no progress bar will be shown.

If you enter zero (0), TextView will adjust the loading size itself for each file. If you enter minus one (-1), TextView will always read the whole file in one turn.

Load Palette

Load a new palette file and change the palette of TextView's screen. You can create a palette file with the Workbench 3.0 Palette Preferences editor.

Load Settings

Load new configuration file for TextView.

Save Settings

Select and save TextView's configuration file.

1.14 settings - gui related

14. Settings - GUI related

This section mostly describes choices related to the GUI.

Miscellaneous->Workbench Icon

Display an icon on the Workbench.

Miscellaneous->Close Gadget

This option was first intended for the screen, because it is not possible to have a closegadget directly on a screen.

If this flag is selected, TextView will open an extra window on the screen with only a system close gadget. The size of the gadget isn't known in advance, so the window will have the size 1x1 when opened. Then TextView finds out the size of the closegadget and resizes the window according to it.

Miscellaneous->Bottom Border

If TextView uses a window on any public screen, the window will have a sizing gadget. This gadget forces a wide border at either the right or the bottom of the window, which is your choice with this option.

Miscellaneous->Make Screen Public

If TextView is using its own custom screen, it will be made public when running under AmigaOS 2.0+. Other applications can then open their windows on TextView's screen in a manner supported by the OS.

Miscellaneous->Screen To Front

Brings the screen to front that TextView opens its window on. It is also pushed back when TextView quits (if it wasn't at front when the window was about to open).

Miscellaneous->Confirm Quit

TextView will open a requester asking if it's ok to quit.

Miscellaneous->Quit=Iconify

When TextView is about to quit, it will instead iconify.

Miscellaneous->Print Convert Tabs

Converts tabs to spaces before text is sent to the printer.

Screen->Use Public Screen

If TextView is running under AmigaOS 1.2/1.3 it will open its window on the Workbench screen. If running under AmigaOS 2.0+ it will first open it on the default public screen, which is often the Workbench screen. If TextView already has its window on any public screen, it will use the next available public screen.

Hold down shift while selecting this one, and TextView will travel through the list of public screens in the opposite direction.

Hold down ctrl to force TextView to open on the default public screen.

Screen->Like Workbench

Use a screen of the same type as Workbench, with the exception of depth and optionally font.

Hold down ctrl when selecting this, and TextView will also use the same depth, font and palette as Workbench.

Screen->Select Type

Lets you select screen type from a requester.

Screen->High Res/High Res Laced

Open screen in hires/hires interlaced mode, with the same size as the Workbench screen.

Hold down shift when you select any of these, and the screen will be in in lores mode instead. Hold down alt, and it will be superhires.

Colors->2/4/8

Specifies depth of screen (1/2/3). Only one color can be used for the text if you select depth 1, but it saves chip memory. If you want a deeper screen, you can select that in the screen mode requester.

Font

The font menu specifies the font that will be used for text output, or the font for the screen titlebar and menus if you hold down the shift key when you pick any of the subitems.

If the selected font is proportional,

GfxBase->DefaultFont (see below) will be used instead when viewing text in hex mode.

Font->Default

When not selected together with any key:

Use GfxBase->DefaultFont for text output (same as the "System Default Text..."-choice in the Font Preferences editor).

Control key:

Use the same font for text output as the one used for the screen titlebar and menus.

Shift:

Use the Workbench screen font ("Select Screen Text..."-choice) for screen titlebar and menus.

Shift+Control:

Use GfxBase->DefaultFont for screen titlebar and menus.

Font->Select

Select font in a requester.

Font->Topaz 8/9

The fonts that live in kickstart.

Scroll Bar->Normal/Border

Use a "normal" scrollbar, or place it in the border of the window. "Border" cannot be selected if TextView is using a custom screen.

The "Border"-choice doesn't care about your left/right/top/bottom selection (rather "Miscellaneous->Bottom Border").

Scroll Bar->Left/Right/Top/Bottom

Have a scrollbar in the preferred location. Only one can be used. Even if it is placed at the top or bottom of the window, it is used for moving the text in the vertical directions only.

Scroll Bar->Width/Height

Change width or height of scrollbar. If you select a value lower than 5, 16 or 10 will be used depending on where the scrollbar is placed (TextView's old default values).

1.15 settings - text viewing

15. Settings - Text viewing

This section mostly describes choices regarding how TextView shows text.

Rendering->Hide/Interpret ANSI Codes

Tells TextView to hide or interpret the codes used for different colors and font styles in some files.

Rendering->Force ANSI

If this flag isn't set, TextView scans the file for ANSI codes, and turns off the previous two options internally if none are found. It searches through 10,000 bytes at most, to be quick even for large files. This flag will force TextView to always Hide/Interpret ANSI codes and skip the checking.

The reason for the check is that TextView has a text output routine that writes characters directly on the screen to be fast, and it is built in a routine that doesn't check for things like ANSI codes. With this option it can be used for files with no ANSI codes, while still files with them are handled automatically.

Rendering->Hide AmigaGuide Commands

TextView will not show commands in AmigaGuide files, which makes them more readable.

Rendering->Remove AmigaGuide Commands

Removes AmigaGuide commands in the current file (not all loaded files) and when new files are loaded.

Rendering->Force AmigaGuide

If this flag isn't set, the previous two options are turned off internally if the file doesn't begin with "@database". Then "Remove AmigaGuide Commands" won't scan the whole file when it is loaded if it isn't an AmigaGuide file. See also "Rendering->Check ANSI".

Rendering->Hide HTML Tags

Hides HTML tags.

Rendering->Remove HTML Tags

Removes HTML tags in the current file (not all loaded files)
and when new files are loaded.

Rendering->Force HTML

If this flag isn't set, the previous two options will be disabled internally if the file hasn't got the string "<!DOCTYPE", "<title>", "<html>" or "<head>" near the beginning.

Rendering->Only ASCII

Non-printable characters will be shown as '.' in hex mode.

Rendering->Use Custom Routines

If this flag is selected, TextView will use its own custom routines for text output and scrolling.

One routine is used for any 8x8 size font, and writes the characters directly on the screen. Two routines use the blitter, one to move text and another to clear areas on the screen.

If this flag isn't selected, there is one routine that might be used anyway. It is used for any 8x8 size font and uses BltTemplate(), like some patches for the Text-function.

This flag cannot be selected if TextView isn't using its own custom screen, or if it's made public.

Rendering->Hide/Replace Characters

TextView can hide characters you want to be hidden, or even replace characters with others.

To hide a character, just write it in the requester or it's ASCII code. To hide all A's, enter "A" or "65". If you want to hide a digit, you must put it in within single quotes. To hide all 0's, enter "'0'" or "48". If you want to hide a range of characters, just enter the first and last characters with a dash (-) between. To hide all characters from A to Z, enter "A-Z" or "65-90".

If you want to replace a character, enter the character you want replaced followed by the one to replace it with, possibly with a comma (,) between. To replace all A's with B's, enter "AB", "A,B" or "65,66". Like when hiding, if you want to replace a digit, you must put it in within single quotes. To replace all 0's with 1's, enter "'0''1'", "'0','1'" or "48,49".

Now for some more useful examples. If you don't want to see non-unprintable characters, enter this in the requester (without the quotes): "0-31 127-159".

If you are reading swedish text in 7-bit ASCII, you can replace characters like }, {, | etc. with a string like this: "@É [Ä \Ö]Å ^Ü `é {ä |ö }å ~ü". You will probably want to use a macro for this.

Text Offsets->Left/Right/Top/Bottom

Set minimum offset for text from screen edges or window borders.

Tab Size

Opens a requester and lets you set the tab size.

Page Overlap

Number of lines that will be shown from one page when going to the next.

Clipboard Unit

Select clipboard unit for copying text to.

Highlight Line

Show the search string at this line from the top of the display. If you enter zero, occurrences will be shown in the middle.

1.16 settings - info string

16. Settings - Info String

This section describes the configuration of TextView's status line in the titlebar, which I call "Info String".

Note that by using this function it is easy to make TextView cause Enforcer hits, and maybe crash the machine.

The "Info String" specifies what will be shown in the titlebar to the right. You enter a quotation mark ("), then a string (called "FormatString") that will be passed directly to the OS-function `exec.library/RawDoFmt()`, another quotation mark, and finally one or more commas (,) with a following special character ("DataStream"). Some examples:

Info String	Possible result
"line %lU col %lU",y,x	line 1 col 1
"line %lU col %lU",y,x	line 18,000 col 1
"line %6lU col %4lU",y,x	line 1 col 1
"line %6lU col %4lU",y,x	line 18,000 col 1
"line %-6lU col %-4lU",y,x	line 1 col 1
"line %-6lU col %-4lU",y,x	line 18,000 col 1
"%s %s %s",w,d,t	Sunday 26-Oct-97 14:55:14
"%.3s %.6s %.5s",w,d,t	Sun 26-Oct 14:55

```

"C:%lu F:%lu",c,f          C:1864504 F:4302392
"T:%lU",ak                T:6,022
"%lU:%lU/%lU %lU%",x,y,l,p  1:31/59 100%
<-----> <----->
FormatString      DataStream

```

The special characters represent information that will be formatted according to the FormatString. Here's a list of characters you can use and what they represent:

```

x - column
y - line
l - number of lines in file
s - size of file
p - percentage shown
v - file currently viewed
u - number of files loaded
c - available chip memory
f - available fast memory
a - total amount of memory available
d - date (%s)
w - week day (%s)
t - time (%s)

```

%s means this character should be used with the % option "s" in the FormatString. Otherwise use d, u or x (upper or lower case). If you enter "k" right after a character, the information it represents will be divided by 1024.

To explain the function RawDoFmt() and the FormatString I'll include the parts where they are explained in the autdocs (with some changes).

RawDoFmt() - FUNCTION:

```

>perform "C"-language-like formatting of a data stream,
>outputting the result a character at a time. Where %
>formatting commands are found in the FormatString, they will
>be replaced with the corresponding element in the
>DataStream. %% must be used in the string if a % is desired
>in the output.

```

RawDoFmt() - INPUTS:

```

>FormatString - a "C"-language-like NULL terminated format
>string, with the following supported % options:
>
> %[flags][width.limit][length]type
>
>flags - only one allowed. '-' specifies left justification.
>width - field width. If the first character is a '0', the
>        field will be padded with leading 0s.
> . - must follow the field width, if specified.
>limit - maximum number of characters to output from a
>        string (only valid for %s).
>length - size of input data defaults to WORD for types d, u,
>         and x, 'l' changes this to long (32-bit).
>type - supported types are:
>        d - signed decimal
>        D - signed decimal using the locale's
>           formatting conventions
>        u - unsigned decimal

```

```

>          U - unsigned decimal using the locale's
>             formatting conventions
>          x - hexadecimal with hex digits in uppercase
>          X - hexadecimal with hex digits in lowercase
>          s - string, a 32-bit pointer to a NULL
>             terminated byte string

```

It's possible to specify the rightmost position of the "Info String", like this:

```
x=500,"%1U:%1U/%1U %1U%%",x,y,l,p
```

By default it's shown as far to the right as possible.

1.17 Output

17. Text Output

Here is how TextView treats characters for text output in files:

```

0x09 (HORIZONTAL TAB) - replaced with spaces.
0x0A (LINE FEED)      - new line.
0x0D (CARRIAGE RETURN) - not shown.

```

If in ANSI mode also:

```

0x08 (BACKSPACE)      - this and previous character is not
                        shown, as long as the text is not
                        moved to the left.
0x1B (ESCAPE)         - triggers ANSI handling.
0x9B (CONTROL SEQUENCE INTRODUCER) - like 0x1B.

```

If AmigaGuide:

```
0x40 ("@") - checks AmigaGuide command.
```

HTML:

```

0x26 ("&") - tries to replace from this one
            until ";" with some character.
0x60 ("<") - often removes everything from this
            until ">".

```

1.18 html

18. HTML

If a HTML tag moves over more than one line like this:

```
<blah ....
... blah>
```

it will be shown even if you select "Hide HTML Tags". If you select "Remove HTML Tags" it will be removed/replaced.

Here is a list of tags that will be replaced with a character, and what character.

Tag	Char
	·

"Remove HTML Tags" will replace these tags with LF:s (0x0A):

<p>

A lot of entities are replaced with single characters, e.g.
& is replaced with &.

1.19 macros

19. Macros

Note that you need the keyfile to execute a macro (see "Keyfile").

TextView will search for a macro file in ENV:TextView/TextView.macros, unless you use the MACROS argument. When you save a macro file, save it in ENVARC:TextView/TextView.macros. From there it will be automatically copied to ENV: when you boot.

Any number of keystrokes or menu selections can be saved in a macro definition and assigned to a key. When you later use that key, they will be played back. You can for example emulate the keys of another textviewer, or make keyboard shortcuts for menu items that don't have any.

"Define Macro" in the Commands menu will start defining a macro. First TextView waits for you to press the key you want to assign the macro to (the titlebar says "Press macro key."). The key(s) you press after that, or menu selection(s) you make will be saved in the macro. Any click on the close gadget will be saved too. When you are finished with a macro definition, select "Define Macro" again.

"Clear Macros" will discard all defined macros.

"Load Macros..." will load a new macro file. If any old macro file is in use, it will be discarded.

"Save Macros..." will save the macro file currently in use.

An old macro definition will be cleared as soon as you press the key it is assigned to for defining a new macro.

1.20 problems

20. Problems

If you bring up the menus while TextView is scrolling text using its custom routines, the menus and some text can get garbled.

All ANSI codes are not interpreted correctly. Colors and styles are reset at the beginning of each line. AmigaGuide and HTML handling is not perfect either.

Blocks marked in hex mode will always be in normal mode if they are saved as a file or copied to the clipboard.

Some menu selections in the macro file will get wrong when the menus change between different versions of TextView.

1.21 Reports

21. Bug Reports

If you find bugs in TextView, I want to know about them. If you report a bug, it is good if you can provide many details. Tell me exactly what you do and what happens.

It can help if I know what system you run, tell me about your:

- ~Hardware (Amiga, CPU, memory, chipset/graphics card)
- Version of:
 - TextView
 - AmigaOS (Kickstart and Workbench)
 - Libraries that can have something do to with the bug(s) (asl.library, reqtools.library, etc.).

Try TextView with as few other things running as possible. A problem with one program may very well be caused by other software.

1.22 changes

22. Changes

1.22 (22.1.99)

- One serious bug fixed, which sometimes would clear four bytes at some location in memory. Even if you haven't experienced any strange problems with TextView, please replace old versions with this one.

1.21 (1.10.98)

- Added "Load Palette..." and removed old palette options.
- Removed all code that used `exec.library/SetTaskPri()`, you can't set TextView's priority anymore (within TextView).
- Will only look for prefs-files etc. in ENV:, unless you tell where they are on the command line.
- If you have a file loaded and are looking at an empty file, TextView will go back to the other file instead of quitting if you press Esc.
- ~Added character u in "Info String", for showing number of files loaded.
- You can both set maximum size of buffer to use for loading whole files, and the size of the buffer used when a file is larger than this size.
- "Max Buffer Size" is ignored when loading a crunched file.
- ~Added DISPLAYBEEP tooltypes/commandline option.
- ~Added "Settings->Rendering->Hide/Replace Characters".
- Miscellaneous bugs fixed.

1.20 (28.2.98)

- Added uudecode function.
- "Info String" can be moved to the left in the titlebar.

1.19 (1.12.97)

- Miscellaneous bugs fixed.

1.18 (25.11.97)

- Name of icon will always be "TextView" instead of the name of the file that was viewed when the icon was added.
- Default font for screen titlebar and menus are now the same as the Workbench screen font.
- Added command "Ascii Dump".
- Renamed command line option ASCII to NORMAL. ASCII now means "Ascii Dump" will be used.
- When TextView detects a binary file, it will be shown in hex or ascii mode, depending on what was used when the settings file was saved.
- Added "Settings->Info String".
- ~Added MOUSEMODE and SCROLLBARMODE options.
- ~"Information" only shows information about loaded files, and nothing about the system anymore.
- Removed ability to save settings to default settings file without using `filerequester`.
- Miscellaneous bugs fixed.

1.17 (25.10.97)

- Maximum size of buffer to allocate for a file can be set.
- Uses different `filerequesters` for settings/macros.
- Added reload function.
- Bytes to read from file between updates of the progress bar can be set.
- Added two options for font usage, used when selecting "Settings->Font->Default" and holding down the ctrl key.
- Miscellaneous bugs fixed.

1.16 (8.9.97)

- Added COLORS command-line option.
- "Remove HTML Tags" handles `<pre>/</pre>`. Made some other

- small changes. Some documents may look worse than earlier.
- Miscellaneous bugs fixed.

1.15 (15.8.97)

- Loading routine rewritten.
Removed internal ppDecrunchBuffer(), powerpacker.library is always required for decrunching.
Uses the progress bar when loading normal files too, not just xpk-files.
Titlebar is set to 'Loading...' only when the file is actually being read.
Checks for user break at more places.
- The progress bar is used when crunching a file.
- Added Refresh command.
- Added "Close All", "Save All" and "Crunch & Save All".
- Searching works even when whole file isn't loaded (forwards only).
- Macro function rewritten.
You can put in as many commands as there is available memory for. Menus can be used in macro.
The contents of the File-, Font-, Screen Mode-, String-, Integer- and Palette requesters can be saved in the macro.
- If you are out of memory, it's likely there isn't enough memory to open a requester to inform about this. So now TextView gives that information in the titlebar also.
- Titlebar shows name of file when loading/(de)crunching.
- Titlebar shows string when searching.
- Esc key terminates TextView if only one file is open.
- ~Removed random palette.
- If a drawer icon is dragged over TextView's window, the filerequester will open and be positioned in that drawer.
- Initial filerequester opens on default public screen.
- Proportional fonts are now supported (except in hex mode).
- Added "Settings->Miscellaneous->AppIcon", so the icon can always be on Workbench, or never (requires that CX_POPKEY have been initialized) or like before only when TextView is iconified (by using the macro facility).
- Added keyfile requirement for some functions.
- Miscellaneous bugs fixed.

1.14 (14.6.97)

- Removed scrolling of pixel for pixel with topaz 8 font.
- If an icon starts TextView as Default Tool, its Tool Types will be read and used by TextView.
- GetFont updated and released separately (Aminet:dev/misc/Font2src).
- Miscellaneous bugs fixed.

1.13 (28.5.97)

- Miscellaneous bugs fixed.

1.12 (17.5.97)

- Miscellaneous bugs fixed.

1.11 (15.5.97)

- File requester has pattern gadget.
 - Reads file like More when there isn't enough memory to
-

load the whole file (searching and other functions are disabled).

- Miscellaneous bugs fixed.

1.10 (23.3.97)

- Can be controlled from Commodities Exchange.
- ~"ICON" argument changed to "CX_POPUP", i.e. "CX_POPUP=NO".
- "Case Sensitive" and "Through All Buffers" options will now be saved together with the other settings.
- ~Searching through the rest of the buffers will start from the beginning in each of them, and not from the current position they have.
- ~Ctrl-T will go to the top of the first file.
- If a file is loaded from a pipe device the menu will show the filename "PIPE:".
- Can pass files to the first started copy of TextView.
- Added ability to define single key macros.
- Pattern matches filenames specified on the commandline.
- Miscellaneous bugs fixed.

1.9 (26.1.97)

- Miscellaneous bugs fixed.

1.8 (18.1.97)

- Removed option to use reqtools.library for file-, font- and screen mode requesters.
- ~Added option for where to show search string.
- ~Doesn't update whole display when you start to scroll after searching.
- ~Some characters that were hidden in ANSI mode (0x00-0x1F and 0x80-0x9F) is now shown.
- ~Removed calls to DisplayBeep() when text won't scroll further.
- Added Hide/Remove HTML Tags.
- Arp.library is now used for commandline parsing on KS 1.3.
- Removed Ctrl Characters option and created the program GetFont instead.
- Reads Workbench tooltypes.
- Added the following tooltypes and commandline options: SCREEN, WINDOW, PUBSCREEN, FONTNAME, FONTSIZE, CLIPUNIT.
- Removed "Blank Mouse" option.
- Added "Through All Buffers" search option.

1.7 (6.9.96)

- Uses powerpacker.library/ppWriteDataHeader() for writing the data header when saving crunched file.
 - Error on loading/saving checks DOS error code, so the requesters give more information.
 - Adds '.xpk' to filename and retries if file is not found.
 - ~Better highlightning of search-string.
 - Informs if asl requester can't open (ASL V38+).
 - Loading/unpacking XPK file can be aborted.
 - Multiple files can be loaded directly by passing multiple arguments from Shell or Workbench, or selecting multiple files in the file requester.
 - The AppIcon will find a new buffer instead of loading over an old file.
-

- Changes file buffer with +/-.
- Limit of 10 files that can be loaded is gone.
- All calls to DisplayAlert() and AutoRequest() replaced with EasyRequest() (KS 2.0+).
- ~Screen mode requester can be used to set any depth.
- Option to use close gadget applies window on public screen too.
- Added "Page Overlap..." in Settings menu.
- ~Loaded files are shown in the menu.
- ~A requester tells what name the screen has when you select Make Screen Public.
- Screen AutoScroll can now be changed in the screen mode requesters.
- ~Removed "Reset To Defaults" choice in the Settings menu.
- ~Settings can now be loaded/saved by using the file requesters.
- Added SETTINGS commandline option.
- Blocks can be marked and copied to the clipboard.
- Added "Hide ANSI Codes".
- Added checking for ANSI codes in file.
- No quick choices for Tab Size in Settings menu.
- ~Added option to set Text Offsets from the window edges.
- ~Project->About... uses a requester, and will not show the command keys.
- Project->Information won't show commands, added Help item for that. Both creates an internal file used almost like any other.
- ~Added "Remove AmigaGuide Commands".
- ~Can print file/block.
- ~"Settings->Miscellaneous" got a few new choices.
- Makes the window a Workbench AppWindow (KS 2.0+).
- ~ASL file requester won't show .info files.
- ~Removed most ReadArgs() emulation for KS 1.2/1.3 to save space, it only takes one argument; [file|dir].
- ~Doesn't update the scrollbar when scrolling with the mouse until scrolling stops, to make the scrolling more softly.
- ~Removed binary view mode.
- ~Scrollbar can be placed in the border.
- ~Font can be selected for screen too.
- ~Saves position and width/height of zoomed window and all asl-requesters in settings-file.
- Miscellaneous bugs fixed.

1.6 (31.5.96)

- Added checking for and loading of files compressed with XPK.
- Sets priority to -1 when decrunching.
- Fixed menus for close gadget window.
- Shows co-processor (68881/68882) if present after main processor when help key is used.
- Miscellaneous bugs fixed.

1.5 (29.4.96)

- Can load files from pipe devices.
- Added ability to have multiple files loaded.
- ~Only one menuitem to crunch and save file.
- Information/Help shows information about the files,

available memory, chipset/CPU/OS info and key commands if they fit. Decimal numbers has commas to be easier to read.

- About/? now only shows copyright and key commands.
- Starts editor set in ENV:EDITOR with Shift-E.
- Added Goto %/Line/Byte.
- Added Forward and Case Sensitive options in Search menu.
- Tab Size item in Settings menu in line with the others.
- Scrolling with the mouse stops when reaching top/bottom, i.e. it won't scroll back when you drag the mouse in the opposite direction.
- Decides about hex/normal view mode by itself, unless BIN/HEX/ASCII argument is given.
- Ability to Iconify added by using an icon on workbench. Can start iconified with the ICON parameter. One file at a time can be opened and read by dragging it over the icon.
- Added ability to choose scrollbar width.
- Characters shown as '.' in bin/hex mode was 0-31, it is now 0-31 and 128-159, and made optional.
- With any 8x8 size font, characters 0-31 and 128-159 can be shown as the inverse of the characters 64-95 and 192-223.
- Scrolling with the mouse scrolls pixel for pixel with topaz 8.
- Switches ANSI on/off with A.
- Miscellaneous bugs fixed.

1.4 (8.2.96)

- Added bin/hex commandline options.
- Miscellaneous bugs fixed.

1.3 (3.2.96)

- Opening window on public screen brings screen to front and then opens window, not the other way around.
 - Opening window on public screen will bring back eventual screen that was in front before.
 - ~Saving settings saves name of public screen, so TextView won't always necessarily open on default public screen.
 - ~Remembers window position and size better.
 - Menu items that has no effect at the moment are ghosted.
 - Made text output faster when using any 8x8 size font.
 - ~Implemented smooth, pixel resolution, realtime scrolling using the scrollbar.
 - ~Scrollbar can be placed at the top or the bottom of the window.
 - Never shows carriage return, so you don't have to select "Settings->ANSI" if you only want them out of the way.
 - Right mouse button is no longer used for scrolling, only the left one.
 - Added ability to directly choose hires and hires-interlace screenmodes.
 - Added menu item "Settings->Colors->Random".
 - Sets priority to -1 when crunching.
 - ~Put back option to use asl.library for file-, font- and screen mode requesters.
 - Non-printable characters are shown as '.' in bin/hex mode.
 - Added close gadget on screen.
 - ANSI flag handles backspace, except when the beginning of
-

the lines aren't shown.

- New menu: Search.
- Can decrunch non-encrypted files without powerpacker.lib.
- Added option "Use Custom Routines". Before they were used when "Make Screen Public" was deselected.
- Word Wrap implemented.
- Miscellaneous bugs fixed.

1.2 (29.7.95)

- ~Removed option to use asl.library for file-, font- and screen mode requesters.
- Removed option to print file.
- ~Menus under KS 1.3 look exactly the same as under 2.0.
- Integrated menu Project->About and Help key functions. This function now also shows file location in memory, available memory and some other info.
- Quits with a return code of 20 (RETURN_FAIL) in case of an error, or if Ctrl-D is received.
- Listens to Ctrl-C, Ctrl-D and Ctrl-F. When Ctrl-C is received TextView will quit. When Ctrl-D is received TextView will quit with a return code of 20 (RETURN_FAIL). When Ctrl-F is received TextView will bring its window to the front and activate it.
- Line length limit extended from 1024 to 65535 characters.
- Can crunch (but not encrypt) file and save it crunched.
- Alt and arrow key moves 1/5th of page.
- Save As and Save PP use separate requester buffer.
- Successful saving adds file in requester buffer.
- ~Highlights string searched for with pen color 3 if second bitplane already is in use.
- ~Added Guide flag in Settings menu.
- Miscellaneous bugs fixed.
- TextView is now shareware.

1.1 (8.11.94)

- Can load encrypted files crunched with PowerPacker.
- Uses ENV:/ENVARC: rather than S: for configuration file.
- Scroll page one char left or right is faster.
- Miscellaneous bugs fixed.

1.0 (31.10.94)

- First release.

1.23 distribution

23. Distribution

No commercial usage of TextView is allowed without written permission from the author. Other distribution is allowed if no files in the original distribution are removed or modified. The special keyfile available from the author is not freely distributable.

TextView may be included on coverdisks for magazines, but

only if a copy of the magazine is sent to me.

Like I have been doing, I intend to upload new versions of TextView to Aminet, so I hope you always can find the latest release on any site (in text/show).

1.24 shareware

24. Shareware

TextView is shareware, which means that if you use it for more than 30 days, you are requested to pay 40 SEK, 5 USD or 10 DM for it. No other currencies will be accepted.

The only way to pay the fee is to send cash to the right address (see "Author"). Postal or international money order will not be accepted. When I receive the money, you become a registered user. In return I will send a keyfile to your email address (see "Keyfile"). If you don't have an email address, you have to give me a stamped envelope and a disk in order to get the keyfile.

Please write your full name and email address clearly. If you want to be listed in the "Credits" section (with or without email address/homepage location), just let me know (write it in the letter, if you tell me by email I'll probably forget it...). If you don't mention anything about this, you will of course not be listed.

1.25 keyfile

25. Keyfile

Some functions in TextView requires a keyfile to work. Please read under "Shareware" for informatin on how to obtain a keyfile.

TextView will look for the keyfile in ENV:TextView/TextView.key, unless you use the KEYFILE argument.

These functions will be unlocked by the keyfile:

- Loading of settings file on startup.
- Ability to execute macros.

Also, the About-requester won't come up automatically when TextView is started.

The name and registration number of the owner to the keyfile is encoded in the keyfile. This information will be shown in the About-requester.

Note that the keyfile may only be copied for personal use.
It is not freely distributable.

1.26 credits

26. Credits

Thanks to:

- Nico François for reqtools.library and powerpacker.library.
- ~The authors of the XPK system.
- Everyone who have helped me in any way with this program (bugreports/betatesting/suggestions etc.).

A very special thank you to all who have payed the shareware fee (75 at the moment). Naturally all of you don't want to be listed here, but I may also have missed some of you. If you want a change in this list of any kind, please tell me and I'll update it. Thanks again:

- Michael Schubert (D.R.M.@gmx.de)
<http://rummelplatz.uni-mannheim.de/~mics>
 - Donald Feldbruegge (dhfeldbr@facstaff.wisc.edu)
 - Peter Suggitt (peter@apsayton.demon.co.uk)
<http://www.apsayton.demon.co.uk/>
 - Torsten Jung (t.jung@bbrandes.in-brb.de)
 - Rickard Bergström (rdb@swipnet.se)
<http://home2.swipnet.se/~w-25592>
 - Michael Link
 - ~Ralph Debusmann (rade@coli.uni-sb.de)
 - ~Kary Lovette (ad959@detroit.freenet.org)
 - Michael Schulz (orbital@t-online.de)
<http://home.pages.de/~amigaclub/>
 - Torben Danø (nebrot@post8.tele.dk)
 - Frank Bunton (bunton@acay.com.au)
<http://www.acay.com.au/~bunton>
 - Gerd Jaunich (icondata@supernet.ab.ca)
 - Pascal Marcelin (pmarce@faxnet.surle.net)
 - Kenneth Jonsson
 - Jon B. Peterson (jonbp@webstar.net)
(alden1@airmail.net)
(alden@null.net)
 - Kjell Asker (kjella@alfaskop.net)
 - Donald Dalley (ab732@torfree.net)
 - Lee McCutcheon (clutch@istar.ca)
 - Sanford Hersh
 - Fred Cookman (fcookman@europa.com)
 - Geoff E Tegerdine (gtegerdine@cix.compulink.co.uk)
 - Peter Mattsson (pmats@algonet.se)
<http://move.to/nature/>
 - S.H.Bouwhuis
 - Jürgen Urbanek
-

- Niels de Koning (nko@cypres.nl)
- Torsten Pieper (torsten.pieper@hamburg.netsurf.de)
- Alex Kunz (neurodancer@another.gun.de)
- Haidinger Walter (walthaid@unix.ict.tuwien.ac.at)
- Jörgen Sjöbom (josjo@swipnet.se)
- ~Andreas Kramer
- ~Blake Scruggs
- ~Thomas Schmidt (Moses@PSP.net)
- ~Graham J. Chilman
- Hans Josefsson (hjos@algonet.se)
- Brian Sharman
- F. Hille
- Jari Karjalainen (jaka@sci.fi)
- ~Timo Kaikumaa (timok@cs.tut.fi)
- Will Stanton
- Francesco Agnoli
- Bryan Westbrook
- Charlene McNulty
- Robert Bromley

1.27 author

27. Author

TextView executable and documentation are made by and
copyright © 1994-1999 Torbjörn A. Andersson.
All Rights Reserved.

Please write to me if you can use email and have any
comments or questions. This is my email address:

d95ta@efd.lth.se

Here is my home page location:

<http://www.efd.lth.se/~d95ta>

This is my snail mail address to my parents home, it has
been and will probably be valid for years:

Torbjörn Andersson
Knöppletorp 4379
S-380 31 LÄCKEBY
SWEDEN

This is my student address. The best would be if you ask me
via email which address I'm at currently. Otherwise use the
one above, as no one here will forward my mail when I'm
with my parents:

Torbjörn Andersson
Magistratsvägen 55, K415
S-226 44 LUND
SWEDEN

P.S.

If you don't know how to get letters such as 'ö', try this:
Press and release Alt-K, then press 'o'.
