

TEXt Plus Professional

A TEX-Frontend-Word-Processor
Version 5.00

©1988–1994 Martin Steppler

Copyright

\TeX t Plus Professional software and documentation are copyright © 1988-1994 by Martin Steppler. No part of the software or documentation may be reproduced, transmitted into other languages, posted to a network, or distributed in any way without the express written permission of the author.

This product comes along 'as is' with no representation of fitness for any particular purpose. The user assumes all risks and responsibilities related to its use. The material within is believed to be accurate, but the author reserves the right to make changes to the software or documentation without notice.

\TeX t Plus Professional V5.00 is **shareware**, i. e. if you use TPP regularly you should send me a donation (DM 50/\$ 30/£ 20). Updates from V2.xx to V5.00 cost DM 30/\$20/£ 12 and from V3.xx to V5.00 DM 10/\$ 10/£ 5. Updates from V4.xx to V5.00 are free. Registered users get the latest version of TPP and a keyfile in order to prevent the nerve-racking requester from popping up. If you have access to e-mail, you can send me your PGP-key, which I will use to return an encrypted keyfile to you. PGP can be obtained from aminet. You can also find the latest version of TPP on aminet.

To all pd-distributors:

TPP with the nerve-racking requester may be distributed by everybody, without by none. The price should not exceed a reasonable fee.

The nerve-racking requester has been implemented in order to make it easier for you to decide whether to send the share or not.

Send your share, bug-reports, etc. to:

Address Martin Steppler
 Roermonder Str. 112a/69
 D-52072 Aachen
 Fed. Rep. of Germany

Net addresses InterNet: steppler@pool.informatik.rwth-aachen.de
 FidoNet: Martin Steppler @ 2:242/7.12

\TeX t Plus Professional was developed on an Amiga 3000 with 10 MB RAM, a 240 MB harddisk, and a HP LaserJet 4L using the Matt Dillon's DICE C compiler.

Acknowledgements

Besides Donald Knuth and Leslie Lamport I would like to thank in particular Georg Hessmann (Pas \TeX), Nico François (reqtools.library, powerpacker.library), Christian Rank (NICET \TeX), Rainer Sengerling (dinbrief.sty) and my beta-testers Jens Aperdannier, Markus Baumeister, Stephan Kohler, Dirk Ludwig, Gregor Mauer, Johannes Maybaum, Karl Skibinski, and Martin Ziemer. Finally, I want to thank Konrad Neuwirth, the distributor of Amiga \TeX for Germany and Austria, for sending me a copy of Amiga \TeX and Markus Erlmeier, the Amiga Site Coordinator of DANTE, for getting me in contact with Mr. Neuwirth.

Preface

TPP is a \TeX frontend word processor, i. e. with the help of TPP you can create \TeX -documents without having to know anything about \TeX . The professional typesetting program \TeX is the product of more than ten years of work of Donald E. Knuth , professor at Stanford University, California.

What are the pros and cons of using \TeX ?

If you want to create documents that distinguish themselves by a flippy design including flashy graphics, you are better advised to look out for a desktop publishing program. But if you intend to create well structured texts, beautiful letters, or scientific publications, \TeX is the state of the art.

A great disadvantage has prevented so far the spreading of \TeX : It is very complicated. You rather have to program a text than to write it.

This is exactly where TPP starts to work. Now a document can be created as follows:

- First, you load a form sheet (e. g. a letter form sheet). Within this form sheet several keywords are located, which dertermine the document's structure. WYSIWYG¹ is almost totally renounced, due to the fact that you only have to care for the proper content and not for the automatically generated design of your document.
- After having filled out the form sheet, the text is parsed and compiled. First, TPP creates a \TeX -file and after that \TeX compiles this file.
- The final product of these two compilations is a DVI²-file, which you are then able to preview or print.

So the great disadvantage of \TeX has hopefully been reduced to a minimum.

Aachen, January 1994

Martin Steppler

¹What You See Is What You Get

²DeViceIndependent

Contents

1	Installing And Starting TPP	1
1.1	Hard- and Software Requirements	1
1.2	Installation	1
1.3	Start	1
2	The Menus	3
2.1	The Structure Of This Manual	3
2.2	Project	3
2.2.1	Load	3
2.2.2	Append file	3
2.2.3	New Window	3
2.2.4	Load & New Window	3
2.2.5	Load Configuration	4
2.2.6	Delete File	4
2.2.7	Save	4
2.2.8	Save as	4
2.2.9	Save & Quit	4
2.2.10	Save Configuration	4
2.2.11	Compile	4
2.2.12	Preview	4
2.2.13	Print	5
2.2.14	Iconify	7
2.2.15	Help	8
2.2.16	About	8
2.2.17	Quit	8
2.3	Block	8
2.3.1	Mark	8
2.3.2	Copy	8
2.3.3	Move	8
2.3.4	Paste	8
2.3.5	Delete block	8
2.3.6	Load	8
2.3.7	Save	8
2.3.8	Delete marks	9
2.3.9	Copy to Tmpfile	9
2.3.10	Clipboard	9
	2.3.10.1 Copy	9
	2.3.10.2 Cut	9
	2.3.10.3 Paste	9
2.4	Prefs	9
2.4.1	Prefs	9
2.4.2	Screen	11
2.4.3	Colors	11

2.4.4	Font	11
2.4.5	Keymap	11
2.4.6	Menus	11
2.4.7	Macros	11
2.4.8	T _E X-Paths	11
2.4.9	Right Edge	12
2.4.10	Left Edge	12
2.4.11	Indent Block	12
2.4.12	Page Length	12
2.4.13	Auto-Save	12
2.5	Style	12
2.5.1	Style	12
2.5.2	Justify	12
2.5.3	Color	12
2.6	Find	13
2.6.1	Find	13
2.6.2	Replace	13
2.7	T _E X	13
2.7.1	Letter	13
2.7.2	Book	14
2.7.3	Global	15
2.7.4	L ^A T _E X	16
2.7.5	User	17
2.7.6	Mailmerge	18
2.7.7	MM-Mark	19
2.7.8	Table	19
2.7.9	Graphics	20
2.8	Extras	20
2.8.1	Files	20
2.8.2	Undo	21
2.8.3	Date	21
2.8.4	Set Mark	21
2.8.5	Go to Mark	21
2.8.6	Go to	21
2.8.7	Go to line x.	21
2.8.8	Extern	21
2.9	Creating Documents	21
3	The ARexx-Port Of TPP	23
3.1	ARexx-Returncodes	23
3.2	ARexx-Commands	24
3.2.1	Activate	24
3.2.2	Append	24
3.2.3	AutoDivMode	24
3.2.4	AutoSave	24
3.2.5	BackTab	25
3.2.6	BackUpMode	25
3.2.7	BBottom	25
3.2.8	BCopy	25
3.2.9	BCopyTmp	25
3.2.10	BDelBackTab	26
3.2.11	BDelete	26
3.2.12	BIndent	26
3.2.13	BInsTab	26

3.2.14	BLeft	26
3.2.15	BLoad	27
3.2.16	Block	27
3.2.17	BMode	27
3.2.18	BMove	27
3.2.19	Bottom	28
3.2.20	BPaste	28
3.2.21	BRight	28
3.2.22	BS	28
3.2.23	BSave	28
3.2.24	BTop	29
3.2.25	CallMacro	29
3.2.26	Casesensitive	29
3.2.27	ChFilename	29
3.2.28	Clear	29
3.2.29	ClearTab	30
3.2.30	ClearTabs	30
3.2.31	ClipCopy	30
3.2.32	ClipCut	30
3.2.33	ClipPaste	30
3.2.34	Clock	31
3.2.35	Color	31
3.2.36	Continuous	31
3.2.37	CR	31
3.2.38	DelBackTab	32
3.2.39	Del	32
3.2.40	DelLeft	32
3.2.41	DelLine	32
3.2.42	DelRight	32
3.2.43	DelSpace	33
3.2.44	Display	33
3.2.45	Down	33
3.2.46	Execute	33
3.2.47	Find	33
3.2.48	First	34
3.2.49	GetChar	34
3.2.50	GetColumn	34
3.2.51	GetFontHeight	34
3.2.52	GetLine	35
3.2.53	GetName	35
3.2.54	GetScreenLeftEdge	35
3.2.55	GetScreenTopEdge	35
3.2.56	GetScreenHeight	36
3.2.57	GetScreenWidth	36
3.2.58	GetScreenName	36
3.2.59	GetWindowLeftEdge	36
3.2.60	GetWindowTopEdge	37
3.2.61	GetWindowWidth	37
3.2.62	GetWindowHeight	37
3.2.63	GetString	37
3.2.64	GetWord	38
3.2.65	GoTo	38
3.2.66	GoToLine	38
3.2.67	GoToMark	38

3.2.68	Iconify	38
3.2.69	InsertDate	39
3.2.70	InsertMode	39
3.2.71	InsTab	39
3.2.72	IsWord	39
3.2.73	Justify	40
3.2.74	Last	40
3.2.75	Left	40
3.2.76	LeftEdge	40
3.2.77	LetterMode	41
3.2.78	Load	41
3.2.79	MoveWindow	41
3.2.80	NewWindow	41
3.2.81	Next	42
3.2.82	NextR	42
3.2.83	PageDown	42
3.2.84	PageLength	42
3.2.85	PageUp	43
3.2.86	Prev	43
3.2.87	PrevR	43
3.2.88	Replace	43
3.2.89	Quit	44
3.2.90	RequestInt	44
3.2.91	RequestString	44
3.2.92	Right	45
3.2.93	RightEdge	45
3.2.94	RX	45
3.2.95	Save	45
3.2.96	SaveAndQuit	46
3.2.97	ScreenBottom	46
3.2.98	ScreenDown	46
3.2.99	ScreenToBack	46
3.2.100	ScreenToFront	46
3.2.101	ScreenTop	47
3.2.102	ScreenUp	47
3.2.103	SetColors	47
3.2.104	SetFont	47
3.2.105	SetMark	47
3.2.106	SetTab	48
3.2.107	SizeWindow	48
3.2.108	Style	48
3.2.109	Tab	48
3.2.110	Top	49
3.2.111	ToMouse	49
3.2.112	Type	49
3.2.113	UnBlock	49
3.2.114	Undo	49
3.2.115	UndoDelLine	50
3.2.116	Up	50
3.2.117	WDelLeft	50
3.2.118	WDelRight	50
3.2.119	WindowToBack	50
3.2.120	WindowToFront	51
3.2.121	WLeft	51

3.2.122WRight	51
A The Keymap Of TPP	53
B The printer.device Escape-Sequences	55
Bibliography	57
Index	58

Chapter 1

Installing And Starting TPP

1.1 Hard- and Software Requirements

In order to run TPP correctly in cooperation with PasTeX or AmigaTeX, you need:

- AmigaOS 2.04 (V37 or higher)
- at least 2 MB RAM
- at least 10 MB harddisk space
- PasTeX V1.3 or higher incl. NICETEX. The distribution of PasTeX contains the macro package NICETEX. Or AmigaTeX of Radical Eye Software.
- L^ATeX (25th March 1992). The files needed by TPP are part of the distribution.

1.2 Installation

Double clicking the icon of the installation script is all you have to do to install TPP. PasTeX or AmigaTeX must have been installed before the installation of TPP, if you want to use TPP in combination with TeX.

1.3 Start

TPP can be invoked from either Workbench by double clicking the respective icon or from CLI.

CLI usage is as follows:

TPP I=ICONIFY/S,C=CONFIG/K,M=MENUS/K,K=KEYS/K,MACROS/K,FILES/M	
Parameter	Description
ICONIFY	open TPP iconified
CONFIG <file>	configuration file
MENUS <file>	menus configuration file
KEYS <file>	keymap configuration file
MACROS <file>	macros configuration file
FILES	files to be loaded

Up to ten filenames may be passed to TPP. The parameters can also be specified as tooltypes of an icon. When invoking TPP you should use the loader TPP:TPP, which notifies the server TPP:TeXtPlusProfessional or starts it, if it is not already running.

Chapter 2

The Menus

2.1 The Structure Of This Manual

First, the meaning of all menus and their associated items are explained in detail. After this the structure of all five document styles, i.e. (business)letter, article book, \LaTeX -article, and \LaTeX -book, are illustrated. Finally, all commands of the ARexx port are described at full length.

2.2 Project

2.2.1 Load

After clicking on this menuitem a filerequester pops up, the current function of which - here: 'Load file' - is displayed in the window-titlebar. The selector is completely dynamic and asynchronous; you may change directories, select files, scroll the file list, etc., while the selector is scanning a directory.

After a file has been chosen, TPP loads it and recognizes automatically if the file is a packed file (PowerPacker), or a plain ASCII-file, if it contains ANSI-control-sequences or if it is a TPP-letter (suffix: '.tpp'). TPP-letters of older versions (suffix '.txp') can still be loaded, too. Multiselection is fully supported, so if you select more than one file in the filerequester, iconified windows are opened on the TPP-screen according to the number of selected files.

If you try to overwrite a modified file, a security-requester will pop up asking you to confirm your decision. The keymap equivalents of the OKAY and the STOP gadget of this and all other requesters are the Return key (OKAY) and Esc (STOP).

2.2.2 Append file

A file is appended to the end of the already existing file. If you want to insert a file at the cursor's current position, you've got to load a block. (see also 2.3.6).

2.2.3 New Window

If sufficient memory is available, a new window is opened. The tabulators, the text-width etc. of the old window are used in the new one, too. TextPlus is able to handle up to 10 windows at the same time.

2.2.4 Load & New Window

The comands of the items 'Load' (see also 2.2.1) and 'New Window' (see also 2.2.3) are executed.

2.2.5 Load Configuration

If you want to reset the settings of TPP to those defined in a configuration-file, you only have to choose this menuitem.

2.2.6 Delete File

The current file is deleted in memory.

2.2.7 Save

The current file is saved under its current filename. The letter-mode determines the format of the saved file. If it is set to ASCII, a plain ASCII-file without any control- sequence is saved, if it is set to ANSI, control-sequences for changes in style or color are saved with the file according to the ANSI-standard. Such files can be loaded by every ANSI-compatible editor or text-display program, e.g. 'MuchMore'. If the letter-mode is set to LETTER, the tabulators, the text-width etc. are saved with the file. The suffix '.tpp' is added to such files.

2.2.8 Save as

After a filename has been chosen, it is saved as described above (2.2.7).

2.2.9 Save & Quit

The commands of the items 'Save' (see also 2.2.7) and 'Quit' (see also 2.2.17) are carried out.

2.2.10 Save Configuration

The global settings of TPP are saved to a configuration-file.

2.2.11 Compile

The gadgets of the compile window have the following meaning:

If 'Parse' is selected, the internal compiler of TPP translates the text of the active window. After that, or if 'Parse' has not been selected, the \TeX -compiler is invoked according to the \TeX -paths definitions (see also 2.4.8). When the internal compiler of TPP is invoked, the document-style (letter, article or book) is deduced from the keyword at the beginning of the text. Furthermore the font-height of the document can be defined. Otherwise the user must explicitly define the format. The compilation can be aborted by pressing Ctrl-C in the log-window. If no log-window was opened, you have to wait until the compiler terminates. If the switch 'Mailmerge' is set, data from a mailmerge-database (see also 2.7.6) is inserted into the text at compile time.

2.2.12 Preview

If the Amiga implementation of \TeX of your choice is Pas \TeX (see also 2.4.1), then Georg Hessmann's previewer 'ShowDVI' is invoked. If you wish, the compiler is run before the ShowDVI is invoked. All the other switches have the same meaning as described under 'Print' (see also 2.2.13). If ShowDVI is already running in the background, the ARexx-script 'TPP:rexx/UpdateShowDVI.tpl' is invoked.

If you choose Amiga \TeX , Radical Eye Software's previewer is started. If Preview is already running in the background, the ARexx-script 'TPP:rexx/UpdatePreview.tpl' is invoked.

2.2.13 Print

The miscellaneous gadgets have the following meaning:

From, To Page	From and to which page is to be printed?				
Copies	PasTeX only. How often should the document be printed?				
Type	PasTeX only. Are only pages with even or odd numbers to be printed, or should TPP not distinguish between odd and even pages?				
HOffset, VOffset	PasTeX only. Set the horizontal and vertical offset of the print-out.				
Reverse	The pages are printed in reverse order.				
Output	Is the document to be printed or should the output be redirected to a file? Furthermore, the output can be send to an IFF- or a postscript-file. For Postscript output you need DVIPS in addition to PasTeX.				
Async	New since 4.10. With the help of this switch you can decide whether you intend to print asynchronously, i.e. in the background, or not. When printing asynchronously, you can only presume working with TPP after the print-out has been finished. The Async-gadget is ignored, if the user selected the mailmerge option or multiple copies (number of copies ≥ 2). When printing asynchronously, the print-out cannot be canceled from within the TPP-environment.				
Date	New since 4.10. If you know, how the Unix-utility 'Make' works, then you already know the meaning of this gadget. Why should you use this switch? Normally, you print out a text not before you have previewed it. If you do not change the settings of the print-requester, after having finished previewing, the whole text will once again be compiled by TPP and TeX. In case of long text-files, this can lead to annoying waiting periods. In order to prevent this you could switch off compiling and parsing. But you would have to switch compiling and parsing on again, before the next print-out/previewing. A more elegant method takes into account the dates of creation of all files that are part of the compiling-printing/previewing-process: If the file of the active window has been modified ('(modified)' appears in the window-titlebar), it is saved before the compilations takes place. Depending on the date entries of the TPP-file (suffix '.tpp') and the TeX- file (suffix '.tex') TPP compiles the TPP-file, if it is the younger one or if the TeX-file does not exist at all. Following the same scheme the TeX-compiler is invoked, if the TeX-file is younger than the dvifile (suffix '.dvi'). After this either printing or previewing follows. If the dvifile is younger than the TPP-file and the TeX-file, parsing and compiling are skipped and printing or previewing is started immediately. TPP behaves as described above, if the date-gadget is switched on. Since 4.10 you can directly send a file to the printer.device, in order to print it out using the printer-driver you set with the help of the respective prefs-program. Of course, you do not need PasTeX for this.				
Options	Depending on the setting of the gadget 'Output' a requester pops up. It contains the following gadgets, if 'Output' is not set to 'PRT:' (PasTeX only): <table> <tr> <td>Printer</td> <td>After scanning 'TeX:config/DVIprint.printers', the configuration file of DVIPrint, all available printer-drivers are displayed. If you want to use the Preferences printer-driver, you have to select the driver 'generic'.</td> </tr> <tr> <td>FontMem</td> <td>What size shall the font memory of ShowDVI/DVIprint have?</td> </tr> </table>	Printer	After scanning 'TeX:config/DVIprint.printers', the configuration file of DVIPrint, all available printer-drivers are displayed. If you want to use the Preferences printer-driver, you have to select the driver 'generic'.	FontMem	What size shall the font memory of ShowDVI/DVIprint have?
Printer	After scanning 'TeX:config/DVIprint.printers', the configuration file of DVIPrint, all available printer-drivers are displayed. If you want to use the Preferences printer-driver, you have to select the driver 'generic'.				
FontMem	What size shall the font memory of ShowDVI/DVIprint have?				

Density	If you print via a Preferences driver, the quality of the print-out is normally set in Preferences. This option provides a way to override the preferences-setting (density).
FontDir	In which additional directory should font-libraries and pk-fonts be searched for?
Resolution	Sets the resolution of the bitmap, if the output is redirected to an IFF-file. The resolution can be defined in two ways: You either specify a number or the sequence number - slash - number. In the first case the horizontal and the vertical resolution are set at the same time. In the second case the first number is interpreted as the horizontal resolution and the second as the vertical resolution, e.g. 120/72.
ShowDVI-Log	Is ShowDVI to create a logfile and what is the name of it?
DVIprint-Log	Is DVIprint to create a logfile and what is the name of it?
Optimize	Should the print-out be optimized? This may enhance the speed of your printer.
Fast	Sets a flag in the printer.device. When set, the speed of the parallel port is increased significantly. If you do not print via the printer.device, then this switch is of no interest for you.
Draft	Sets the quality of the print-out. If you want to print a document just to see what comes out of the printer, this switch might be useful, because the minor quality of the print-out plays an inferior role.
Unidirect.	Is the printer to print uni- or bidirectionally? Bidirectional printing might be faster, but take into account that the quality of the print-out worsens in most of the times.
Landscape	The print-out is rotated by 90 degrees.
Physical	New since 5.00. The physical order of the pages within the dvi-file is respected during the print-out. This is why every page has its own unique number.
No FormFeed	Is the output of a form-feed to be suppressed at the end of the last page?
Preload	All fonts are loaded into memory at the beginning of the print-out, and not when they are needed.
Statistic	A more detailed logfile is created.
Debug	New since 5.00. The logfile is closed after every write access.
Mark	New since 5.00. All used fonts are marked.
SpecialHost	Is SpecialHost to be invoked before the printing starts? If the text to be printed contains IFF-graphics, this switch must be set. SpecialHost calculates the new dimensions of the graphics for DVIprint (see also 2.7.9).

If the gadget 'Output' is set to 'PRT:', the text of the active window is sent to the printer.device. The following gadgets of the options-requester control the print-out:

Pitch	You can select one of four pitches. Do not conceive the given pitches as unchangeable. Escape-sequences may be changed at any time and can be saved along with the
-------	--

configuration. The labels 'NLQ', 'Normal', 'Elite', and 'Condensed' are only used as memo-strings and do not affect the print-out. It has to be pointed out, that you only may use escape-sequences which are understood by the printer.device. These are not necessarily identical to those mentioned in your printer manual. The escape-sequences of the printer.device are listed in the appendix (see page B). Unfortunately, the printer.device does not support all facilities a plenty of printers offer.

What has to be done, if you want to print in a pitch the printer.device does not support?

Fortunately, you can send raw, i.e. unprocessed, data via the printer.device to your printer. This is done by the escape-sequence '27/91/n/34/114', where n is the number of raw data. Let's say your printer manual states that the sequence '27/40/115/50/48/72/27/40/115/52/49/48/49/84' switches on printing in courier (20cpi). The number of raw data is 14. This is why you have to send '49/52' as n to your printer, because 49 is the ASCII-code of 1 and 52 the one of 4. The total sequence reads now:

'27/91/49/52/34/114/27/40/115/50/48/72/27/40/115/52/49/48/49/84'

This sequence was found in the printer manual of the HP DeskJet 500.

Init String	If you want to send some data to the printer before the print-out, you can enter it here.
Left/Right Margin	The left and the right margin are set according to the content of these gadgets.
Pagenumbering	All pages are numbered automatically during the print-out.
Linenumbering	Every line is numbered automatically during the print-out.
Proportional	The file is printed with a proportional character set.
FormFeed	If set, then the output of a form-feed is not suppressed at the end of the last page.
Color 1-4 On/Off	TPP is not a WYSIWYG ¹ word processor. If you want to use other fonts, pitches etc., you have to mark the respective text in one of the four available colors. When encountering the beginning or the end of such a text during the print-out, TPP sends either the on- or off-sequence of the respective color to the printer.device. You may only use sequences, the printer.device understands, as mentioned in subsection 'Pitch'.

Printing can be aborted by pressing Ctrl-C in the log-window.

2.2.14 Iconify

The current window is iconified and opened on either the Workbench or the TPP-screen. The latter takes place if there is at least one uniconified window besides the current one left over on the TPP-screen. Closing down the TPP-screen is very useful for memory hungry multitaskers, because lots of memory is returned to the system.

¹What You See Is What You Get

Clicking the menu-button uniconifies the active window. If the TPP-screen is closed, you can also return to it by pressing the hotkey. Default values are Left + Right Amiga and Y (see also 2.4.1).

After iconifying the last window, an appicon is placed on the Workbench. By double-clicking this icon you get back to TPP.

2.2.15 Help

If AmigaGuide is installed and the documentation of TPP is located within the AmigaGuide search-path, online hypertext-documentation is available on a single keystroke. Otherwise the keymap of TPP is displayed.

2.2.16 About

Informations about the author.

2.2.17 Quit

The active window is closed and, if this window was the last open one, TPP quits. If the text of the active window has been modified in any way, a security requester pops up and asks the user to confirm his decision.

2.3 Block

2.3.1 Mark

Depending on the mark-mode a block can be marked LINewise, WORDwise or BLOCkwise. The marked characters are displayed in reversed colors. The state of the block-mode, which can be left by invoking 'Delete marks' or 'Delete block', is indicated by the button 'BLO' above the ruler. A block can also be marked by moving the mouse while holding down the left mouse button.

2.3.2 Copy

The marked block is inserted into the text at the cursor's current position. You can also copy a block by doubleclicking the left mouse button.

2.3.3 Move

The marked block is moved to the cursor's current position. (mouse equivalent: ctrl-doubleclick)

2.3.4 Paste

The marked block is pasted to the cursor's current position. The difference between 'Copy' and 'Paste' is that 'Copy' inserts the block into the text and 'Paste' pastes over the text.

2.3.5 Delete block

The marked block is deleted. (shift-doubleclick)

2.3.6 Load

A normal file is inserted into the text at the current position of the cursor. The way the file is inserted depends on the mark-mode.

2.3.7 Save

The marked block is saved to disk.

2.3.8 Delete marks

The block markers are deleted for the current window. (alt-doubleclick)

2.3.9 Copy to Tmpfile

If a block is marked, it is saved to disk. The name of this block is set in Prefs/Prefs (see also 2.4.1). If no block is marked 'TP.tmp' is loaded and inserted into the text depending on the mark-mode. This is an easy way to exchange blocks between windows.

2.3.10 Clipboard

2.3.10.1 Copy

The current block is copied as IFF-data to the clipboard.

2.3.10.2 Cut

The current block is copied as IFF-data to the clipboard and after this deleted.

2.3.10.3 Paste

If the clipboard contains usable data, it is loaded as a block to the current position of the cursor.

2.4 Prefs

2.4.1 Prefs

The meaning of the gadgets is as follows:

Write	The insert-mode is either switched on or off. If switched on, it is indicated by the word 'INS' above the ruler. Characters are inserted in the already existing text and do not overwrite the text.
Format	The letter-mode is set to LETter, ANSi, or ASCii. Its only purpose is to determine how a file is to be saved. (see also 2.2.7).
Mark Mode	You can choose whether you want to mark blocks WORdwise, LINewise or BLocKwise. (see also 2.3.1).
TeX	Choose the Amiga implementation of T _E X, you want to work with.
Auto-Div-Mode	<p>TPP was originally designed to be a German word-processor. Due to the occurrence of ultra-long words in German a function called Auto-Div which divides ultra-long words in accordance with German grammar has been provided in order to decrease the number of ultra-long gaps in the text. Auto-Div divides English words in accordance with German grammar, too. This is why you better keep it switched off. If Auto-Div is switched on and if a block is marked and then formatted (left/right-aligned, centered, justification), words which are too long for the respective line are divided in order to avoid large gaps. Auto-Div is not perfect. This is why you should always check the division. Words divided by TPP end with this dash '-' (ASCII 173) and only this dash is accepted by TPP as indicator for a division in case of reformatting.</p> <p>Due to the fact that T_EX also hyphenates automatically, this function is only interesting, if you process texts that are not going to be run through the T_EX compiler.</p>

Backup	If switched on, TPP checks the disk for the same filename you want to save. Providing the file exists, the suffix '.bak' is added to this file in order to protect it from being overwritten.
Save Icon	Should TPP attach an icon to the file when saving?
Space Tabs	When saving eight spaces are replaced by one tab. You can dramatically reduce the size of your file with this option.
Smart <CR>	When hitting return the cursor is either moved to the first column of the following line (off) or under the beginning of the preceding line (on).
Tmp-File	Name of the temporary file, that is used by block operations. (see also 2.3.9).
FReq-Path	Which path is to be the default path of the file-requester of TPP? A single dot ('.') means, that the current directory of the invoking process is to be used.
FReq-Pattern	What pattern shall be used as a filter in the file-requester?
Hotkey	Which hotkeys should reopen iconified windows? (see also 2.2.14)

The syntax of a hotkey is as follows:

[<qualifier> [<qualifier>...]] <key>

Qualifiers

alt	both Alt keys
ralt	right Alt key
lalt	left Alt key
shift	both Shift keys
rshift	right Shift key
lshift	left Shift key
capslock	Caps Lock key
rcommand	right Amiga key
lcommand	left Amiga-key
control	Control key
numericpad	enables the use of a key of the numeric pad

Keys

a .. z, 0 .. 9, etc.	normal keys
f1 .. f10	function keys
up, down, left, right	Cursor keys
help	Help key
del	Delete key
return	Return key
enter	Enter key (must be combined with 'numericpad')
backspace	Backspace key
esc	Escape key
space	Space key
comma	Comma key
upstroke	Upstroke key

Examples

lcommand rcommand y

lalt ralt enter

alt x

Clock With the help of the switches 'Clock', 'Time', 'Date', 'Memory' and 'Reverse' you are able to define the design of the clock.

2.4.2 Screen

Choose your favorite monitor type and the desired resolution of your screen. Furthermore, you can decide, whether TPP is to run on the Workbench or any other public screen.

2.4.3 Colors

The colors of the screen are to be set to which values?

2.4.4 Font

This function enables you to define the font of the text-field. This font has no meaning for the print-out.

2.4.5 Keymap

New since 5.00. An internal command, an internal macro, a TPP-ARexx-command, or an ARexx-macro can be mapped to whatever key combination you like (key plus qualifier, i. e. Ctrl, Shift and/or Alt). In order to tell the system which keys you want to be mapped to a certain function, press the Record button and then enter the respective key combination. If not specified otherwise, TPP looks as a default for the configuration file `TPP:Config/language/TPP.keys`.

2.4.6 Menus

New since 5.00. A menu structure consisting of menus, menuitems and subitems can freely be defined, whereas once again an internal command, an internal macro, a TPP-ARexx-command, or an ARexx-macro can be mapped to every menuitem or subitem. If not specified otherwise, TPP looks as a default for the configuration file `TPP:Config/language/TPP.menu`.

2.4.7 Macros

New since 5.00. If you intend to automate a certain process, but do not want to spend the time needed for creating an ARexx-macro, this function is ideal for you. After having pressed the Record button, TPP from now on protocols all key pressings and menu selections until the menuitem Macros is reselected. All key pressings and menu selections are compiled to a single macro, which you can now give a name. After this you can map this macro to a key or a menuitem and thus invoke it. If not specified otherwise, TPP looks as a default for the configuration file `TPP:Config/TPP.macros`.

2.4.8 T_EX-Paths

Where are all the needed T_EX binaries located? If PasT_EX is installed correctly, you do not have to change anything here. For every document-type TPP provides two alternatives to invoke the compiler; this was implemented in order to easily, for example, switch from the German L^AT_EX version to the American one when compiling a letter.

If a document is to be compiled by TPP's internal compiler, TPP acts as follows:

First, the selected table of special characters is loaded. Every character of the respective document is checked whether it is included in the table, i. e. whether it has to be translated into T_EX-specific

syntax. Furthermore, an '.parse'-file is read in before the compilation. The instructions specified in this file define the main frame of the '.tex'-file that is created by TPP. The '.parse'-files may freely be changed while, of course, obeying to the rules stated in the files in more detail. Therefore the macros and T_EX-styles which are part of the TPP distribution are not meant to be unchangeable but to be an invitation to create your own macros and to design your own documents. If you want to change the '.parse'-files, you should at least be a T_EXpert, to whom [2] is not unknown. The normal 'power-user' should fully be satisfied by the included macros.

2.4.9 Right Edge

The right edge must range between 10 and 255. If a block has been marked, it is formatted according to the new width. Reformatting requires the mark-mode to be set to LINewise.

2.4.10 Left Edge

The left edge must range between 1 and 145. If a block has been marked, it is formatted according to the new left edge, provided the mark-mode is set to LINewise.

2.4.11 Indent Block

If a block is formatted, its first line is indented automatically by the number of columns entered into the 'Indent Block' integer gadget. The indentation value must range between 0 and the half of the text-width.

2.4.12 Page Length

The page length must range within 10 and 255.

2.4.13 Auto-Save

Whenever the time interval elapses (0 to 60 minutes, 0 = OFF), the respective file is saved automatically under its current name.

If Auto-Save is switched on, it is displayed above the ruler (highlighted 'ASA'). Auto-Save might be switched on in every window using different time intervals. Both Auto-Save and the clock need the TeXtPlus-Handler in order to work properly.

2.5 Style

2.5.1 Style

You can set the style to either normal, underlined, **bold** or *italic*. If a block has been marked, it is displayed in the new style.

2.5.2 Justify

The current line is formatted according to the new alignment, or if a block has been marked, the block is formatted provided that the mark-mode is set to LINewise. All comands of this submenu have no effect on the printout.

2.5.3 Color

The drawing color is set to type 1, 2, 3, 4 or normal. If a block has been marked, it is displayed in the new color.

Color 1, 2 and 3 have a special meaning:

Color one is reserved for keywords only, whereas text marked in color 2 appears in the print-out in `typewriter-style`. Text marked in color 3 yields to a *slanted* print-out.

2.6 Find

2.6.1 Find

After the search pattern has been entered, TPP starts to search for it starting from the cursor's current position. If TPP finds the search string, the cursor is placed at the beginning of the respective word. Next and Prev cause TPP to search for the next/previous occurrence of the search pattern. If the casesensitive switch is on, a string is only found if it exactly matches the search pattern, e.g. if the search string is 'Gargleblaster', the string 'Gargleblaster' is found, but 'gargleblaster' is not.

2.6.2 Replace

If a string matches the search pattern, the user is asked whether he wants to replace it by the replace string or not. If 'Continuous' is switched on, TPP continues replacing until the end of the file.

Next and Prev cause TPP to find next/previous occurrence of the search pattern and replace it by the replace pattern.

2.7 T_EX

Keywords may be entered in two different ways: You either type a keyword while color 1 is active, or you select one of the keywords that are displayed when clicking one of the items 'Letter', 'Book', or 'Global'. In many cases not only one but a whole group of keywords is inserted into the text, if those keywords require one another (e. g. `TeX` and `End`). Keywords must, if not marked with a star (*), start in the first column, in order to be recognized by the compiler.

2.7.1 Letter

The following keywords must only be used in a letter:

Letter	The document to be compiled is a letter. This keyword must appear at the very beginning of a document.
Businessletter	The document is a businessletter.
Sender	The sender of the letter may contain a name and five further lines.
StandardSender	This is the rather boring alternative to the normal sender, but it abides by the DIN 5008 standard.
Retouraddress	This keyword causes two folding marks and the sender's address to be set. If you use envelopes with windows this keyword is very helpful.
Address	What is the addressee's name?
YourRef	Must only be used in business-letters: Your-Ref entry of the format-line.
MyRef	Must only be used in business-letters: My-Ref entry of the format-line.
DirectDialing	Must only be used in business-letters: The part of the telephone number that precedes your number.
Telephone	Must only be used in business-letters: Your phone number.
Place	Must only be used in business-letters: Place, where the letter is written. The date is inserted automatically (dd/mm/yyyy).

Subject	What is the subject of the letter?
Opening	One line is reserved for the opening.
Closing	One line is reserved for the closing.
Signature	Specify your signature.
Enclosure	What did you enclose to the letter?
CarbonCopy	You will also read this letter?
PS	The postscriptum may contain paragraphs.
List	This keyword specifies the beginning of a twocolumn, staggered list, like the one you are already reading. The widest key of the list must appear behind the keyword 'List', in order to tell T _E X how wide the left column has to be. Normally, staggered lists are used to enumerate some terms in the left column and to explain them in the other column. This can be achieved by entering the keywords Key (*) and Item (*). After Key you must specify only one word or a short sequence of words whereas the description (Item) can be longer and may contain paragraphs.
NList	A special list is the numbered list. In contrast to List the width of the left column and the keyword ' Key ' must not be specified, because T _E X already knows the width. Numbering starts from 1. Lists are allowed to be nested up to a depth of six lists. If lists are nested on every new level the way of numbering items changes (e. g. 1. (a) i. A.).
LEnd	A staggered list must end with a LEnd .
Indent	The following text, which may contain paragraphs, is indented a little bit. The indentation must be closed with an End .

2.7.2 Book

The following keywords must only be used in a book:

Book	The document to be compiled is a book. This keyword must appear at the very beginning of a document.
TitlePage	The keywords TitlePage , Release , Author , and Abstract define the lay-out of the title page. TitlePage allows you to specify a one line title and Release to place additional information on the title page. The Release line is a bold, centered line; this is why it must not be longer than one line. Author is reserved for the names of the author or the group of authors and may be longer than one line. Finally, you have the opportunity to write a short summary about your document and to place it as an Abstract in the lower half of the title page.
Titleline	If you do not want to waste a whole page for the title, you may use this keyword to specify one or more title lines, that may contain paragraphs.
Section	By using the keywords Section , SubSection , and SubSubSection you can structure a text. All sections are automatically numbered. Behind the sections-keywords you may specify the title of the respective section.
Contents	This keyword induces T _E X to automatically create a table of contents of all sections. During the compilation the titles of all sections and their pagenumbers are written to a '.toc'-file. Due to this you have to compile a text two times in order to be absolutely sure you have the correct table of contents. Normally, you only have to do this when you completed your document. It is important to know that you do not have to care for the table of contents at all. Everything is done automatically.

- Appendix** Starting from now all sections begin with a capital letter.
- Box** The following text which may contain paragraphs is set in a box. **End** quits boxing.
- Label (*)** In a longer document you cannot do without references to topics dealt with at other places but located within the same document. A place you want to refer to from somewhere else in the document is marked with the sequence '**Label** name of the label **End**'. Usually, you want to refer to either the pagenummer ('**PRef** (*) name of the label **End**') or to the number of the respective section ('**CRef** (*) Name der Label **End**'). It is easy to understand, that doing all the references manually results in great pain, because during the creation of a document the section-numbers and even more the pagenumbers are subject of change. As described under 'Contents', you habe to compile your document two times in order to be sure that all labels are refered to correctly.
- FRef (*)** **FRef** is used to refer to a figure (see also 2.7.9) and
- TRef (*)** **TRef** to refer to a table (see also 2.7.8).
- List** (see also 2.7.1)
- NList** (see also 2.7.1). Here you have to specify the width of the left column. This is done by telling \TeX the maximum number of ciphers to be used ('1' for max. 9, '2' for max. 99, and '3' for max. 999 items).
- AList** The list is numbered alphanumerically (a), b), c) etc.). The width must not be specified.
- RList** Numbering is done with small roman figures. Here, too, the width must not be specified.

2.7.3 Global

The following keywords may be used in any document (letter, article, larticle, book, lbook):

- Article** The document to be compiled is an article. This keyword must appear at the very beginning of the document and must, of course, only appear within an article.
- Begin** This keyword indicates the start of the document's main part. **Begin** must appear in every document and must be closed with a corresponding **End**.
- TeX (*)** If certain \TeX instructions should be inserted unchanged into the '.tex'-file, you can accomplish this with the sequence '**TeX** instructions etc. **End**'.
- Date (*)** The current date with a full-length month-name is inserted into the text.
- Today (*)** The current date in short format (dd/mm/yy) is inserted into the text.
- Leftaligned** Normally, the whole text is justified as a block. If you wish to align several paragraphs to the left, just insert **Leftaligned** at the top of the paragraphs and **End** at the bottom. If only a single line has to be justified to the left, then you should use **Left**. Analogously, you get rightaligned and centered justification by using **Rightaligned** or **Right**, **Centered** or **CenterLine**.
- Headline** With the help of this keyword you are able to define a line which appears at the top of every page. **Footline** applies analogously to the bottom of every page.
- Footnote (*)** Inserting '**Footnote** comment **End**' into the text is the way \TeX provides to comment on a term with a footnote. During the compilation all footnotes are automatically numbered and are set at the bottom of the respective pages. The only important thing to know is that you do not have to care for the correct placing and numbering at all. All your work is done by inserting the footnote into the text.
- NewPage** After this keyword the subsequent text is set on a new page.

2.7.4 L^AT_EX

Since 4.10 TPP fully supports Leslie Lamport's macro-package L^AT_EX . The styles book (LBook) and article (LArticle) are now available.

If a keyword may only be used in a LBook or LArticle environment, it is mentioned at the proper place. The following keywords are at your disposal:

LBook	The document is a L ^A T _E X-book. This keyword must appear at the very beginning.
LArticle	The document is a L ^A T _E X-article. This keyword must appear at the very beginning.
TitlePage	The keywords TitlePage, Author, and Date define the lay-out of the title page. TitlePage allows you to specify a multiple line title. Author is reserved for the names of the author or the group of authors and may be longer than one line. With the help of Date you can place the date of creation of the file on the titlepage. Finally, you have the opportunity to write a short summary about your document and to place it as an Abstract on the titlepage. The keyword Abstract may only be used within a LArticle environment and not within a LBook one.
RawTitlePage	Instead of using the L ^A T _E X commands for arranging the titlepage, you can design it on your own behind this keyword.
Contents	See subsection 2.7.2.
Chapter	A L ^A T _E X-book can be divided by the following six keywords, which are enumerated in their priority: Chapter, Section, SubSection, SubSubSection, Paragraph, and SubParagraph. Normally, the title of each chapter, section or paragraph is specified directly behind the keyword. Every division-keyword creates a running number that is positioned in front of the title. Starting from the chapter down to the subsubsection all numbers and titles are added to the table of contents. Within the environment L ^A T _E X-article the keyword Chapter may not be used.
Appendix	Starting from now all chapters (LBuch) - sections (LArticle) respectively - begin with a capital letter. In contrast to the environment Book the keyword End has to be specified at the end of the appendix.
Label	See subsection 2.7.2. In contrast to the environment Book the keywords used for referring to sections, tables and figures are replaced by a single keyword, namely Ref. The references are written to an auxiliary file (suffix '.aux').
List	See subsection 2.7.1.
NList	See subsection 2.7.1.
Box	See subsection 2.7.2.
PNumArabic	From now on all pages are numbered with arabic ciphers. PNumroman switches to small roman ciphers, PNumRoman to capital roman ciphers, PNumalpha to small letters, and PNumAlpha to capital letters.
Bibliography	This keyword starts the definition of the bibliography, which is closed by BibEnd. Scientific publications very often contain a bibliography, the entries of which are referred to from within the text, e.g. by numbering all entries of the bibliography and referring to the respective number. At creation time of the publication the bibliography is often subject of change. It would be quite laboriously, if the whole file had to be scanned in order to adjust the references after a change of the bibliography. TPP supports the L ^A T _E X command set, which allows both, formatting of the bibliography and automatical adjusting of references.

An entry of the bibliography is specified by `BibItem` and a subsequent name of the item. The following lines may be arranged according to your own wishes. Normally, an entry of a bibliography contains the name of the author, the title, the publishing company and the year of publication. An entry ends at the beginning of an ensuing entry (`BibItem`) or at the end of the bibliography (`BibEnd`). All entries are numbered starting from 1. All numbers are embraced by two brackets.

The command sequence 'Cite name of the bibitem End' refers to an entry of the bibliography. The respective number embraced by two brackets is inserted into the text.

In order to get correct references the respective file has to be compiled twice.

Index

Just like the table of contents an index can be created, too. The command sequence 'Index entry End' writes an index entry to the index file (suffix '.idx'). All these entries are sorted alphabetically by the program `MakeIndex` and are written to a sorted index file (suffix '.ind'). The command `PrintIndex` causes T_EX to insert the sorted index into the text. This is normally done at the end of a file.

In order to create a correct index the respective file has to be compiled twice.

The argument of the `Index` command controls the design of the index:

Page	Entry
iv	<code>Index Mailmerge End</code>
3	<code>Index ARexx!ToMouse End</code>
7	<code>Index ARexx!WDelLeft End</code>
15	<code>Index Installation (End</code>
17	<code>Index Installation) End</code>
20	<code>Index Block@Any text End</code>

The above index entries create the following index:

```
TeX~ Item ARexx,
TeX~ Item   ToMouse, 3
TeX~ Item   WDelLeft, 7
TeX~ Item Any text, 20
TeX~ Item Installation, 15-17
TeX~ Item Mailmerge, iv
```

The exclamation mark produces a subentry. You may only use subdivisions down to a depth of three, i. e. there are no subsusbubentries available. Every entry is sorted independently. You can create a from-to-page-entry by appending the characters '|(' (from) and '|)' (to). If you specify an at-sign (@), the string in front of the at-sign determines the alphabetical position of the entry, while the string behind the at-sign produces the text of the entry. Depending on the table of special characters used for parsing (e. g. `TPP:parse/specchars1.parse`), the character '|' eventually has to be replaced by 'TeX | End'. Further informations on `MakeIndex` can be found in a separate dvifile, which is part of this distribution.

2.7.5 User

New since 4.10. From now on user defined keywords may be used within the text. Of course, these keywords may not be identical to already existing ones. The definition of a keyword is to be appended to the respective parse-file. The way an entry has to look alike is described within the

parse-files. If a definition of a keyword expects input (<insert>), the keyword has to be accompanied by a subsequent **End** when used in the text. You may define an infinite number of keywords, while only the first 24 keywords can be selected via the mouse. In the file 'TPP:parse/lbook.parse' you can find some examples of user defined keywords.

2.7.6 Mailmerge

TPP's mailmerge function is very useful if you want to write letters that only differ in the address and the opening. Each of the 10 mailmerge marks can be inserted into the text as often as you like (see also MM-Mark). During the print-out these mailmerge-marks are replaced by data read in from a separate mailmerge file. Due to the fact that lots of data processors save their data bases in different formats, you can define the format by yourself. If the separator 'Backslash' ('Komma') is selected, this means that the length of every data set is one line and that every datum is separated from the next one by a backslash '\ ' (comma ',').

Excerpt from a mailmerge file:

```
Meyer\James\12 Broadway\Paris, Tennessee
Miller\Arthur\325 Randall Drive\Paris, Texas
Robinson\Anthony\15 Willowfield Crescent\Seattle, Washington
Smith\Michael\48 Parkway\Denver, Colorado
Taylor\John\77 Sunset Strip\Beverly Hills, California
etc.
```

If a separator different to those mentioned above are used in your mailmerge file, you can define the separator by yourself, too. The separator may also consist of a string of characters. If 'Return' was selected, every line contains one datum. This is why every data set is ten lines long. The memo-strings are only meant to help you remember the meaning of a datum within a data set. The pattern-strings define which data sets should not be filtered out. A data set is not filtered out, if every datum of a data set matches its pattern.

Following wildcards are available:

- ? Matches a single character.
- # Matches the following expression 0 or more times.
- (ab|cd) Matches any one of the items separated by '|'.
 - ~ Negates the following expression. It matches all strings that do not match the expression (e. g. ~(foo) matches all stirngs that are not exactly "foo").
- [abc] Character class: matches any of the characters in the class.
- [^abc] Character class: matches any of the characters not in the class.
- [a-z] Character range (only within character classes).
- % Matches 0 characters always (useful in "(foo|bar|%)").
- * Synonym for "#?".

If the first pattern-string was set to 'M*', the data sets 'Meyer\...' and 'Miller\...' of the above mailmerge file were not filtered out.

The name of the mailmerge file is to be entered into the datafile string-gadget.

Before the compilation or the print-out a window is opened and the mailmerge file is loaded. The window is only opened, if a window with the name of the mailmerge file does not already exist. Subsequently the selected data sets are displayed and you can decide whether you want to modify or skip the data, or if you want to print all data sets without being bothered again with a requester asking for confirmation.

2.7.7 MM-Mark

One of the ten mailmerge marks (`Mail0` ... `Mail9`) is inserted into the text.

2.7.8 Table

Shopping list			
Product	Quantity	Price in DM	Price in \$
Computer	1	3000,-	1700.00
Modem	1	900,-	500.00
Disks	100	90,-	50.00
Aspirin	10	20,-	9.00
Total		4010,-	2259.00

Table 2.1: An example table

TPP provides a simple way to typeset complex tables in T_EX . A table created by TPP normally consists of a title, some columns, the names of those columns and finally of a caption. When creating a new table all fields of the table window are empty at the beginning. If the cursor is located within an already existing definition of a table when invoking the table window, all fields are filled out using the information stored in the text. You may define up to 20 columns and for every column one of five different alignments: Besides left- and rightaligned and centered justification comma- and dot-alignment are available, i. e. in those columns all entries are aligned to a comma or a dot (useful for columns that contain lots of numbers). The entries of one row are separated by a backslash (`'\'`).

Following keywords may be used within a table definition:

Table	This keyword marks the beginning of a table, which must be closed by an End .
Title	What is the table's title?
Caption	A comment is placed at the bottom of the table. Tables used in a book are automatically numbered. If <code>'TLabel name of the label'</code> appears within the table definition, you can refer to the table using TRef (see also 2.7.2).
Width	You only have to specify the width of the whole table. T _E X automatically finds out the correct width of every column.
Horizontal	Is every row of the table to be separated by an horizontal rule? Analogous: Vertical .
Border	The table is framed.
OnlyNames	Only the names of the columns are separated from the other rows by a horizontal rule. (Horizontal must be switched off for this purpose.)
Columns	The definition of the columns' names follows. The alignment of the respective column (TabLeft , TabRight , TabCentered , TabComma , TabDot) is followed by its name.
Entries	The entries of a row are separated by a backslash (<code>'\'</code>).

Of course, you do not have to enter those keywords manually, because TPP automatically inserts them into the text after leaving the table window.

2.7.9 Graphics

If you wish to illustrate your document with IFF-graphics, just insert a graphics definition into the text. The picture is then loaded during the print-out or while previewing. For this purpose 'SpecialHost' must run in the background (see also 2.2.13). During compilation \TeX reserves space for the picture.

It is not possible to let text flow around the picture, yet. Following keywords are available and like the definition of tables (see also 2.7.8) they do not have to be entered manually.

Graphics	This keyword marks the beginning of a picture, which must be closed by an End .
IFFFile	Name of the iff-file.
Caption	A comment is placed at the bottom of the picture. Pictures used in a book are automatically numbered. If ' FLabel name of the figure' appears within the graphics definition, you can refer to the figure using FRef (see also 2.7.2).
Width	What dimensions (Width,Height) shall the picture have?
DefColors	Are the default colors to be used? If not, then you can specify (Red, Green, Blue, Bright) the colors and the brightness of the picture. The values must range from 0 to 15.
Mode	Is the picture to be printed as a black and white (BW), a coloured (Colour) or as an HAM -picture?
Position	Shall the picture be printed leftaligned (PosLeft), rightaligned (PosRight) or centered (PosCenter)?

Finally, I have to point out, that the portability of a '.dvi'-file gets lost by including an IFF-picture into a document.

Figure 2.1: Fractal graphics

2.8 Extras

2.8.1 Files

A filerequester pops up and due to the selected menuitem all files are either deleted or renamed. Of course, multiselect is supported, too. Further you are able to create a directory. 'Change filename' only affects the name of the active window.

2.8.2 Undo

Undo the current line.

2.8.3 Date

The current date is inserted at the cursor's position.

2.8.4 Set Mark

One of ten marks is initialized. TPP remembers both column and line of the cursor's current position.

2.8.5 Go to Mark

If the mark has already been set (see also 2.8.4), TPP jumps to the marked place. 'Set Mark' and 'Go to Mark' provide an easy way to find one's way even in larger documents.

2.8.6 Go to ...

After the string which is to be searched for has been entered, TPP is searching for it at the beginning of every line and only finds the string if it starts in the first column. This routine is quite useful for C-programmers, because one can find modules, e.g. `main()`, very fast especially in long sourcecodes.

2.8.7 Go to line x.

TPP jumps to line x. This routine, too, is useful for programmers, because compilers tell the user the line in which an error has occurred.

2.8.8 Extern

This is a menu, which has been created for the invocation of the ARexx macros that are also part of the distribution. As an example I would like to mention the macro 'SDVI to Front' / 'Preview to Front'. This macro switches the previewer into the foreground, if it is already running in the background. If you map 'F10' / '1' in the previewer analogously to the macro 'TPPToFront.sd' / '1.preview' (also part of the TPP-package), you can read a document using the previewer and then switch back to TPP to correct the typing-errors you found without having to use the mouse.

2.9 Creating Documents

In order to create a document you normally load a form sheet, which you only have to fill out and extend according to your wishes. Due to the fact that you learn at most by examples, there are lots of them for all document-types.

By inserting an empty line you start a new paragraph.

In general a letter consists of a sender, an address, a subject, an opening and finally of a closing. A postscriptum and comments on enclosures and carbon copies may be added at the bottom. I assume that understanding how to create a letter is not too difficult.

The lay-out of a book follows the scheme:

- **Titlepage** with specification of the author and the title. Eventually the titlepage also contains a summary of the content.
- **Table of contents**, which contains the titles of all sections and subsections.
- **Sections**, which are automatically numbered. Every sections may be divided in further subsections and subsubsections.

- **Appendix**, e.g. for references. If the appendix consists of several sections, these sections are numbered with capital letters.
- Every page, except the titlepage, has head- and footlines, that contain informations about the page number and the title of the section that is started or continued on the respective page.

Articles are the most shapeless type of document, which can be created with TPP. Normally, you choose your document to be an article, if you neither want to write a letter nor a book.

Since 4.10 \LaTeX -books and -articles can be created with the help of TPP. The general structure of those document-types follows the same scheme as described above. But in contrast to the above description of designing books a bibliography and a sorted index (using `MakeIndex`) can automatically be created. Furthermore, a text can be divided in chapters in the `LBuch` environment.

Chapter 3

The ARexx-Port Of TPP

ARexx is the implementation of the programming language Rexx¹. Rexx was developed by Michael F. Cowlshaw on an IBM VM/SP under VM/CMS and has been ported to the Amiga by William S. Hawes. ARexx distinguishes itself from other languages by its powerful command interface, i.e. ARexx programs can issue commands to external programs that provide a suitable command interface and thus can have an impact on the behaviour of these programs. Since V3.0 TPP has an ARexx-interface and provides more than 100 different commands. The hostname of TPP is 'TextPlus'. By pressing 'Esc' you get into the command line. ARexx macros for TPP may be invoked in two ways: Implicitly, by specifying the name of a macro on TPP's command line, followed by any arguments to be passed to the macro. The macro name must not be the same as an internal TPP command. Note that the entire command string is passed to the ARexx interpreter "as is", with the case of any arguments, special characters, etc. being preserved. By hitting a function key or choosing a menu item of the 'Extern' menu. The implicit form of a macro command line, with any arguments, may be mapped to a key or a menu item. If a function key is pressed when in command line mode the command mapped to the function key is not executed but displayed in the command line. ARexx macros for TPP should be given the filename extension '.t`pl`'. Furthermore they must reside in the current directory, or the directory 'rexx:' is assigned to. The current directory is searched first, followed by the "rexx:" directory.

3.1 ARexx-Returncodes

ErrorCode	Severity	Meaning
0	10	no window available for execution of sent command
1	4	error while loading
2	4	error while printing
3	4	couldn't open window
4	4	error while saving
5	4	couldn't find window
6	4	error while executing 'Execute'
7	1	operation on block failed
8	1	find / replace failed
9	1	changing of prefs failed
10	1	operation on window failed

Table 3.1: The ARexx-Returncodes of TPP

¹Restructured EXtended eXecutor

3.2 ARexx-Commands

3.2.1 Activate

NAME `Activate Filename`

FUNCTION `Activate window with the name 'filename'.`

EXAMPLE `Activate MAN:TP-ARexx!.doc`

RETURN `In case of failure: ErrorCode 5 Severity 4`

SEE ALSO

3.2.2 Append

NAME `Append Filename`

FUNCTION `Append the file 'filename' to the end of the current file.`

EXAMPLE `Append RAM:TP.tmp`

RETURN `In case of failure: ErrorCode 4 Severity 1`

SEE ALSO `BLoad`

3.2.3 AutoDivMode

NAME `AutoDivMode switch`

FUNCTION `Switch auto-div-mode on or off.`

EXAMPLE `AutoDivMode on`

`AutoDivMode off`

RETURN

SEE ALSO `BackUpMode, BMode, InsertMode, Lettermode`

3.2.4 AutoSave

NAME `AutoSave n`

FUNCTION `Save the current file every n minutes.`

EXAMPLE `AutoSave 10`

`AutoSave 0 (schaltet AutoSave aus)`

RETURN

SEE ALSO

3.2.5 BackTab

NAME BackTab

FUNCTION Jump to previous tabulator.

EXAMPLE BackTab

RETURN

SEE ALSO Tab, BInsTab, BDelBackTab, InsTab, DelBackTab, BLeft, BRight

3.2.6 BackUpMode

NAME BackUpMode switch

FUNCTION Switch backup-mode on or off.

EXAMPLE BackUpMode on
BackUpMode off

RETURN

SEE ALSO AutoDivMode, BMode, InsertMode, Lettermode

3.2.7 BBottom

NAME BBottom

FUNCTION Go to the last line of the current block.

EXAMPLE BBottom

RETURN

SEE ALSO BTop

3.2.8 BCopy

NAME BCopy

FUNCTION Copy the current block to the cursor's current position.

EXAMPLE BCopy

RETURN

SEE ALSO BMove, BPaste, BDelete

3.2.9 BCopyTmp

NAME BCopyTmp

FUNCTION If a block has been defined, it will be saved as TP.tmp. If not, TP.tmp will be loaded and inserted at the cursor's current position.

EXAMPLE BCopyTmp

RETURN

SEE ALSO BLoad, BSave

3.2.10 BDelBackTab

NAME BDelBackTab

FUNCTION Move the marked block to the previous tabulator.

EXAMPLE BDelBackTab

RETURN

SEE ALSO BInsTab, Tab, BackTab, InsTab, DelBackTab, BLeft, BRight

3.2.11 BDelete

NAME BDelete

FUNCTION Delete the marked block.

EXAMPLE BDelete

RETURN

SEE ALSO BCopy, BMove, BPaste, Unblock

3.2.12 BIndent

NAME BIndent n

FUNCTION Indent the first line of the marked block by n columns.

EXAMPLE BIndent 3

BIndent 0

RETURN In case of failure: ErrorCode 9 Severity 1

SEE ALSO

3.2.13 BInsTab

NAME BInsTab

FUNCTION Move the marked block to the next tabulator.

EXAMPLE BInsTab

RETURN

SEE ALSO BDelBackTab, InsTab, DelBackTab, Tab, BackTab, BLeft, BRight

3.2.14 BLeft

NAME BLeft

FUNCTION Move the marked block one column to the left.

EXAMPLE BLeft

RETURN

SEE ALSO BRight, BInsTab, BDelBackTab, InsTab, DelBackTab, Tab, BackTab

3.2.15 BLoad

NAME BLoad Filename

FUNCTION Load 'filename' and insert it at the cursor's current position.

EXAMPLE BLoad MAN:TP-ARexx.doc

RETURN In case of failure: ErrorCode 1 Severity 4 or ErrorCode 7 Severity 1

SEE ALSO BSave, BCopyTmp

3.2.16 Block

NAME Block

FUNCTION Mark start or end of block.

EXAMPLE Block (start marked)

Block (end marked)

Block (new start marked)

RETURN

SEE ALSO Unblock

3.2.17 BMode

NAME BMode mode

FUNCTION Set mark-mode to mode.

EXAMPLE BMode word

BMode line

BMode block

RETURN

SEE ALSO AutoDivMode, BackUpMode, InsertMode, LetterMode

3.2.18 BMove

NAME BMove

FUNCTION Move the marked block to the cursor's current position.

EXAMPLE BMove

RETURN

SEE ALSO BCopy, BPaste, BDelete, Unblock

3.2.19 Bottom

NAME Bottom

FUNCTION Jump to bottom of file.

EXAMPLE Bottom

RETURN

SEE ALSO Top, ScreenBottom, ScreenTop

3.2.20 BPaste

NAME BPaste

FUNCTION Paste the marked block at the cursor's current position.

EXAMPLE BPaste

RETURN

SEE ALSO BCopy, BMove, BDelete, Unblock

3.2.21 BRight

NAME BRight

FUNCTION Move the marked block one column to the right.

EXAMPLE BRight

RETURN

SEE ALSO BLeft, BInsTab, BDelBackTab, InsTab, DelBackTab, Tab, BackTab

3.2.22 BS

NAME BS

FUNCTION Backspace (delete the char to the left of the cursor).

EXAMPLE BS

RETURN

SEE ALSO Del

3.2.23 BSave

NAME BSave Filename

FUNCTION Save the marked block to a file.

EXAMPLE BSave RAM:Block

RETURN In case of failure: ErrorCode 4 Severity 4 or ErrorCode 7 Severity 1

SEE ALSO BLoad, Load, Save

3.2.24 BTop

NAME BTop

FUNCTION Jump to the top of the marked block.

EXAMPLE BTop

RETURN

SEE ALSO BBottom

3.2.25 CallMacro

NAME CallMacro

FUNCTION Calls an internal macro by its name.

EXAMPLE CallMacro Name_of_a_macro

RETURN

SEE ALSO

3.2.26 Casesensitive

NAME Casesensitive switch

FUNCTION Switch 'casesensitive' on or off.

EXAMPLE Casesensitive on
Casesensitive off

RETURN Find, Replace, Next, NextR, Prev, PrevR, Continuous

SEE ALSO

3.2.27 ChFilename

NAME ChFilename Filename

FUNCTION Change the name of the current file.

EXAMPLE ChFilename MAN:TP-ARexx.doc

RETURN

SEE ALSO GetName

3.2.28 Clear

NAME Clear

FUNCTION Delete the current file.

EXAMPLE Clear

RETURN

SEE ALSO

3.2.29 ClearTab

NAME ClearTab n

FUNCTION Delete the tabulator of column n.

EXAMPLE ClearTab 7

RETURN

SEE ALSO SetTab, ClearTabs

3.2.30 ClearTabs

NAME ClearTabs

FUNCTION Delete all tabulators.

EXAMPLE ClearTabs

RETURN

SEE ALSO SetTab, ClearTab

3.2.31 ClipCopy

NAME ClipCopy

FUNCTION The current block is copied as IFF-data to the clipboard.

EXAMPLE ClipCopy

RETURN

SEE ALSO ClipCut, ClipPaste

3.2.32 ClipCut

NAME ClipCut

FUNCTION The current block is copied as IFF-data to the clipboard and after this deleted.

EXAMPLE ClipCut

RETURN

SEE ALSO ClipCopy, ClipPaste

3.2.33 ClipPaste

NAME ClipPaste

FUNCTION If the clipboard contains usable data, it is loaded as a block to the current position of the cursor.

EXAMPLE ClipPaste

RETURN

SEE ALSO ClipCopy, ClipCut

3.2.34 Clock

NAME `Clock switches`

FUNCTION Change the clock-display.

EXAMPLE `Clock on` (switch clock on)
`Clock off` (switch clock off)
`Clock on time date` (switch time and data-display on)
`Clock off memory` (switch memory-display off)
`Clock on time date off reverse` (switch time and date-display on and reverse-display off)

RETURN

SEE ALSO

3.2.35 Color

NAME `Color n`

FUNCTION Set the current drawing color to n.

EXAMPLE `Color 1`
`Color 2`
`Color 3`
`Color 4`

RETURN

SEE ALSO `Style, Justify`

3.2.36 Continuous

NAME `Continuous switch`

FUNCTION Switch 'Continuous'-mode on or off.

EXAMPLE `Continuous on`
`Continuous off`

RETURN

SEE ALSO `Casesensitive, Find, Replace, Next, NextR, Prev, PrevR`

3.2.37 CR

NAME `CR`

FUNCTION Carriage return at the cursor's current position.

EXAMPLE `CR`

RETURN

SEE ALSO

3.2.38 DelBackTab

NAME DelBackTab

FUNCTION Delete the line up to the previous tabulator.

EXAMPLE DelBackTab

RETURN

SEE ALSO InsTab, BDelBackTab, BInsTab, BRight, BLeft, Tab, BackTab

3.2.39 Del

NAME Del

FUNCTION Delete the character under the cursor.

EXAMPLE Del

RETURN

SEE ALSO BS

3.2.40 DelLeft

NAME DelLeft

FUNCTION Delete beginning of line.

EXAMPLE DelLeft

RETURN

SEE ALSO DelRight, DelSpace

3.2.41 DelLine

NAME DelLine

FUNCTION Delete the current line.

EXAMPLE DelLine

RETURN

SEE ALSO UndoDelLine

3.2.42 DelRight

NAME DelRight

FUNCTION Delete rest of line.

EXAMPLE DelRight

RETURN

SEE ALSO DelLeft, DelSpace

3.2.43 DelSpace

NAME DelSpace

FUNCTION Delete all spaces until reaching end of line.

EXAMPLE DelSpace

RETURN

SEE ALSO DelLeft, DelRight

3.2.44 Display

NAME Display string

FUNCTION Display a string in the command-line above the ruler.

EXAMPLE Display 'Hello, World!'

RETURN

SEE ALSO

3.2.45 Down

NAME Down

FUNCTION Move cursor down one line.

EXAMPLE Down

RETURN rc = 1 (Severity), ErrorCode 11, if you are already in the last line
rc = 0 otherwise

SEE ALSO Up, Left, Right, First, Last

3.2.46 Execute

NAME Execute command

FUNCTION Execute a CLI-command.

EXAMPLE Execute C:\NewShell 'CON:////TPPSHELL/AUTO/WAIT/SCREENTextPlus'

RETURN In case of failure: ErrorCode 6 Severity 4

SEE ALSO

3.2.47 Find

NAME Find findstring

FUNCTION Set search pattern to findstring and do a 'Next'.

EXAMPLE Find TextPlus

RETURN In case of failure: ErrorCode 8 Severity 1

SEE ALSO Next, Prev, Casesensitive, Replace, NextR, PrevR, Continuous

3.2.48 First

NAME First

FUNCTION Move to the first column.

EXAMPLE First

RETURN

SEE ALSO Last, Left, Right, WLeft, WRight

3.2.49 GetChar

NAME GetChar

FUNCTION Copy the character under the cursor to the 'result' string. If there's no char under the cursor, 'result' is empty.

EXAMPLE GetChar

Say result

RETURN

SEE ALSO GetString, GetWord

3.2.50 GetColumn

NAME GetColumn

FUNCTION Returns the number of the current column.

EXAMPLE GetColumn

say result

RETURN

SEE ALSO GetScreenWidth, GetScreenHeight

3.2.51 GetFontHeight

NAME GetFontHeight

FUNCTION Returns the screen's font-height.

EXAMPLE GetFontHeight

say 'The screen-font-height is' result 'pixels'

RETURN

SEE ALSO

3.2.52 GetLine

NAME `GetLine`

FUNCTION Returns the number of the current line.

EXAMPLE `GetLine`

```
say 'The cursor is in line' result
```

RETURN

SEE ALSO `GetColumn`

3.2.53 GetName

NAME `GetName`

FUNCTION Copy the name of the current window to the 'result' string.

EXAMPLE `GetName`

```
Say result
```

RETURN

SEE ALSO `ChFilename`

3.2.54 GetScreenLeftEdge

NAME `GetScreenLeftEdge`

FUNCTION Return the left edge of the TPP screen.

EXAMPLE `GetScreenLeftEdge`

```
say result
```

RETURN

SEE ALSO `GetScreenTopEdge`, `GetScreenWidth`, `GetScreenHeight`

3.2.55 GetScreenTopEdge

NAME `GetScreenTopEdge`

FUNCTION Return the top edge of the TPP screen.

EXAMPLE `GetScreenTopEdge`

```
say result
```

RETURN

SEE ALSO `GetScreenLeftEdge`, `GetScreenWidth`, `GetScreenHeight`

3.2.56 GetScreenHeight

NAME `GetScreenHeight`

FUNCTION Returns the height of the TPP-screen.

EXAMPLE `GetScreenHeight`

```
say result
```

RETURN

SEE ALSO `GetColumn`, `GetScreenHeight` `GetScreenLeftEdge`, `GetScreenTopEdge`, `GetScreenWidth`

3.2.57 GetScreenWidth

NAME `GetScreenWidth`

FUNCTION Returns the width of the TPP-screen.

EXAMPLE `GetScreenWidth`

```
say result
```

RETURN

SEE ALSO `GetColumn`, `GetScreenHeight` `GetScreenLeftEdge`, `GetScreenTopEdge`, `GetScreenHeight`

3.2.58 GetScreenName

NAME `GetScreenName`

FUNCTION Returns the public screen name of the TextPlus screen.

EXAMPLE `GetScreenName`

```
say result
```

RETURN

SEE ALSO

3.2.59 GetWindowLeftEdge

NAME `GetWindowLeftEdge`

FUNCTION Returns the left edge of the larges TPP window.

EXAMPLE `GetWindowLeftEdge`

```
say result
```

RETURN

SEE ALSO `GetWindowTopEdge`, `GetWindowWidth`, `GetWindowHeight`

3.2.60 GetWindowTopEdge

NAME `GetWindowTopEdge`

FUNCTION Returns the top edge of the larges TPP window.

EXAMPLE `GetWindowTopEdge`
`say result`

RETURN

SEE ALSO `GetWindowLeftEdge`, `GetWindowWidth`, `GetWindowHeight`

3.2.61 GetWindowWidth

NAME `GetWindowWidth`

FUNCTION Returns the width of the larges TPP window.

EXAMPLE `GetWindowWidth`
`say result`

RETURN

SEE ALSO `GetWindowLeftEdge`, `GetWindowTopEdge`, `GetWindowHeight`

3.2.62 GetWindowHeight

NAME `GetWindowHeight`

FUNCTION Returns the height of the larges TPP window.

EXAMPLE `GetWindowHeight`
`say result`

RETURN

SEE ALSO `GetWindowLeftEdge`, `GetWindowTopEdge`, `GetWindowWidth`

3.2.63 GetString

NAME `GetString`

FUNCTION Copy the current line to the 'result' string.

EXAMPLE `GetString`
`Say result`

RETURN

SEE ALSO `GetChar`, `GetWord`

3.2.64 GetWord

NAME `GetWord`

FUNCTION Returns the word under the cursor. If there is no character under the cursor, 'result' is empty.

EXAMPLE `GetWord`
`Say result`

RETURN

SEE ALSO `GetString`, `GetChar`, `IsWord`

3.2.65 GoTo

NAME `GoTo string`

FUNCTION Go to the line, which starts with 'string'.

EXAMPLE `GoTo main()`

RETURN In case of failure: `ErrorCode 8 Severity 1`

SEE ALSO `GoToLine`, `Find`

3.2.66 GoToLine

NAME `GoToLine n`

FUNCTION Jump to line `n`.

EXAMPLE `GoToLine 35`

RETURN In case of failure: `ErrorCode 8 Severity 1`

SEE ALSO `GoTo`, `Find`

3.2.67 GoToMark

NAME `GoToMark n`

FUNCTION Jumps to mark no. `n` (`n` out of `[1,10]`). The mark must be previously set by `SetMark`.

EXAMPLE `GoToMark 4`

RETURN

SEE ALSO `SetMark`

3.2.68 Iconify

NAME `Iconify`

FUNCTION Iconify the current window. Uniconify will be carried out automatically, if an iconified window receives a command.

EXAMPLE `Iconify`

RETURN

SEE ALSO

3.2.69 InsertDate

NAME `InsertDate`

FUNCTION Insert the current date.

EXAMPLE `InsertDate`

RETURN

SEE ALSO

3.2.70 InsertMode

NAME `InsertMode` switch

FUNCTION Switch 'InsertMode' on or off.

EXAMPLE `InsertMode on`
`InsertMode off`

RETURN

SEE ALSO `AutoDivMode`, `BackUpMode`, `LetterMode`, `BMode`

3.2.71 InsTab

NAME `InsTab`

FUNCTION Insert spaces until reaching the next tabulator.

EXAMPLE `InsTab`

RETURN

SEE ALSO `DelBackTab`, `BInsTab`, `BDelBackTab`, `BRight`, `BLeft`, `Tab`, `BackTab`

3.2.72 IsWord

NAME `IsWord`

FUNCTION Is the cursor located at the beginning of a word?

EXAMPLE `IsWord`

RETURN `rc = 1` (Severity), `ErrorCode 11`, if cursor is not at the beginning of a word.
`rc = 0` otherwise

SEE ALSO `GetWord`

3.2.73 Justify

NAME `Justify alignment`

FUNCTION Justify marked block or current line.

EXAMPLE `Justify left`
`Justify right`
`Justify center`
`Justify block`

RETURN

SEE ALSO `Style, Color`

3.2.74 Last

NAME `Last`

FUNCTION Move one beyond the last non-space in a line.

EXAMPLE `Last`

RETURN

SEE ALSO `First, Left, Right, WLeft, WRight`

3.2.75 Left

NAME `Left`

FUNCTION Cursor left.

EXAMPLE `Left`

RETURN `rc = 1` (Severity), `ErrorCode 11`, if you are already in line 1, column 1.
`rc = 0` otherwise

SEE ALSO `Right, First, Last, WLeft, WRight`

3.2.76 LeftEdge

NAME `LeftEdge n`

FUNCTION Set the left edge to column n.

EXAMPLE `LeftEdge 10`

RETURN In case of failure: `ErrorCode 9 Severity 1`

SEE ALSO `RightEdge`

3.2.77 LetterMode

NAME LetterMode mode

FUNCTION Set letter-mode to mode.

EXAMPLE LetterMode letter
LetterMode ansi
LetterMode ascii

RETURN

SEE ALSO AutoDivMode, BackUpMode, BMode, InsertMode

3.2.78 Load

NAME Load filename

FUNCTION Load a file.

EXAMPLE Load T:TP.tmp

RETURN In case of failure: ErrorCode 4 Severity 1

SEE ALSO BLoad, Append

3.2.79 MoveWindow

NAME MoveWindow dx dy

FUNCTION Move the current window.

EXAMPLE MoveWindow -10 10
MoveWindow 5 -20

RETURN In case of failure: ErrorCode 10 Severity 1

SEE ALSO SizeWindow, NewWindow

3.2.80 NewWindow

NAME NewWindow (filename) (x y dx dy)

FUNCTION Open a new window.

EXAMPLE NewWindow
NewWindow name_of_the_new_window
NewWindow 0 11 640 245
NewWindow name_of_the_new_window 0 11 640 245

RETURN In case of failure: ErrorCode 3 Severity 4

SEE ALSO MoveWindow, SizeWindow

3.2.81 Next

NAME Next

FUNCTION Find next occurrence of search pattern and place cursor at the beginning of the word or send an error-code.

EXAMPLE Next

RETURN In case of failure: ErrorCode 8 Severity 1

SEE ALSO Prev, Find, Replace, NextR, PrevR, Casesensitive, Continuous

3.2.82 NextR

NAME NextR

FUNCTION Find next occurrence of search pattern and replace it by replace pattern or send an error-code. If 'Continuous' is switched on, NextR will carry on replacing until reaching end of file.

EXAMPLE NextR

RETURN In case of failure: ErrorCode 8 Severity 1

SEE ALSO PrevR, Replace, Continuous, Casesensitive, Next, Prev, Find

3.2.83 PageDown

NAME PageDown

FUNCTION Move one page down.

EXAMPLE PageDown

RETURN rc = 1 (Severity), ErrorCode 11, if the last line was reached.
rc = 0 otherwise

SEE ALSO PageUp, ScreenUp, ScreenDown

3.2.84 PageLength

NAME PageLength n

FUNCTION Set the page-length to n lines.

EXAMPLE PageLength 60

RETURN In case of failure: ErrorCode 9 Severity 1

SEE ALSO

3.2.85 PageUp

NAME PageUp

FUNCTION Jumps one page up.

EXAMPLE PageUp

RETURN rc = 1 (Severity), ErrorCode 11, if the first line was reached.
rc = 0 otherwise

SEE ALSO PageDown, ScreenUp, ScreenDown

3.2.86 Prev

NAME Prev

FUNCTION Find previous occurrence of search pattern and place cursor at the beginning of the found word or send an error-code.

EXAMPLE Prev

RETURN In case of failure: ErrorCode 8 Severity 1

SEE ALSO Next, Find, Casesensitive, Continuous, NextR, PrevR, Replace

3.2.87 PrevR

NAME PrevR

FUNCTION Find previous occurrence of search pattern and replace it by replace pattern or send error-code. If 'Continuous' is switched on, NextR will carry on replacing until reaching begin of file.

EXAMPLE PrevR

RETURN In case of failure: ErrorCode 8 Severity 1

SEE ALSO NextR, Replace, Casesensitive, Continuous, Find, Next, Prev

3.2.88 Replace

NAME Replace findstring replacestring

FUNCTION Set search pattern to 'findstring' and replace pattern to 'replacestring' and do a 'NextR'

EXAMPLE Replace TextMinus TextPlus

if rc = 1 then Display 'Unable to find TextMinus'

RETURN In case of failure: ErrorCode 8 Severity 1

SEE ALSO Find, Next, Prev, Casesensitive, NextR, PrevR, Continuous

3.2.89 Quit

NAME Quit

FUNCTION Close the current window and - if it was the last one to be open - quit TPP.

EXAMPLE Quit

RETURN

SEE ALSO SaveAndQuit

3.2.90 RequestInt

NAME RequestInt hailstring'\'definteger

FUNCTION An requester pops up and interactively asks the user to enter an integer. [range: -9999,99999]. You may provide a title line and a default number. Both strings are separated by a '\'\'.

EXAMPLE

1. hailstring = 'Please enter your age!'
 definteger = '23'
 RequestInt hailstring'\'definteger
 if result ~= '''' then
 'Display' 'You are' result 'years old!'
 else
 'Display' 'Abort: STOP has been pressed'
2. /* Invocation with one or no argument is possible, too */
 RequestInt 'This is the title line'
3. RequestInt

RETURN result contains the entered number. result is empty, if the user aborted the requester.

SEE ALSO RequestString

3.2.91 RequestString

NAME RequestString hailstring'\'defstring

FUNCTION A requester pops up and interactively asks the user to enter a string. You may provide a title line and a default string. Both strings are separated by a '\'\'.

EXAMPLE

1. hailstring = 'Please enter your name!'
 defstring = 'Marvin'
 RequestString hailstring'\'defstring
 if result ~= '''' then
 'Display' 'You are so depressing,' result
 else
 'Display' 'Abort: STOP has been pressed'
2. /* Invocation with one or no arguments is possible, too */
 RequestString 'This is the title line'
3. RequestString

RETURN result contains the entered string. result is empty, if the user aborted the requester.

SEE ALSO RequestInt

3.2.92 Right

NAME `Right`

FUNCTION Move cursor right.

EXAMPLE `Right`

RETURN `rc = 1` (Severity), ErrorCode 11, if the last line and the last column have been reached.
`rc = 0` otherwise

SEE ALSO `Left`, `First`, `Last`, `WLeft`, `WRight`

3.2.93 RightEdge

NAME `RightEdge n`

FUNCTION Set the right edge of the current window to column `n`.

EXAMPLE `RightEdge 10`

RETURN In case of failure: ErrorCode 9 Severity 1

SEE ALSO `LeftEdge`

3.2.94 RX

NAME `RX macro args`

FUNCTION Execute a macro and pass arguments to it or not.

EXAMPLE `RX my_first_macro.tpl one two three_arguments`

RETURN

SEE ALSO

3.2.95 Save

NAME `Save (filename)`

FUNCTION Save current file. If no filename was specified, the file will be saved under the current window's name.

EXAMPLE `Save`

`Save DF0:filename`

RETURN In case of failure: ErrorCode 4 Severity 4

SEE ALSO `BSave`, `BCopyTmp`, `SaveAndQuit`

3.2.96 SaveAndQuit

NAME `SaveAndQuit`

FUNCTION Saves the file of the active window and subsequently closes it. If this window was the last one to be open, TPP quits.

EXAMPLE `SaveAndQuit`

RETURN In case of failure: `ErrorCode 4 Severity 4`

SEE ALSO `BSave`, `BCopyTmp`, `Save`, `Quit`

3.2.97 ScreenBottom

NAME `ScreenBottom`

FUNCTION Move to bottom of current screen.

EXAMPLE `ScreenBottom`

RETURN

SEE ALSO `ScreenTop`, `Bottom`, `Top`

3.2.98 ScreenDown

NAME `ScreenDown`

FUNCTION Move one screen down.

EXAMPLE `ScreenDown`

RETURN `rc = 1` (Severity), `ErrorCode 11`, if the last line was reached.
`rc = 0` otherwise

SEE ALSO `PageUp`, `PageDown`, `ScreenUp`

3.2.99 ScreenToBack

NAME `ScreenToBack`

FUNCTION Thrust TextPlus-screen into the background.

EXAMPLE `ScreenToBack`

RETURN

SEE ALSO `ScreenToFront`, `WindowToFront`, `WindowToBack`

3.2.100 ScreenToFront

NAME `ScreenToFront`

FUNCTION Thrust TextPlus-screen into the foreground.

EXAMPLE `ScreenToFront`

RETURN

SEE ALSO `ScreenToBack`, `WindowToFront`, `WindowToBack`

3.2.101 ScreenTop

NAME `ScreenTop`

FUNCTION

EXAMPLE `ScreenTop`

RETURN Move to top of current screen.

SEE ALSO `ScreenBottom`, `Top`, `Bottom`

3.2.102 ScreenUp

NAME `ScreenUp`

FUNCTION Move one screen up.

EXAMPLE `ScreenUp`

RETURN `rc = 1` (Severity), `ErrorCode 11`, if the last line was reached.
`rc = 0` otherwise

SEE ALSO `PageDown`, `PageUp`, `ScreenDown`

3.2.103 SetColors

NAME `SetColors`

FUNCTION Set screen colors.

EXAMPLE `SetColors AAA 000 FFF 57B`

RETURN

SEE ALSO

3.2.104 SetFont

NAME `SetFont fontname fontsize`

FUNCTION Set the font of the current window. Note: Don't use proportional-fonts.

EXAMPLE `SetFont topaz.font 11`

RETURN In case of failure: `ErrorCode 9 Severity 1`

SEE ALSO

3.2.105 SetMark

NAME `SetMark n`

FUNCTION Set the mark no. `n` (range: `[1,10]`). Both line and column are remembered.

EXAMPLE `SetMark 4`

RETURN

SEE ALSO `GoToMark`

3.2.106 SetTab

NAME SetTab

FUNCTION Set a tabulator at column n.

EXAMPLE SetTab 17

RETURN In case of failure: ErrorCode 9 Severity 1

SEE ALSO ClearTab, ClearTabs

3.2.107 SizeWindow

NAME SizeWindow dx dy

FUNCTION Resize window.

EXAMPLE SizeWindow 0 -100 SizeWindow -300 20

RETURN In case of failure: ErrorCode 10 Severity 1

SEE ALSO ClearTab, ClearTabs

3.2.108 Style

NAME Style what

FUNCTION Switch text-style to what.

EXAMPLE Style normal

Style underlined

Style bold

Style italic

RETURN

SEE ALSO Justify, Color

3.2.109 Tab

NAME Tab

FUNCTION Jump to next tabulator.

EXAMPLE Tab

RETURN

SEE ALSO BackTab, InsTab, DelBackTab, BInsTab, BDelBackTab, BLeft, BRight

3.2.110 Top

NAME Top

FUNCTION Move to top of file.

EXAMPLE Top

RETURN

SEE ALSO Bottom, ScreenTop, ScreenBottom

3.2.111 ToMouse

NAME ToMouse

FUNCTION Move cursor to mouse-position.

EXAMPLE ToMouse

RETURN

SEE ALSO

3.2.112 Type

NAME Type string

FUNCTION Enter text as if typed.

EXAMPLE Type 'Hello, World!'

RETURN

SEE ALSO

3.2.113 UnBlock

NAME UnBlock

FUNCTION Clear the block markers for the current window.

EXAMPLE UnBlock

RETURN

SEE ALSO Block

3.2.114 Undo

NAME Undo

FUNCTION Undo changes of the current line.

EXAMPLE Undo

RETURN

SEE ALSO

3.2.115 UndoDelLine

NAME UndoDelLine

FUNCTION Insert the last deleted line.

EXAMPLE UndoDelLine

RETURN

SEE ALSO DelLine

3.2.116 Up

NAME Up

FUNCTION Cursor up.

EXAMPLE Up

RETURN rc = 1 (Severity), ErrorCode 11, if first line was already reached.
rc = 0 otherwise

SEE ALSO Down

3.2.117 WDelLeft

NAME WDelLeft

FUNCTION Delete previous word. If the cursor is in the middle of a word, this word will be deleted.

EXAMPLE WDelLeft

RETURN

SEE ALSO WDelRight

3.2.118 WDelRight

NAME WDelRight

FUNCTION Delete next word. If the cursor is in the middle of a word, this word will be deleted.

EXAMPLE WDelRight

RETURN

SEE ALSO WDelLeft

3.2.119 WindowToBack

NAME WindowToBack

FUNCTION Thrust the current window into the background.

EXAMPLE WindowToBack

RETURN

SEE ALSO WindowToFront, ScreenToBack, ScreenToFront

3.2.120 WindowToFront

NAME WindowToFront

FUNCTION Thrust the current window into the foreground.

EXAMPLE WindowToFront

RETURN

SEE ALSO WindowToBack, ScreenToBack, ScreenToFront

3.2.121 WLeft

NAME WLeft

FUNCTION Move to the beginning of the previous word. If the cursor is in the middle of a word, move to beginning of current word.

EXAMPLE WLeft

RETURN rc = 1 (Severity), ErrorCode 11, if the begin of the first word of the first line was already reached

rc = 0 otherwise

SEE ALSO WRight

3.2.122 WRight

NAME WRight

FUNCTION Move to the beginning of the next word.

EXAMPLE WRight

RETURN rc = 1 (Severity), ErrorCode 11, if the end of the last word of the last last was already reached

rc = 0 otherwise

SEE ALSO WLeft

Appendix A

The Keymap Of TPP

Key	Function
Ctrl-b	mark begin/end of Block
Ctrl-d	Delete mark
Ctrl-f	change mark-mode
Ctrl-c/p/m/x	Copy/Paste/Move/delete block
Ctrl-j	block <-> tp.tmp
Ctrl-a/e/k	Auto-div/insErt/bacKup-mode on/off
Ctrl-r	change letteR-mode
Ctrl-o	indent the first line of the marked bLOck
Ctrl-g	Go to ...
Ctrl-l	go to Line x
Ctrl-y	set left edge
Ctrl-s	set page length
Ctrl-t	display length and end of Text
Ctrl-u	Undo
Ctrl-z	insert date
Ctrl-v	auto-saVe
Ctrl-q	save file, close window and Quit
Ctrl-w	new Window
Ctrl-n	load file & New window
Ctrl-f[1-10]	set mark
Alt-f[1-10]	goto mark
Ctrl-1/2/3/4/5/6/7	letter/book/global/latex/user/table/graphics
Shift-Del	delete line
Shift-CR	insert last deleted line
Ctrl-Bs/Del	delete begin/rest of line
Alt-Bs/Del	delete previous/next word
Shift-B	delete all spaces until reaching end of line
Alt-Tab	move line to next tab-stop or - if a block has been marked - move block to next tab-stop
Ctrl-Tab	delete line to previous tab-stop or move block to previous tab-stop.
Esc	toggle command-line-mode

Key	Function
Shift-Alt-Crsr-Up/Dwn	go to begin/end of the marked block
Shift-Crsr-Up/Dwn	jump one screen up/down
Alt-Crsr-Up/Dwn	jump to begin/end of file
Ctrl-Crsr-Up/Dwn	jump to begin of the previous/next page
Shift-Alt-Crsr-Lft/Rght	jump to begin/end of line
Shift-Crsr-Lft/Rght	same, but won't stop at the end of the screen.
Alt-Crsr-Lft/Rght	jump to begin of the previous/next word
Ctrl-Crsr-Lft/Rght	move block left/right
Rght-Amiga-g/a/d	load/Append/Delete file
Rght-Amiga-w/s	save/Save as ...
Rght-Amiga-x	save & quit
Rght-Amiga-c	Compile
Rght-Amiga-v	preView
Rght-Amiga-p	Print
Rght-Amiga-y	iconifY
Rght-Amiga-q	quit
Rght-Amiga-k/m	load/save block
Rght-Amiga-t	right edge
Rght-Amiga-'	mailmerge
Rght-Amiga-5	insert mailmerge mark
Rght-Amiga-n/u/b/i	style: Normal/Underlined/Bold/Italic
Rght-Amiga-l/e/c/j	justify Left/right, Center, Justification
Rght-Amiga-0/9/8/7/6	Color: normal/one/two/three/four
Rght-Amiga-f/1/2	Find/next/previous
Rght-Amiga-r/3/4	Replace/next/previous
Rght-Amiga-o	change filename

Appendix B

The printer.device Escape-Sequences

Escape-sequence	Function
27/99	Reset
27/35/49	Initialize
27/68	Linefeed
27/69	Return, Linefeed
27/77	Reverse linefeed
27/91/48/109	Normal char set
27/91/51/109	Italics on
27/91/50/51/109	Italics off
27/91/52/109	Underline on
27/91/50/52/109	Underline off
27/91/49/109	Boldface on
27/91/50/50/109	Boldface off
27/91/48/87	Normal pitch
27/91/50/87	Elite on
27/91/49/87	Elite off
27/91/52/87	Condensed fine on
27/91/51/87	Condensed fine off
27/91/54/87	Enlarged on
27/91/53/87	Enlarged off
27/91/54/34/122	Shadow print on
27/91/53/34/122	Shadow print off
27/91/52/34/122	Doublestrike on
27/91/51/34/122	Doublestrike off
27/91/50/34/122	NLQ on
27/91/49/34/122	NLQ off
27/91/50/118	Superscript on
27/91/49/118	Superscript off
27/91/52/118	Subscript on
27/91/51/118	Subscript off

Escape-sequence	Function
27/91/48/118	Normalize the line
27/76	Partial line up
27/75	Partial line down
27/40/66	US char set
27/40/82	French char set
27/40/75	German char set
27/40/65	English char set
27/40/69	Danish char set I
27/40/72	Swedish char set
27/40/89	Italian char set
27/40/90	Spanish char set
27/40/74	Japanese char set
27/40/54	Norwegian char set
27/40/67	Danish char set II
27/91/50/112	Proportional print on
27/91/49/112	Proportional print off
27/91/n/32/69	Set proportional offset
27/91/53/32/70	Auto left justify
27/91/55/32/70	Auto right justify
27/91/54/32/70	Auto full justify
27/91/48/32/70	Auto justify off
27/91/51/32/70	Letter space (justify)
27/91/49/32/70	Word fill (auto center)
27/91/48/122	1/8 inch line spacing
27/91/49/122	1/6 inch line spacing
27/91/n/116	Perf skip n
27/35/57	Set left margin
27/35/48	Set right margin
27/35/56	Set top margin
27/35/50	Set bottom margin
27/91/xx/59/yy/114	Set top and bottom margin
27/91/xx/59/yy/115	Set left and right margin
27/35/51	Clear margins
27/72	Set horizontal tabulator
27/74	Set vertical tabulator
27/91/48/103	Clear horizontal tabulator
27/91/51/103	Clear all horizontal tabulators
27/91/49/103	Clear vertical tabulator
27/91/52/103	Clear all vertical tabulators
27/35/52	Clear all tabulators
27/35/53	Set tabulators
27/91/n/34/114	Next n chars are raw

Bibliography

- [1] Leslie Lamport. \LaTeX – A Document Preparation System. Addison-Wesley Co., Inc., Reading, MA, 1985
[1] is the standard manual for \LaTeX .
- [2] Donald E. Knuth. The \TeX – Book. Addison-Wesley Co., Inc., Reading, MA, 1990. ISBN 0-201-13448-9.
[2] provides a complete description of \TeX
- [3] Michael F. Cowlishaw. The REXX Language. A Practical Approach to Programming. Prentice-Hall 1985.
Comprehensive introduction and description of the Rexx programming language written by the author himself.
- [4] Commodore-Amiga, Inc. Amiga Programmer's Guide to ARexx. Commodore-Amiga, Inc., West Chester, Pennsylvania, 1991.
[4] extensively describes the implementation of the Rexx programming language on the Amiga.

Index

Symbols

I.preview, 21

A

Abstract, 14, 16

Appendix, 15, 16

ARexx, 23

 Activate, 24

 Append, 24

 AutoDivMode, 24

 AutoSave, 24

 BackTab, 25

 BackUpMode, 25

 BBottom, 25

 BCopy, 25

 BCopyTmp, 25

 BDelBackTab, 26

 BDelete, 26

 BIndent, 26

 BInsTab, 26

 BLeft, 26

 BLoad, 27

 Block, 27

 BMode, 27

 BMove, 27

 Bottom, 28

 BPaste, 28

 BRight, 28

 BS, 28

 BSave, 28

 BTop, 29

 CallMacro, 29

 Casesensitive, 29

 ChFilename, 29

 Clear, 29

 ClearTab, 30

 ClearTabs, 30

 ClipCopy, 30

 ClipCut, 30

 ClipPaste, 30

 Clock, 31

 Color, 31

 Continuous, 31

 CR, 31

 Del, 32

 DelBackTab, 32

 DelLeft, 32

 DelLine, 32

 DelRight, 32

 DelSpace, 33

 Display, 33

 Down, 33

 Execute, 33

 Find, 33

 First, 34

 GetChar, 34

 GetColumn, 34

 GetFontHeight, 34

 GetLine, 35

 GetName, 35

 GetScreenHeight, 36

 GetScreenLeftEdge, 35

 GetScreenName, 36

 GetScreenTopEdge, 35

 GetScreenWidth, 36

 GetString, 37

 GetWindowHeight, 37

 GetWindowLeftEdge, 36

 GetWindowTopEdge, 37

 GetWindowWidth, 37

 GetWord, 38

 GoTo, 38

 GoToLine, 38

 GoToMark, 38

 Iconify, 38

 InsertDate, 39

 InsertMode, 39

 InsTab, 39

 IsWord, 39

 Justify, 40

 Last, 40

 Left, 40

 LeftEdge, 40

 LetterMode, 41

 Load, 41

 MoveWindow, 41

 NewWindow, 41

 Next, 42

 NextR, 42

 PageDown, 42

 PageLength, 42

PageUp, 43
 Prev, 43
 PrevR, 43
 Quit, 44
 Replace, 43
 RequestInt, 44
 RequestString, 44
 Right, 45
 RightEdge, 45
 RX, 45
 Save, 45
 SaveAndQuit, 46
 ScreenBottom, 46
 ScreenDown, 46
 ScreenToBack, 46
 ScreenToFront, 46
 ScreenTop, 47
 ScreenUp, 47
 SetColors, 47
 SetFont, 47
 SetMark, 47
 SetTab, 48
 SizeWindow, 48
 Style, 48
 Tab, 48
 ToMouse, 49
 Top, 49
 Type, 49
 Unblock, 49
 Undo, 49
 UndoDelLine, 50
 Up, 50
 WDelLeft, 50
 WDelRight, 50
 WindowToBack, 50
 WindowToFront, 51
 WLeft, 51
 WRight, 51
 Article, 15
 \LaTeX ~, 16
 Async, 5
 Auto-div-mode, 9
 Auto-save, 12

B

Backup-mode, 10
 Bibliography, 16, 57
 Block
 Clipboard, Copy, 9
 Clipboard, Cut, 9
 Clipboard, Paste, 9
 Copy, 8
 Copy from/to tmpfile, 9
 Delete, 8

 Delete marks, 9
 Load, 8
 Mark, 8
 Move, 8
 Paste, 8
 Save, 8
 Bold, 12
 Book, 14
 \LaTeX ~, 16
 Box, 15
 Businessletter, 13

C

Chapter, 16
 Colors, 7
 Compile, 4
 Configuration, 4
 Contents, 14
 Cowlshaw, Michael F., 23

D

Date, 5, 21
 Delete file, 4
 DVIPrint.printers, 5

E

Escape-sequences, 55
 Extern menu, 21

F

File, temporary, 10
 Find, 13
 Footnote, 15
 FormFeed, 7
 François, Nico, ii

G

Graphics, 20

H

H/S-requirements, 1
 Hawes, William, 23
 Headline, 15
 Help, 8
 Hessmann, Georg, ii
 Hotkey, 11

I

Icon, 10
 Iconify, 8
 Indent block, 12
 Index, 17
 Init String, 7
 Insert-mode, 9
 Installation, 1
 Italic, 12

J

Justify, 12
 Center, 12, 15
 Left, 12, 15
 Right, 12, 15

K

Keymap, 11
 Knuth, Donald E., iii

L

Lamport, Leslie, 16
 LArticle, 16
 L^AT_EX-article, 16
 L^AT_EX-book, 16
 LBook, 16
 Left edge, 12
 Letter, 13
 Letter-mode, 9
 Linenumbers, 7
 List, 14, 15
 Load, 3
 Logfile, 6

M

Mailmerge, 18
 MakeIndex, 17
 Margin
 Left, 7
 Right, 7
 Mark-mode, 9
 Menus, 11

N

NewWindow, 3

P

Page length, 12
 Pagenumbering, 7, 16
 Paragraph, 16
 Parse-file, 12, 17
 Pitch, 6
 Postscript, 5
 PowerPacker, 3
 Preview, 4
 Print, 5
 printer.device, 6
 PrintIndex, 17
 PRT:, 6

R

Raw data, 7
 Raw titlepage, 16
 References, 15, 16
 Replace, 13
 Right edge, 12

S

Save, 4
 Save As, 4
 Section, 14, 16
 Set Mark, 21
 Slanted, 13
 Style, 12
 Subparagraph, 16
 Subsection, 14, 16
 Subsubsection, 14, 16
 System-prefs, 11–12

T

Table, 19
 Tabulators, 10
 T_EX-paths, 12
 Titlepage, 14, 16
 TPPToFront.sd, 21
 Typewriter type, 13

U

Underlined, 12
 Undo, 21
 Usage, 1
 User, 17
 User keyword, 17

W

Wildcards, 18