# The pdfcprot.sty Package.\*

#### Carsten Schurig<sup>†</sup>Tobias Schlemmer

#### June 10, 2005

#### Abstract

This package was written to provide the "normal"  $\text{LAT}_{\text{E}}X 2_{\varepsilon}$  user an easy way to use the special character protruding feature invented by pdfTEX. Further this package provides an easy interface to adjust the character protrusion for different fonts and choosing the right adjustment automatically depending on the font.

### Contents

1	Introduction 1		1
	1.1	A bit of History	1
	1.2	Provided Features	2
<b>2</b>	Usiı	ng this package.	<b>2</b>
	2.1	Requirements	2
	2.2	Character Protruding	2
		2.2.1 The Simple User Interface	2
		2.2.2 The Advanced User Interface	3
	2.3	The Character Protruding Adjustment – Creating and Using a New One	
		for a Specific Font	6
	2.4	Customising the Package	7
		2.4.1 System-wide customisation	$\overline{7}$
	2.5	Other Commands and Options	7
3	Bug	s and Caveats	8
4	Con	tributing	8
5	Ack	nowledgements	8
$\mathbf{A}$	The	Code Itself	9
	A.1	The Package	9
	A.2	The Provided .CPA Files	24
		A.2.1 Palatino	24

### 1 Introduction

As the abstract stated this package exists to provide a simple user interface to the character protruding feature of  $pdfT_{E}X^{1}$ . This is a special way to do a margin kerning. By that it is possible to achieve a visual more "smooth" margin. When not doing a margin kerning, the margins seems to be flattered a bit, especially at hyphens and

<sup>\*</sup>This file has version v1.7a dated 2005/05/23.

<sup>&</sup>lt;sup>†</sup>This package depends intensely on code originally from Hàn Thế Thàn.

<sup>&</sup>lt;sup>1</sup>You'll find pdfT<sub>E</sub>X on CTAN:/sytems/pdftex.

punctuation. Character Protruding is a simple but nevertheless effective way to achieve a margin kerning. For more information about margin kerning and especially character protruding see the thesis of Hàn Thế Thành  $[1]^2$ .

#### 1.1 A bit of History

Why did I write this package, so you can use it now? Well, it all started with my diploma thesis. I used  $IAT_EX 2_{\mathcal{E}}$  on a regular basis before that, but here I still had to increase my knowledge, esp. in typography. So I read somewhere in a book something about margin kerning (some book of Jan Tschichold). So I wanted to test, how it would look like.

But my first questions in de.comp.text.tex where a bit unsuccessful. The only answers where, that it could be achieved with T<sub>E</sub>X, but that it is not without problem. As an example the  $\[MT_EX-Companion was mentioned, at least the German edition[2] was$ put with a kind of hanging punctuation. But in the preface Frank Mittelbach stated, $that there doesn't exist a ready to use package for <math>\[MT_EX 2_{\varepsilon}\]$  and one would have to wait for  $\[MT_EX 3\]$  very likely. And as you can easily see, only the hyphens and the punctuation signs were protruded. That was the situation of November 1994.

But the situation seemed to be the same in 2000. I found *nothing* really usable with  $IeT_EX 2_{\varepsilon}$  to put a longer text with (longer than some paragraphs). Then someone pointed out, that character protruding is a feature provided by the new pdfTEX. At first I was a bit sceptical because I didn't want to switch, needing the possibility to use EPS input (I heavily used psfrag). But soon it was pointed out, that pdfTEX supports DVI output and that in this mode pdfTEX has no limitations compared to IeTEX. Shortly after getting this information I found protcode.tex provided by Han Thế Thành.

That I transferred to a packaged for my own use and as some people showed interests on de.comp.text.tex I posted it there.<sup>3</sup> But it was never what I expected by a package to submit it to CTAN. It lacked documentation and the support of setting the character protruding depending on the used font.

First I wanted to write a package adjusting some of the additional features of pdfT<sub>E</sub>X (for instance the security features). But in discussions with Heiko Oberdiek he convinced me, that it is better to have some smaller packages instead of putting all into one. Additionally the resulting pdfcprot package grew more complex than expected at first. So the code is complex enough for me to maintain and Heiko does a really good job with pdfsec.sty (he supports more than pdfT<sub>E</sub>X).

#### **1.2** Provided Features

This package provides a very simple interface to support a font dependent selection of the adjustment of the character protrusion. You can easily choose, which fonts will be protruded depending on high level  $L^{AT}EX 2_{\varepsilon}$  font selection not knowing which font actually will be used. Further on it is very simple to adjust the character protrusion for fonts not "supported" directly (only for Palatino character protrusion adjustment set (CPA) is provided and that will be used for *all* fonts, not having a special CPA).

Further on an option to switch the  $pdfT_EX$  output to DVI and a command testing, whether you use PDF output or not are provided.<sup>4</sup>

### 2 Using this package.

First you have to invoke it with a \usepackage{pdfcprot} in the preamble of your document. Of course you have to use pdfT<sub>E</sub>X to get any effect, but the package just gives a warning if you don't.

<sup>&</sup>lt;sup>2</sup>It could be found on the web, but I don't remember where I got it.

<sup>&</sup>lt;sup>3</sup>The first version was named optrandausgleich.sty and a second pdftexfeats.sty.

<sup>&</sup>lt;sup>4</sup>That was originally from KOMA-Script.

family	series	shape
\rmfamily	\mdseries	\upshape
		\itshape
	\bfseries	\upshape
\sffamily	\mdseries	\upshape
		\itshape
		\slshape
\sffamily	\bfseries	\upshape

sizes for all fonts: \footnotesize, \small, \normalsize and \large

Table 1: Fonts for which character protrusion will be enabled using the package option avtivate=normal.

#### 2.1 Requirements

This package need just two packages, which should be part of every  $\mathbb{IAT}_{E}X 2_{\varepsilon}$  distribution not too old: keyval.sty (part of the graphicx bundle) and ifthen.sty). If they're not installed, get them from CTAN.

#### 2.2 Character Protruding

For using the character protruding two user interfaces are implemented. If you just want to activate it for the most common used fonts, not thinking much about internals just use the *simple* interface.

When you don't like the preselection of the fonts being adjusted for character protruding, you may want to use the *advanced* user interface. As the usage of many fonts for character protruding is very memory consuming, there may be some cases you want to choose non-ambiguous the fonts getting character protruding sparing any useless font.

#### 2.2.1 The Simple User Interface

#### activate [=none, normal]

If you just want to activate character protruding for the most used fonts just say activate or activate=normal (these two forms are synonyms). For which fonts the character protrusion will be enabled is shown in table 1.

By default character protruding won't be enabled, but if you want the options to reflect it say activate=none.

Some comments: You may wonder why character protruding is activated for so many font types and not just for  $\mfamily\mdseries\upshape$  and maybe  $\ffamily\mdseries\upshape$  using activate or activate=normal as package option. That is done because I don't want to bother the "normal"  $\mbox{IAT}_EX 2_{\mathcal{E}}$ -user with more than passing the one option to the package but I still want to catch the fonts, most likely to be used in cases "needing" character protrusion (this means justified text). As this may appear not only for the text type (but e.g. also for captions), though the two fonts mentioned above are not sufficient.

#### 2.2.2 The Advanced User Interface

In some cases it may be favourable to have a bit more control over the font selection for the character protruding. With the commands described in this section a very fine selection is possible.

The first command to be mentioned has a very simple syntax. It allows the user to select a specific font encoding to be set up. It will perform the same setup for the given fontencodings as the activate=normal package option does without selection of font encoding. As for now pdfcprot has no opportunity to automagically detect the loaded font encodings, it will be necessary for setting up character protuding in cases where more than one font encodings are used.

\setupcharprotrudingforencoding{encodings}

*encodings* This parameter is a comma sepapated list of font encodings to be set up. White space is not allowed.

Example 1: The usage of

\usepackage[activate=none]{pdfcprot}
\setupcharprotrudingforencoding{T1,T2A}
\activatecharprotruding

will set up character protruding for both T1 and T2A font encodings. This might be nessessary for example, if you want to use both German and Russian text in one document.

The second command to be mentioned has a high level syntax. For that it's possible to choose the font by high level  $\text{LAT}_{\text{E}} X 2\varepsilon$  font selection commands but only *one* font at a time is selectable. To setup the character protruding for more than one font, this command has to be called more than once.

\setupcharprotruding{encoding=enc,family=fm, series=sr, shape=sh, size=sz[, textcomp=tc]}				
encoding	This tag specifies the encodings to be used. Valid values are all valid font encodings. If more than one font encoding shall be set up, they have to be seperated by commas enclosed in braces. This tag is optional. If it is not used, the command will behave like in versions prior to v1.7			
family	This tag specifies the family to be used. <b>rmfamily</b> and <b>sffamily</b> are valid values. Any other value will be supposed as a low level font name (see below). <i>This tag is mandatory.</i>			
series	This tag is used to chose the font series. Valid values are <b>mdseries</b> and <b>bfseries</b> . Any other value will be supposed as a low level font name (see below). <i>This tag is mandatory</i> .			
shape	The font shape is chosen using this tag. As for the other tags valid values are the high level $IAT_EX 2_{\varepsilon}$ font selection commands without trailing backslash, as are: upshape, itshape, slshape, and scshape. Any other value will be supposed as a low level font name (see below). This tag is mandatory.			
size	This tag is to select the wanted font sizes. Valid values are also the LATEX $2\varepsilon$ high level font size selection commands without trailing backslash, that are: Huge, huge, LARGE, Large, large, normalsize, small, footnotesize, scriptsize, and tiny. This tag is mandatory.			
textcomp	This is a boolean switch for choosing whether to activate character protrud- ing for TS1 encoded fonts also or not. Values are true (on, yes, 1), or false (off, no, 0). This tag is optional. If this tag is not specified it defaults to false.			
The low the high le	<b>level font selection.</b> The "normal" parameters for font selection are evel $\Pr_{E} X 2_{\varepsilon}$ font selection commands without trailing backslash. Using any			

the high level  $IAT_EX 2_{\varepsilon}$  font selection. The hormal parameters for four selection are the high level  $IAT_EX 2_{\varepsilon}$  font selection commands without trailing backslash. Using any option other than that for the tags family, series, and shape, a low level font selection command will be suggested. By that it is possible to setup the character protruding for a font with a user defined font selection command but without struggling with internal commands of pdfcprot.sty.

Example 1: The usage of

value	family	series	shape	size
0	none	none	none	none
1	sffamily	bfseries	scshape	tiny
2	rmfamily	mdseries	slshape	scriptsize
4	nil	nil	itshape	footnotesize
8	nil	nil	upshape	small
16	nil	nil	nil	normalsize
32	nil	nil	nil	large
64	nil	nil	nil	Large
128	nil	nil	nil	LARGE
256	nil	nil	nil	huge
512	nil	nil	nil	Huge

Table 2: The numbers to use with \setupcharprotrudingnumeral.

 $\cite{times}$ 

will result in a command to setup character protruding like:
\fontencoding{T2A}\selectfont\rmfamily\fontseries{elec}\selectfont
\fontshape{ui}\selectfont\normalsize\CPROT@setprotcodes@font

This means for the font \rmfamily with font series elec and font shape ui character protruding for T2A encoding will be adjusted.

#### Example 2: Using

 $\ \$ 

will lead to a command to setup character protruding like:
{\CPROT@setprotcodes@font}
{\rmfamily\bfseries\upshape\normalsize\CPROT@setprotcodes@font}

A low level selection of the font size is not possible with \setupcharprotruding. For setting up the character protruding using other font size commands than the high level  $IAT_EX 2_{\varepsilon}$  ones, the usage of internal pdfcprot commands is needed.

To setup more than two or three fonts with the command above described would be a bit long-winded. So there's another command provided with which it is possible to select more than one font at a time (with some limitations).

#### \setupcharprotrudingnumeral{encoding=enc,family=fm, series=sr, shape=sh, size=sz[, textcomp=tc]}

The principal meaning of the tags is the same as for \setupcharprotruding, especially encoding and textcomp is exactly the same. The only thing differing for family, series, shape, and size is the way the font has to be chosen.

Here a scheme is used similar to the numeral interface of chmod on UNIX systems. That means every  $L^{AT}EX 2_{\varepsilon}$  high level font selection command got a number assigned. More than one fonts are choose able by adding the fitting numbers. The suitable values are shown in table 2. If one parameter is zero, the actual command will have no effect,

Example 1: For \rmfamily\mdseries the character protruding is to be adjusted for \upshape and \itshape for the sizes \large, \normalsize and \footnotesize. Further for the font \rmfamily\bfseries\upshape with the same font sizes, but in T2A encoding, character protruding is wanted. To achieve that, one would have to call:

```
\usepackage[activate=none]{pdfcprot}
\setupcharprotrudingnumeral{family=2,series=2,shape=12,size=52}
\setupcharprotrudingnumeral{encoding=T2A,family=2,series=1,shape=8,size=52}
\activatecharprotruding
```

Explanation: family and series are directly readable from table 2, as is shape for the second call of \setupcharprotrudingnumeral. To get the value of shape for the first command, the values for shape=upshape and shape=itshape (8 and 4) must be added. Similar it's for size, one has to add the values for size=footnotesize (4), size=normalsize (16), and size=large (32).

The setup of character protruding using \setupcharprotrudingnumeral is just possible for fonts accessed by high level  $IAT_EX 2_{\varepsilon}$  font selection commands.

**Remarks.** Both commands to select the fonts for which character protrusion will be adjusted are *only* usable in the preamble of the document.

The attentive reader may have noticed another command needed to activate the character protrusion when calling the package with the option activate=none. That is due to the fact, that the \setupcharprotruding commands just create a command to call at the start of the document to adjust the amount of the character protruding for a special font (on a character basis), but does not activate the using of character protruding itself.

#### \activatecharprotruding[activate]

Valid values for *activate* are true (yes, on, 1), false (no, off, 0), or compatible (compatibility). Calling \activatecharprotruding without any parameter means true. With true and false the character protruding is switched on and off respectively. Why the parameter compatible? As the character protrusion moves some character into the margins, the word spaces on a line may change. When using activate=true, pdflatex will take this additional space into account and by that the line breaks may change compared to the use of "normal" latex. If the line breaks are wanted as got by using latex but with a active character protruding, you'll have to call \activatecharprotruding with the parameter compatible.

The activation of the character protruding is group specific. So it is possible to activate and deactivate the character protrusion for some parts of a text (but not the *amount* of the character protrusion of a special font).

The package output is very informative when looking at the log file, but the output to stdout is normally not. To change that, there's one option

#### quiet=qt

Possible values for this parameter are the booleans used in this package (true, on, yes, 1, false, off, no, and 0). When saying quiet=no many of the package info will be warnings instead. That is a good method to see, which font character protrusion adjustment actually will be used (and for which font).

# 2.3 The Character Protruding Adjustment – Creating and Using a New One for a Specific Font

This section is a bit technical and a study of the source code may be very helpful (and is recommended).

To understand what the purpose of this section is, a description of the strategy of the package is needed. So what does the package do? The command \setupcharprotruding and its numerical equivalent will create a command with the fitting font selection commands and a command \CPROT@setprotcodes@font. The whole command will be called at the start of the document. Then the font selection takes place and

\CPROT@setprotcodes@font looks, what font was requested actually.<sup>5</sup> Then it looks if a command named \f@family \f@series\f@shape\f@encoding is defined (the concatenate values of the commands). If it is, this command will be used to setup the character protruding, if not pdfcrpot will try to load a file named fontname.cpa containing this command. If that fails as well, a fallback font will be used (by default it will be pplmn with the appropriate font encoding).

To define the command to set up the character protruding, an external file was chosen to get an easier maintenance. Defining it in the package would enlarge the package noticeably and the package would soon become overcrowded.

So if a special character protruding setting for a font is wanted, copy the file **pplmnT1.cpa** (for T1 encoding) to the required fontname.**cpa** and change the definition of the command accordingly.

Example: A special character protruding setup for the bold version of Palatino in T1 encoding is wanted. As the LATEX 2<sub>c</sub> command \bfseries selects an bold expanded version (bx), the font name would be pplbxnT1.cpa and the command to define \pplbxnT1. That means to copy pplmnT1.cpa to pplbxnT1.cpa and change the definition of \csname pplmnT1\endcsname to \csname pplbxnT1\endcsname and further on to adjust the wanted \rpcode and \lpcode values.

For the meaning of the commands and values in the CPA see the documentation of the code below and Hàn Thế Thành's thesis [1]. There he writes:

"... A set of common character protruding factors gave quite reasonable results in most cases. Non-typical type faces may require further adjustments, which can be done easily."

So for the most fonts the default values may be acceptable, even though they are adjusted for Palatino. By using the CPA files and defining an extra command per font it is very easy to add setups for "non-typical" type faces without bothering a "normal"  $\text{LATEX} 2\varepsilon$  user of choosing the right setup because it gets automatically selected.

### 2.4 Customising the Package

There's one command provided to customise the package.

#### $\setfallbackfont{string}$

As described in the section before, this package will look for a character protrusion adjustment for a *fallback* font, when not finding a CPA for the actual font. By default this is pplmn. That is changeable using \setfallbackfont, if a special CPA was created and suits better. The *string* used in \setfallbackfont specifies the font *without* the character encoding. That will be determined by the package.

#### 2.4.1 System-wide customisation

For further customisation this package looks for a

pdfcprot.cfg

somewhere in your  $T_{EX}$  paths. If found it'll be included before any option is validated. So you can change some settings on a system-wide basis. This file is the best place to change some internal commands, if needed, for instance to change which fonts will be activated for character protruding when using the option activate=normal (by redefining \CPROT@setupcharprotruding@normal).

<sup>&</sup>lt;sup>5</sup>The commands used to determine which font is loaded ( $\{f@family, f@series, f@shape, f@encoding$ ) return the font requested to  $T_{E}X$  not the font actually used, so it seems. That means for instance if you request a bold Palatino font with bfseries,  $T_{E}X$  will try to load ppl/bx/n but will only find and use ppl/b/n (at least on my system). So as series the actual font will be b and not bx which will be reported by the use of f@series. Though one could think it would be needed to adjust the character protruding in a CPA for pplbn, it will be looked for pplbxn instead.

#### 2.5 Other Commands and Options

Belonging not really to character protrusion this package provides two other commands and one additional option, that may be useful in dealing with  $pdfT_EX$ .

First there's is an option to get a DVI output using  $pdfT_{EX}$ .

#### DVIoutput

This may be useful if one can't switch completely to  $pdfT_EX$ , because in this mode all the additional features of  $pdfT_EX^6$  (as character protruding) are available but besides there're no compatibility problems and it behaves the same as "normal" LATEX. So it is for instance possible to include EPS (and use psfrag!). Principally it is not needed to supply such an option, because by redefining \pdfoutput to 0 you'll get DVI. The only problem is, that some packages only look if \pdfoutput is defined to decide, if PDF output is wanted. So this option does an additional \let\pdfoutput\undefined.<sup>7</sup>

Further on there are two  $L^{T}EX 2_{\varepsilon}$ -commands to decide, if pdfTEX is used and if PDF as output is wanted.

\ifpdftex{*true*}{*false*}

and

#### \ifpdfoutput{true}{false}

The latter one was originally taken from the KOMA-Script package. So you can use the same \ifpdfoutput command with or without KOMA-Script. The meaning shouldn't change neither in KOMA-Script nor in this package and the actual definition is not cruel for using this command<sup>8</sup>.

The command ifpdftex is very similar but leads to a decision depending on the usage of pdfT<sub>E</sub>X or not. This is useful for the activation of special features of pdfT<sub>E</sub>X not being special to the production of PDF itself (as character protruding is).

### **3** Bugs and Caveats

Hopefully there're no bugs left, but only features ;->. But one thing: right now this package contains just character protruding settings for OT1 and T1 encoded fonts (and one character of TS1). Further there are some (small) problems (two).

First, the creating of the code to do the adjustment of the character protruding at the **\begin{document}** is a bit time consuming. The more fonts you want to use with character protruding, the longer is the time needed. Also the needed memory by pdfTEX can not be neglected. Here it's also especially a problem when using many fonts (in companion with character expansion it's just increasing). But that's not a big problem, because the standard amount of memory for pdfTEX is for computer systems of today just a bit small adjusted (65535 bytes!). Though you may want to change (or set) the parameter pdf\_mem\_size in your texmf.cnf.<sup>9</sup>

### 4 Contributing

Almost any contribution is welcome. Really needed is the contribution of CPA's for other encodings than T1 and OT1. But T1 and OT1 CPA's may not be perfect. The glyphs are complete for German (I hope so) but I don't know for other languages.

Also CPA for fonts not looking good with the distributed "default" values would be nice, even if these fonts are not widely used.

<sup>&</sup>lt;sup>6</sup>That's not completely true though. The supported type of images is in DVI mode the same as using latex. <sup>7</sup>Most actual versions of packages having had problems may treat it correctly by now. But who has all packages installed in most recent versions?

<sup>&</sup>lt;sup>8</sup>If you want a TEX \if switch, look for ifpdf.sty on CTAN written by Heiko Oberdiek.

<sup>&</sup>lt;sup>9</sup>That is a suggestion of myself. I don't know what the authors of pdfT<sub>E</sub>X say to that!.

Any bug hunting is welcome. Also changes to improve the performance are likely to be included, if the performance increase is significant and the readability of the package is not distorted too much (a good readability was a main focus when writing this package; that's one reason why almost on all places where possible  $\text{LAT}_EX 2_{\varepsilon}$  commands are used.).

If you have some changes, requests, ideas or any other things regarding this package feel free to mail to Tobias Schlemmer: keinstein\_junior@gmx.net.

### 5 Acknowledgements

All of the people are already mentioned in the text above, but I have to emphasise some things.

For these bundle I used many from KOMA-Script. Particularly the boring but needed things around the package (README.txt, INSTALL.txt and the preambles in pdcprot.ins) were designed using files from KOMA-Script. In some parts I just changed KOMA-Script to pdfcprot (were appropriate). So I owe much to Markus Kohm (or the users of this package, because these files would be much shorter and maybe indistinct). As already stated there's some code taken from KOMA-Script.

Further on some important role played Heiko Oberdiek (I don't think he guesses so). As stated in the History I first wanted to create a package for many (all) special pdfTEX features (character protruding, font expansion, PDF security options). But he convinced me to write a package just concerning character protruding – luckily, because it's got to more code than I expected first.

And last but not least the authors of  $pdfT_EX$  have to be mentioned. Some code comes directly from them.

### References

- Hàn Thế Thành. Micro-typographic extensions to the T<sub>E</sub>X typesetting system. Dissertation, Masaryk University Brno: Faculty of Informatics, October 2000.
- [2] Michel Goosens, Frank Mittelbach, and Alexander Samarin. Der LATEX-Begleiter. Addison-Wesley, 1st edition, 1994.

### A The Code Itself

First some remarks: the documentation may be inaccurate in some places, so look at the code and it'll be very likely that the documentation is incomplete.

The main attention writing this code was turned on creating a good human readable code. So I decided to use as much  $\text{LAT}_{\text{EX}} 2_{\varepsilon}$  control sequences as possible and as less TEX commands as needed. This may slow down the code, but I don't think that's really important.

### A.1 The Package

```
1 \langle * \mathsf{package} \rangle
```

First the requirement of  $\mathbb{A}T_{\mathbf{E}} X 2\varepsilon^{10}$  and the declaration of the package.

- 2 \NeedsTeXFormat{LaTeX2e}[1994/12/01]
- 3 \ProvidesPackage{pdfcprot}[2002/02/27 v1.5 character protruding using 4 pdflatex (cs)]

Then the required packages are loaded. Only ifthen.sty and keyval.sty are needed. These should be part of any decent  $\operatorname{IATFX} 2_{\mathcal{E}}$ -distribution ;->.

- \RequirePackage{ifthen}
- 5 \RequirePackage{ifthen} 6 \RequirePackage{keyval}

<sup>&</sup>lt;sup>10</sup>The version needed was taken from ifthen.sty, because this package does request it. But I don't know if pdfcprot.sty may require some newer version because of some construct being used. If someone finds something which requires a newer version of  $IATEX 2\varepsilon$  please tell me.

These are the counters to determine for which font character protruding will be activate by using \setupcharprotruding and \setupcharprotrudingnumeral. The possible values are listed in table 2. Further on CPROT@family, CPROT@series, and CPROT@shape can be -1, which is used for the low level font selection by \setupcharprotruding. The ... @temp counters are needed when creating \CPROT@setprotcodes@.

7\newcounter{CPROT@family} \newcounter{CPROT@family@temp} 8 \newcounter{CPROT@series} 9 10\newcounter{CPROT@series@temp} \newcounter{CPROT@shape} 11 \newcounter{CPROT@shape@temp} 12 \newcounter{CPROT@size} 1314 \newcounter{CPROT@size@temp} For determining if the character protruding will be activated for the corresponding TS1 encoded font a boolean is used. \newboolean{CPROT@textcomp} 15 For providing the quiet option a (global) boolean is used. By default it is set to true, quiet so there're little warnings (but it will be written as info to the logs). \newboolean{CPROT@quiet} 16\setboolean{CPROT@quiet}{true} 17

A temporary counter (\@tempcnta could be used instead).

\newcounter{CPROT@temp@chars}

\CPROT@lowlevel@encoding Some "vars" are needed to store the low level names if some low level font was requested \CPROT@lowlevel@family using \setupcharprotruding. (These are just temporary commands only used while creating the command to do the actual adjustment of the character protrusion.) \CPROT@lowlevel@series

\CPROT@lowlevel@shape	\newcommand*{\CPROT@lowlevel@encoding}{}
20	$\  \  \  \  \  \  \  \  \  \  \  \  \  $
21	\newcommand*{\CPROT@lowlevel@series}{}
22	\newcommand*{\CPROT@lowlevel@shape}{}
\CPROT@resetall	This is to reset all the counters to zero, so that a new <b>\setupcharprotruding</b> command
	won't activate something requested with a former command.
23	\newcommand*{\CPROT@resetall}{%
24	\setcounter{CPROT@family}{0}%
25	\setcounter{CPROT@series}{0}%
26	\setcounter{CPROT@shape}{0}%
27	\setcounter{CPROT@size}{0}%
28	\setboolean{CPROT@textcomp}{false}%
29	\edef\CPROT@lowlevel@encoding{}%
30	}
\CPROT@setprotcodes@	These are internal macros regarding the creation of the actual command, to adjust the
\CPROT@setprotcodes@add	character protruding for the fonts wanted. \CPROT@setprotcodes@ actually will contain
\CPROT@setprotcodes@temp	the command, whereas \CPROT@setprotcodes@temp is used to store part of this com- mand temporary. \CPROT@setprotcodes@add is called by \setupcharprotruding and

- \setupcharprotrudingnumeral. \newcommand\*{\CPROT@setprotcodes@}{} 31 32 \newcommand\*{\CPROT@setprotcodes@add}{% The counter CPROT@family@temp is set, which will be used internally to determine which fonts were requested. \setcounter{CPROT@family@temp}{\value{CPROT@family}}% 33 Here the actual evaluation will take place starting with the font encoding.
  - \CPROT@aac@encoding% }
  - 35

34

18

\newcommand\*{\CPROT@setprotcodes@temp}{} 36

\CPR0T@packageinfo	These are just substitutes for \PackageInfo, \PackageWarning, and \PackageError. It's
	just to spare some writing (esp. $\{$ snf $\}$ are "difficult" to reach on a German keyboard)
\CPPOT@packagewarning	and it's simplify to customic if I would want to do something other than normal
(CFR01@packagee1101	and it's simpler to customise if I would want to do something other than normal.
37	\newcommand{\CPROT@packageinfo}[1]{%
38	\PackageInfo{pdfcprot}{#1}%
39	}
40	\newcommand{\CPROT@packagewarning}[1]{%
41	\PackageWarning{pdfcprot}{#1}%
42	}
43	\newcommand{\CPROT@packageerror}[1]{%
44	\PackageError{pdfcprot}{#1}%
45	}
CPROT@packageinfo@or@warning	To support the Option quiet this command is used. It decides on the boolean
	CPROT@quiet wether to give out a message as warning or as info.
46	\newcommand{\CPROT@packageinfo@or@warning}[1]{%
47	\ifthenelse{\boolean{CPROT@quiet}}{%
48	\CPROT@packageinfo{#1}%
49	<b>}</b> -{%
50	\CPROT@packagewarning{#1}%
51	}%
52	}%
encoding	
53	\define@key{CFKU1}{encoding}{%
54	\itcheneiset\equal#ijtencodingderalitjf{%
55	<pre>\eaci \CPkUI@iowlevei@encoding{\encodingdefault}% \encodingdefault will be set.</pre>
56	H% I assume a low level encodingname is given
57	\renewcommand*{\CPRU1@lowlevel@encoding}{#1}%
58	5% 5%
59	}
family	
60	\define@kev{CPRNT}ffamilu}{%
61	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
62	\setcounter{CPR0T0family}{0} \set
63	
64	\ifthenelse{\equalf#1}fsffamilv}f%
65	\setcounter{CPRIT@family}}/sffamily will be set
66	f(x) = f(x) +
67	\set counter{CPRITAtempAchars}{0}
68	\evenandafter\@tfor\evenandafter\@temnh\evenandafter.\evenandafter=#1\dof%
69	(expanded for (corporate to (corporate for ))
70	Y
70	Jn \ifthanelse{\ualue{CPRNT@temn@chars}>4}{%
72	(It then the Counter (on the Competent None of the keywords 'all' \MassageBraak
72	(on the space again to go the second of the Adjustic and a second barries and a second barries and the second barr
74	limitanity of situatity was found. I assume you gave a hissagebiant
74	Tow-forthown has more the A characters. I assume that a knowledge break
15	tex-tonchame has more than 4 chalacters. I assume you know wessagebreak
70	what you're doing and continue, but the selection of the messagebreak
11	Iont may fall!%
78	<i>F</i> / <sub>6</sub>
79	\CPRUT@packageinio{lt seem's as you want to use a low level\MessageBreak
80	command for font selection You're for yourself.%
81	5%
82	J{J%
83	\setcounter{CPROT@family}{-1}%
84	\renewcommand*{\CPROT@lowlevel@family}{#1}%
85	}%
86	}%
87	}
series	
88	\definewkey{CFKUT}{series}{%

89	\ifthenelse{\equal{#1}{mdseries}}{% medium series
90	\setcounter{CPR0T@series}{2}%
91	}{%
92	\ifthenelse{\equal{#1}{bfseries}}{% bold series
93	\setcounter{CPR0T@series}{1}%
94	$\{\%$ here the low level interface is suspected
95	\setcounter{CPROT@temp@chars}{0}%
96	\expandafter\@tfor\expandafter\@tempb\expandafter:\expandafter=#1\dof%
97	\stepcounter{CPR0T@temp@chars}%
98	ly
99	\ifthenelsef\valuefCPR0T@temn@chars}>4}{%
100	(CPR0T0nackageinfo@or@warning(None of the keywords 'all' \MessageRreak
100	indepring our observation and in a server way a Massaga Praak
101	Industries of Distries was found. I assume you gave a disesseptiean
102	of $A$ observations and basic basic basic sections and the point observations and the point of
103	doing and continue but the selection of the fort will MessageBreak
104	using and continue, but the selection of the font will desagebreak
105	Very likely fall:%
100	
107	(CrRUTepackageIntofit seem's as you seem want to use a low level (Messagebreak
108	command for font selection You're for yourself./
109	5% 2011
110	
111	\setcounter{CPRUT@series}{-1}%
112	\renewcommand*{\CPROT@lowlevel@series}{#1}%
113	}%
114	}%
115	}%
shape	
116	\define@key{CPROT}{shape}{%
117	\ifthenelse{\equal{#1}{upshape}}{%
118	\setcounter{CPROT@shape}{8}%
119	}{%
120	\ifthenelse{\equal{#1}{itshape}}{%
121	\setcounter{CPROT@shape}{4}%
122	}{%
123	\ifthenelse{\equal{#1}{slshape}}{%
124	\setcounter{CPR0T@shape}{2}%
125	}-{%
126	\ifthenelse{\equal{#1}{scshape}}{%
127	\setcounter{CPROT@shape}{1}%
128	}{% low lewel format
129	\setcounter{CPR0T@temp@chars}{0}%
130	\expandafter\@tfor\expandafter\@tempb\expandafter:\expandafter=#1\dof%
131	\stercounter{CPR0T0temp@chars}%
132	
132	\ifthenalcef\ualuefCPROT@temp@charc}>?}{%
134	(PROTOnackage info Morowarning (None of the keywords 'all') \MessageBreak
135	() horspackage intersected with ing (none of a new relation of a construction of a c
136	found I accume you gave a low level four name but it/MaccaneBreak
130	is at least unusual that a share identifier has more MagageBreak
137	then 2 characteria I acquine you know that you're deine Macagenerak
130	and continue but the selection of the font will year. MessageBreak
139	and continue, but the selection of the four will very (nessagebleak
140	
141	
142	(orkul@packageinioit seem's as you seem want to use a low level(messagebreak
143	command for font selection fourre for yoursell.
144	
145	$f(f_{A})$
146	\setCounter(UPKUIUsnape){-1}%
147	/renewcommang//crknimiowieveimsuabe}/#1}%
148	5%
149	5%
150	۲۸ ۲۸
151	36
152	3

size	
153	\define@key{CPROT}{size}{%
154	\ifthenelse{\equal{#1}{Huge}}{%
155	\setcounter{CPROT@size}{512}%
156	<b>}{%</b>
157	\ifthenelse{\equal{#1}{huge}}{%
158	\setcounter{CPR0T@size}{256}%
159	<b>}</b> -{%
160	\ifthenelse{\equal{#1}{LARGE}}{%
161	\setcounter{CPROT@size}{128}%
162	}{%
163	\ifthenelse{\equal{#1}{Large}}{%
164	\setcounter{CPROT@size}{64}%
165	}-{%
166	\ifthenelse{\equal{#1}{large}}{%
167	\setcounter{CPROT@size}{32}%
168	}{%
169	\ifthenelse{\equal{#1}{normalsize}}{%
170	\setcounter{CPROT@size}{16}%
171	
172	\ifthenelse{\equal{#1}{small}}{%
173	\setcounter{CPRUT@size}{8}%
174	
175	\lthemelset\equal(#1)flootnoteslze}}{%
176	\setCounter(CPKUI@siZe;{4;%
177	
178	<pre>(litheneise()equal(#1;tscriptsize;f;t, )actorymtar/CPD07dairal())</pre>
179	\SetCounter\UFKUT@SIZef\25% \5%
100	Jlo \ifthonoloof\oguolf#1\f+inullf%
182	\interests(requark#if(inty)f(%
182	Second longestable
184	CPROTOnackageerror{%
185	None of the keywords 'all' 'normal' or any IaTeX2e\MessageBreak
186	font name was found. MessageBreak
187	\space For more information how to use\MessageBreak
188	\string\setupcharprotrudingnumeral see the
189	pdfcprot manual.%
190	3%
191	}{}%
192	}%
193	3%
194	}%
195	}%
196	}%
197	3%
198	}%
199	5%
200	5%
201	}
202	\define@key{CPROT}{textcomp}{%
203	\ifthenelse{\equal{#1}{true}\or\equal{#1}{or\equal{#1}{1}\or\equal{#1}{yes}}{%
204	\setboolean{CPROT@textcomp}{true}%
205	}{%
206	\ifthenelse{\equal{#1}{false}\or\equal{#1}{off}\or\equal{#1}{0}\or\equal{#1}{no}}{%
207	\setboolean{CPROT@textcomp}{false}%
208	
209	\UPKUI@packageerror{%
210	"#1'' is no setting for ``textcomp''. Use\MessageBreak
211	one of 'true', 'on', 'yes', '1', or 'false', 'off', \MessageBreak
212	'no', 'U' 1nstead.\MessageBreak
∠13 214	5/6 29
214 915	5 /v 2 /
210 216	۲ ۱۳
217	\define@kev{CPROTnum}{encoding}{%

```
218
           ifthenelse{=}{#1}{0}{%}
219
             \renewcommand*{\CPROT@lowlevel@encoding}{}% No encoding will be set.
220
           Դ√%
221
             \ifthenelse{\equal{#1}{1}}{%
222
                \edef\CPROT@lowlevel@encoding{\encodingdefault}% \encodingdefault will be set.
             }{%
223
224
                \ifthenelse{\equal{#1}{encodingdefault}}{%
                  \edef\CPR0T@lowlevel@encoding{\encodingdefault}% \encodingdefault will be set.
225
226
               }{% I assume a low level encodingname is given
227
                  \renewcommand*{\CPROT@lowlevel@encoding}{#1}%
               7%
228
             }%
229
230
           }%
         }
231
232
         \define@key{CPROTnum}{family}{%
           ifthenelse{#1>0 \  \  1<4}{%}
233
             \setcounter{CPROT@family}{#1}%
234
235
             7.4%
             ifthenelse{#1=0}{%}
236
237
               \CPROT@packageinfo{%
238
                 As you selected '0' (that means 'none') for\MessageBreak
239
                 selection of the font family, no font will
\MessageBreak
240
                 be set up for char protruding.%
241
                 }%
               }{%
242
243
                \CPROT@packageerror{%
                Your given value '#1' to family in \string\setupcharprotrudingnumeral\MessageBreak
244
245
                 is not valid. Sensible values are between 1 and 3 (inclusive).\MessageBreak
246
                 For further information see the pdfcprot manual.%
247
                 7%
248
               }%
249
             }%
           }
250
         \define@key{CPROTnum}{series}{%
251
           ifthenelse{#1>0 \  \  1<4}{%}
252
253
             \setcounter{CPROT@series}{#1}%
254
             }{%
             ifthenelse{#1=0}{%}
255
256
               \CPROT@packageinfo{%
257
                 As you selected '0' (that means 'none') for\MessageBreak
258
                 selection of the font series, no font will\MessageBreak
                 be set up for char protruding.%
259
260
                 7%
261
               }{%
262
                \CPROT@packageerror{%
                Your given value '#1' to series in \string\setupcharprotrudingnumeral\MessageBreak
263
264
                 is not valid. Sensible values are between 1 and 15 (inclusive).\MessageBreak
265
                 For further information see the pdfcprot manual.%
266
                 }%
               }%
267
268
             }%
           7
269
270
         \define@key{CPROTnum}{shape}{%
           ifthenelse{#1>0 \and #1<16}{%}
271
272
             \setcounter{CPROT@shape}{#1}%
273
             }{%
             ifthenelse{#1=0}{%}
274
275
               \CPROT@packageinfo{%
                 As you selected '0' (that means 'none') for\MessageBreak
276
277
                 selection of the font shape, no font will\MessageBreak
278
                 be set up for char protruding.%
279
                 7%
280
               }{%
281
                \CPROT@packageerror{%
                Your given value '#1' to shape in \string\setupcharprotrudingnumeral\MessageBreak
282
283
                 is not valid. Sensible values are between 1 and 15 (inclusive).\MessageBreak
```

284	For further information see the pdfcprot manual.%
285	}%
286	3%
287	χ{
288	
289	define@kev{CPR0Tnum}{size}{%
200	\ifthenelsef#150 \and #1<102/15%
201	(inductory) (DDOTAGizel/#11/
291	
292	ፓጊሉ እ. ት ታት
293	$(\text{II thene is } e_{i} = 0) $
294	(CPRUIupackageinio{%
295	As you selected '0' (that means 'none') for MessageBreak
296	selection of the font size, no font will\MessageBreak
297	be set up for char protruding.%
298	}%
299	}{%
300	\CPROT@packageerror{%
301	Your given value '#1' to size in \string\setupcharprotrudingnumeral\MessageBreak
302	is not valid. Sensible values are between 1 and 1023 (inclusive).\MessageBreak
303	For further information see the pdfcprot manual.%
304	7%
305	
306	34
307	
307	ر المراجع معرفة معرفة معرفة معرفة معرفة معرفة المعرفة معرفة
308	(definewey(crothum)(crotextcomp)(crue)(%
309	\setkeystoPk01fttextcomp=#1f%
310	}
activate	
311	\define@key{CPROTpackage}{activate}[normal]{%
312	\ifthenelsef\equal{#1}{normal}}{%
313	\CPR0T@setupcharprotruding@normal%
314	\coupling_components
315	f(t) just activate that not when something was set
216	) (% Just a double of the proton and some onling was been
217	\activatecharprotruung[true]/
317	5/6 10/
318	
319	\itheneise{\equal{#1}{none}}{%
320	54%
321	\CPROT@packageerror{%
322	Value of activate can be 'normal' or 'none'.\MessageBreak%
323	\space For more low level setup use \string\setupcharprotruding\MessageBreak%
324	and see the manual%
325	}%
326	3%
327	3%
328	3%
329	\define@key{CPROTpackage}{quiet}[true]{%
330	\ifthenelsef\equal{#1}{true}\or\equal{#1}fon}\or\equal{#1}fves}\or\equal{#1}f1}f%
331	\setboolean{CPR0T@quiet}{true}%
332	14%
333	\ifthenelse{\equal{#1}fslse}\or\equalf#1}foff\\or\equalf#1}foff\\or\equalf#1}fof}\or\equalf#1}fof}
334	\setbolean{CPRIT@mijet}false}
22K	///
222	) CODUTIONS STREAM ST
000 007	Vertuispackageerioita
337	value of quiet should be 'true' ('on', 'yes, '1') or MessageBreak,
338	'Ialse' ('OII', 'no', 'U'). You didn't seem to use any\MessageBreak%
339	of them.%
340	3.4
341	}%
342	}%
343	}%

These are internal fuctions to evaluate the counters used to indicate which font variation are to get protruding.

344	\newcommand{\CPROT@aac@encoding}{%
345	\ifthenelse{\equal{\CPROT@lowlevel@encoding}{}}{%
346	\setcounter{CPROT@family@temp}{\value{CPROT@family}}%
347	\CPROT@aac@family{}%
348	}{%
349	\@for \CPROT@aac@encoding@first:=\CPROT@lowlevel@encoding \do {%
350	\setcounter{CPROT@family@temp}{\value{CPROT@family}}%
351	\expandafter\ifx\csname T@\CPROT@aac@encoding@first\endcsname\relax%
352	\ifthenelse{\boolean{CPROT@quiet}}{%
353	\CPROT@packagewarning{Encoding \CPROT@aac@encoding@first not defined\MessageBreak
354	(see log file for more information)}
355	} <del>0</del> %
356	\CPROT@packageinfo@or@warning{%
357	You've requested char protruding for \CPROT@aac@encoding@first encoding\MessageBreak
358	but \CPROT@aac@encoding@first encoding doesn't seem to be loaded.\MessageBreak
359	Maybe you forgot a '\string\usepackage[\CPROT@aac@encoding@first]{fontenc}'.%
360	}%
361	\else
362	\CPROT@aac@family{\protect\fontencoding{\CPROT@aac@encoding@first}\selectfont}%
363	\fi
364	}%
365	}%
366	}
367	$\$ \changes{1.5}{2002/02/27}{added a some more protection for each font
368	% selection command (new  adds something to
369	% some of the font size selection commands)}
370	$\$ \changes{1.7}{2004/06/28}{added parameter for using with encodings}
371	% \begin{macrocode}
372	\newcommand{\CPROT@aac@family}[1]{%
373	\ifthenelse{\value{CPROT@family@temp}>1}{%
374	\protected@edef\CPROT@setprotcodes@temp{#1\protect\rmfamily}%
375	\setcounter{CPROT@series@temp}{\value{CPROT@series}}%
376	\CPROT@aac@series{\CPROT@setprotcodes@temp}%
377	\addtocounter{CPROT@family@temp}{-2}%
378	\CPROT@aac@family{#1}%
379	}{%
380	\ifthenelse{\value{CPROT@family@temp}>0}{%
381	\protected@edef\CPROT@setprotcodes@temp{#1\protect\sffamily}%
382	\setcounter{CPROT@series@temp}{\value{CPROT@series}}%
383	\CPROT@aac@series{\CPROT@setprotcodes@temp}%
384	\addtocounter{CPROT@family@temp}{-1}%
385	\CPROT@aac@family{#1}%
386	}{%
387	\ifthenelse{\value{CPROT@family@temp}<0}{%
388	\protected@edef\CPROT@setprotcodes@temp{#1\fontfamily{\CPROT@lowlevel@family}\selectfont}%
389	\setcounter{CPROT@series@temp}{\value{CPROT@series}}%
390	\CPROT@aac@series{\CPROT@setprotcodes@temp}%
391	}{}%
392	3%
393	}%
394	}
395	\newcommand{\CPROT@aac@series}[1]{%
396	\ifthenelse{\value{CPROT@series@temp}>1}{%
397	\protected@edef\CPROT@setprotcodes@temp{#1\protect\mdseries}%
398	\setcounter{CPROT@shape@temp}{\value{CPROT@shape}}%
399	\CPR0T@aac@shape{\CPR0T@setprotcodes@temp}%
400	\addtocounter{CPR0T@series@temp}{-2}%
401	\CPRUT@aac@series{#1}%
402	H%
403	\ifthenelse{\value{CPROT@series@temp}>0}{%
404	<pre>\protected@edef\CPR0T@setprotcodes@temp{#1\protect\bfseries}%</pre>
405	<pre>\setcounter{CPR0T@shape@temp}{\value{CPR0T@shape}}%</pre>
406	\CPRUT@aac@shape{\CPROT@setprotcodes@temp}%
407	\addtocounter{CPR0T@series@temp}{-1}%
408	\CPROT@aac@series{#1}%
409	よし しょうしょう しょう

410	\ifthenelse{\value{CPROT@series@temp}<0}{%
411	\protected@edef\CPR0T@setprotcodes@temp{%
412	#1\fontseries\expandafter{\CPROT@lowlevel@series}\selectfont}%
413	\setcounter{CPROT@shape@temp}{\value{CPROT@shape}}%
414	\CPROT@aac@shape{\CPROT@setprotcodes@temp}%
415	}{}%
416	}%
417	}%
418	}
419	\newcommand{\CPROT@aac@shape}[1]{%
420	\ifthenelse{\value{CPROT@shape@temp}>7}{%
421	\protected@edef\CPROT@setprotcodes@temp{#1\protect\upshape}%
422	\setcounter{CPROT@size@temp}{\value{CPROT@size}}%
423	\CPROT@aac@size{\CPROT@setprotcodes@temp}%
424	\addtocounter{CPROT@shape@temp}{-8}%
425	\CPROT@aac@shape{#1}%
426	}{%
427	\ifthenelse{\value{CPROT@shape@temp}>3}{%
428	\protected@edef\CPROT@setprotcodes@temp{#1\protect\itshape}%
429	\setcounter{CPROT@size@temp}{\value{CPROT@size}}%
430	\CPROT@aac@size{\CPROT@setprotcodes@temp}%
431	\addtocounter{CPROT@shape@temp}{-4}%
432	\CPROT@aac@shape{#1}%
433	}{%
434	\ifthenelse{\value{CPROT@shape@temp}>1}{%
435	\protected@edef\CPROT@setprotcodes@temp{#1\protect\slshape}%
436	\setcounter{CPROT@size@temp}{\value{CPROT@size}}%
437	\CPROT@aac@size{\CPROT@setprotcodes@temp}%
438	\addtocounter{CPROT@shape@temp}{-2}%
439	\CPROT@aac@shape{#1}%
440	}{%
441	\ifthenelse{\value{CPROT@shape@temp}>0}{%
442	\protected@edef\CPROT@setprotcodes@temp{#1\protect\scshape}%
443	\setcounter{CPROT@size@temp}{\value{CPROT@size}}%
444	\CPROT@aac@size{\CPROT@setprotcodes@temp}%
445	\addtocounter{CPROT@shape@temp}{-1}%
446	\CPROT@aac@shape{#1}%
447	}{%
448	\ifthenelse{\value{CPROT@shape@temp}<0}{%
449	\protected@edef\CPROT@setprotcodes@temp{%
450	#1\fontshape\expandafter{\CPROT@lowlevel@shape}\selectfont}%
451	\setcounter{CPROT@size@temp}{\value{CPROT@size}}%
452	\CPROT@aac@size{\CPROT@setprotcodes@temp}%
453	}{}%
454	}%
455	}%
456	}%
457	۶% ۲
458	}
459	\newcommand{\CPROT@aac@size}[1]{%
460	\ithenelse{\value{CPRUT@size@temp}>511}{%
461	\protected@edef\CPROT@setprotcodes@temp{#1\protect\Huge}%
462	\CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
463	\addtocounter{CPR0T@size@temp}{-512}%
464	\CPROT@aac@size{#1}%
465	
466	\ithenelse{\value{CPRUT@size@temp}>255}{%
467	\protected@edet\CPKUI@setprotcodes@temp{#1\protect\huge}%
468	\CPKUT@aac@textcomp{\CPKUT@setprotcodes@temp}%
469	\addtocounter{CPRUT@size@temp}{-256}%
470	\CFKUT@aac@size{#1}%
471	
472	\irtheneise(\value(CFKUI@size@temp}>12/}{%
473	\protectequedei\Urkuiusetprotcodes@temp{#1\protect\LAKGE}%
474	\CFKUIWaacwtextcompt\CFKUIWsetprotcodes@temp}%
4/0	\addtocounter(\FRUIwsizewtemp}{-128}%

476	\CPROT@aac@size{#1}%
477	}{%
478	\ifthenelse{\value{CPROT@size@temp}>63}{%
479	\protected@edef\CPROT@setprotcodes@temp{#1\protect\Large}%
480	\CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
481	\addtocounter{CPROT@size@temp}{-64}%
482	\CPROT@aac@size{#1}%
483	}{%
484	\ifthenelse{\value{CPROT@size@temp}>31}{%
485	\protected@edef\CPROT@setprotcodes@temp{#1\protect\large}%
486	\CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
487	\addtocounter{CPR0T@size@temp}{-32}%
488	\CPROT@aac@size{#1}%
489	}{%
490	\ifthenelse{\value{CPROT@size@temp}>15}{%
491	\protected@edef\CPR0T@setprotcodes@temp{#1\protect\normalsize}%
492	\CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
493	\addtocounterfCPRDT@size@temp}{-16}%
494	\CPR0T@aac@sizef#1}%
495	14%
496	\ifthenelse{\value{CPROT@size@temn}>7}{%
497	\protected@def(CPR0T@setprotcdes@temp{#1\protect\small}%
108	\CPRDT@ascHastcomp{\CPRDT@setprotcodesetemp[#T(protect(smail));
490	\addtaceutarf(DB0Tdeizedtamp1/s)
4 <i>33</i>	
500	\\FRUIWaaCWSIZE(#IJ%) \FW
501	<i>Γ</i> ιδ \ : feb an al a a f\ ua l ua f αDDOT α a i a a αt a mal > 21 f %
502	<pre>(illemenese()value()rollesize(cempf*/)rotort)footrotorigo)*/</pre>
505	(protectedueder ()rwisserprotectedesstempt#1(protect()rothotesizer/%)
504	CPRUIGACGUEXICODE (CPRUIGSEDFOCCOGESGUEMPS)
505	
506	\CPRUIGAACGS12e1#17%
507	
508	\lithenelset \value{\PKU @size@temp}>1}{%
509	\protecteduedet\UPKUIusstprotcodesutembl#1\protect\scriptsize}%
510	(CPRUI@aac@textcomp{ (CPRUI@setprotcoaes@temp}%
511	\addtocounter(JFKU10812eqtemp)(-2/%
512	\CPRUI@ac@size{#1}%
513	
514	(litheneise(Value(CPK))@size@temp}>0;%
515	\protected@edef\CPRUI@setprotcodes@temp{#1\protect\protect\tiny}%
516	\CPRUT@ac@textcomp{\CPRUT@setprotcodes@temp}%
517	
518	2.4 2.4 2.4
519	37
520	5% 
521	F%
522	3%
523	3%
524	3%
525	}%
526	3%
527	}
528	\newcommand{\CPROT@aac@textcomp}[1]{%
529	\ifthenelse{\boolean{CPR0T@textcomp}}{%
530	\@ifundefined{T@TS1}{%
531	\CPROT@packageinfo@or@warning{%
532	You've requested char protruding for TS1 encoding\MessageBreak
533	but TS1 encoding doesn't seem to be loaded.\MessageBreak
534	Maybe you forgot a '\string\usepackage{textcomp}'.%
535	}%
536	\setboolean{CPROT@textcomp}{false}%
537	}{}%
538	$\mathcal{F}$
539	\ifthenelse{\boolean{CPROT@textcomp}}{%
540	\protected@xdef\CPROT@setprotcodes@{%
541	\CPROT@setprotcodes@{%
	-

542	#1\protect\CPROT@setprotcodes@font{%
543	\fontencoding{TS1}\selectfont\protect\CPR0T@setprotcodes@font%
544	}%
545	}%
546	3%
547	}{%
548	\protected@xdef\CPROT@setprotcodes@{%
549	\CPR0T@setprotcodes@{%
550	#1\protect\CPROT@setprotcodes@font%
551	}%
552	}%
553	}%
554	}
otruding	
555	\newcommand*{\activatecharprotruding}[1][true]{%

# \activatecharpr

555	\newcommand*{\activatecharprotruding}[1][true]{%
556	%
557	\ifthenelse{\equal{#1}{true}\or\equal{#1}{on}\or\equal{#1}{1}\or\equal{#1}{%
558	\CPR0T@good@pdftex@version{%
559	\global\pdfprotrudechars=2%
560	}
561	}{%
562	\ifthenelse{\equal{#1}{compatibility}\or\equal{#1}{compatible}}{%
563	\CPROT@good@pdftex@version{%
564	\global\pdfprotrudechars=1%
565	}%
566	}{%
567	\ifthenelse{\equal{#1}{false}\or\equal{#1}{off}\or\equal{#1}{0}\or\equal{#1}{no}}{%
568	\global\pdfprotrudechars=0%
569	}{%
570	\CPROT@packageerror{%
571	''#1'' is no setting for \activatecharprotruding. Use\MessageBreak
572	one of 'true', 'on', 'yes', '1', or 'false', 'off', \MessageBreak
573	'no', '0', or 'compatible', 'compatibility' instead.%
574	}%
575	}{%
576	\CPROT@packageinfo@or@warning{%
577	You want to activate char protruding, but it does\MessageBreak
578	look as your are NOT using pdftex. So I can't\MessageBreak
579	activate it, as ''pdfcprot.sty'' only supports\MessageBreak
580	pdftex.}%
581	}%
582	}%
583	}%
584	}{}%
585	}
ding	
586	\newcommand*{\setupcharprotruding}[1]{%
587	%
588	\CPROT@resetall{}%
589	\@ifundefined{CPROT@save@KV@errx}{%
590	\let\CPROT@save@KV@errx=\KV@errx%
591	\renewcommand*{\KV@errx}[1]{%
592	\CPROT@packageerror{option ##1 for \string\setupcharprotruding}{%
593	You've tried to use the option ##1 with \string\setupcharprotruding.\MessageBreak
594	But there is no option with that name \MessageBreak

#### \setupcharprotru

uaing	
586	\newcommand*{\setupcharprotruding}[1]{%
587	%
588	\CPROT@resetall{}%
589	\@ifundefined{CPROT@save@KV@errx}{%
590	\let\CPROT@save@KV@errx=\KV@errx%
591	\renewcommand*{\KV@errx}[1]{%
592	\CPROT@packageerror{option ##1 for \string\setupcharprotruding}{%
593	You've tried to use the option ##1 with \string\setupcharprotruding.\MessageBreak
594	But there is no option with that name.\MessageBreak
595	See the manual of ''pdfcprot.sty'' for information
596	about the usage of $\string\setupcharprotruding.%$
597	}%
598	}%
599	\setkeys{CPROT}{#1}%
600	\let\KV@errx=\CPROT@save@KV@errx%
601	<pre>\let\CPROT@save@KV@errx=\relax%</pre>
602	}{%

603 604

614

\setkeys{CPROT}{#1}% }%

605

\CPROT@setprotcodes@add{}% 606

}{%

\CPR01	[@pacl	kage	einfo@d	or@wai	rning{%
You	want	to	setup	char	protru

You want to setup char protruding, but it does\MessageBreak look as if you're NOT using pdftex. So I can't\MessageBreak setup it, as ''pdfcprot.sty'' only supports\MessageBreak pdftex.}%

611}%

}

612

613

```
\Conlypreamble\setupcharprotruding%
```

#### $\verb+setupcharprotruding numeral+$

615	\newcommand*{\setupcharprotrudingnumeral}[1]{%
616	%
617	\CPROT@resetall{}%
618	\@ifundefined{CPROT@save@KV@errx}{%
619	\let\CPR0T@save@KV@errx=\KV@errx%
620	\renewcommand*{\KV@errx}[1]{%
621	\CPROT@packageerror{option ##1 for \string\setupcharprotrudingnumeral}{%
622	You've tried to use the option ##1 with \string\setupcharprotrudingnumeral.\MessageBreak
623	But there is no option with that name.\MessageBreak
624	See the manual of ''pdfcprot.sty'' for information
625	about the usage of \string\setupcharprotrudingnumeral.%
626	}%
627	}%
628	\setkeys{CPROTnum}{#1}%
629	\let\KV@errx=\CPROT@save@KV@errx%
630	\let\CPROT@save@KV@errx=\relax%
631	}{%
632	\setkeys{CPROTnum}{#1}%
633	}%
634	\CPROT@setprotcodes@add{}%
635	<b>}{%</b>
636	\CPROT@packageinfo@or@warning{%
637	You want to setup char protruding, but it does\MessageBreak
638	look as if you're NOT using pdftex. So I can't\MessageBreak
639	setup it, as ''pdfcprot.sty'' only supports\MessageBreak
640	pdftex.}%
641	}%
642	}
643	\@onlypreamble\setupcharprotrudingnumeral

#### \ifpdftex

644	\newcommand{\ifpdftex}{%
645	\ifx\pdftexversion\@undefined
646	\expandafter\@secondoftwo
647	\else
648	\ifx\pdftexversion\relax
649	$\sum \sqrt{\frac{1}{2}}$
650	\else
651	$\exp \pi \pi$
652	\fi
653	\fi
654	}

#### 

655	\newcommand{\CPROT@packageerror@bad@pdftex}{%
656	\CPROT@packageerror{You used a pdftex version older than 0.14f.\MessageBreak
657	pdfcprot does not support such old versions of pdftex. \MessageBreak
658	Please install a new version of pdftex.}%
659	}
660	\newcommand{\CPROT@good@pdftex@version}[1]{%
661	\ifnum\pdftexversion < 14

662	\CPR0T@packageerror@bad@pdftex %
663	\else
664	\ifnum\pdftexversion = 14
665	\ifnum \expandafter'\pdftexrevision < 'f
666	\CPROT@packageerror@bad@pdftex %
667	\else
668	#1
669	\fi
670	
671	#1
672	π1 \f;
673	\11 \fi
674	\[ \
074	r
\ifpdfoutput	This was taken from KOMA-Script and is provided for convenience.
675	\norrespond+1\CDD0TAifndfoutnutLfV
675	\iekcomeand*1\CFR01@IIpuf0utput/%
070	(licase 0%
677	\11X\paroutput\@underined 1%
678	\else
679	\irx\pdfoutput\relax 1%
680	\else
681	\ifcase\pdfoutput 1%
682	\fi
683	\fi
684	\fi
685	\space
686	\expandafter\@firstoftwo
687	\else
688	\expandafter\@secondoftwo
689	\fi
690	}
691	\@ifundefined{ifpdfoutput}{\let\ifpdfoutput\CPROT@ifpdfoutput}{%
692	\CPROT@packageinfo{%
693	\string\ifpdfoutput\space already defined.\MessageBreak
694	That may happen when using KOMA-Script together\MessageBreak
695	with "'pdfcprot.sty''. I have changed the definition\MessageBreak
696	from Markus' KOMA-Script, so if you experience errors\MessageBreak
697	try \string\let\string\ifdfoutbut\string\relax\space before loading ''pdfcprot.sty''. }%
698	}
699	\newcommand*{\CPROT@packageoptions}[1]{%
700	\@ifundefined{CPROT@save@KV@errx}{%
701	\let\CPROT@save@KV@errx=\KV@errx%
702	\renewcommand*{\KV@errx}[1]{%
703	\PackageError{pdfcprot}{unknown option ``#1`'}{%
704	You've tried to use the option ''#1''.\MessageBreak
705	But there is no option with that name. $MessageBreak$
706	See the manual of ''pdfcprot.sty'' for information
707	about known options.%
708	}%
709	3%
710	\setkeys{CPROTpackage}{#1}%
711	\let\KV@errx=\CPROT@save@KV@errx%
712	\let\CPROT@save@KV@errx=\relax%
713	}{%
714	\setkeys{CPR0Tpackage}{#1}%
715	}%
716	}
717	\DeclareOption{DVIoutput}{%
718	\CPR0T@nackagewarning{%
710	It is not a good idea to use the DVLoutput nackage
720	ontion. \MessageBreak
720	Use it only if you experience some using other
722	nackages. MessageBreak
723	And don't forget to file a bug report against these
724	packages.\MessageBreak

725 726	The best is to add just \string\pdfoutput=0\space at the beginning\MessageBreak of your preamble or even before \string\documentclass.
727	} \:====================================
728	\ipaitex1%
730	\let\pdfoutput\undefined
731	$+\Omega^{*}$
732	}
	Define a standard command for undefined options. Actually this is just a redirection to a <b>\setkeys</b> command.
733	\DeclareOption*{%
734	%
735	\expandafter\CPROT@packageoptions\expandafter{\CurrentOption}%
736	}{}%
737	<i>}%</i>
\CPR0I@fallbackfont	The internal command \CPROT@fallbackfont stores the font to be looked for, if no CPA is
\setfallbackfont	found for the actual font. By default it's Palatino. With \setfallbackfont it is possible
	to customise this, if required.
738	\newcommand*{\CPROT0fallhackfont}{nn}
739	\newcommand*{\setfallbackfont}[1]{\renewcommand*{\CPROT@fallbackfont}{#1}}
ra	When the people is loaded with ention estimate or estimate rearrant this command
lesetupcharprotrudingenormal	when the package is loaded with option activate of activate-normal this command
	does the actual adjustment and by that defines which fonts will get character protrucing
	with that options.
740	\newcommand*{\CPROT@setupcharprotruding@normal}{%
741	\setupcharprotrudingnumeral{tam1y=3, series=1, shape=8, size=60, textcomp=1}%
742	\setupcharprotrudingnumeral{Iamily=2,series=2,shape=12,size=60,textcomp=1}%
743	}
tupcharprotrudingforencoding	This command
745	\newcommand*{\setupcharprotrudingforencoding}[1]{%
746	\setupcharprotrudingnumeral{encoding={#1},iamily=3,series=1,snape=0,size=60,textcomp=1}%
748	<pre>\setucharprotrudingnumeral{encoding={#}, family=1.series=2.shape=14.size=60.textcomp=1}</pre>
749	}
750	\@onlypreamble\setupcharprotrudingforencoding
pdfcprot.cfg	
751	\InputIfFileExists{pdfcprot.cfg}
752	{************************************
753	<pre>* Local config file pdfcrpot.cfg used *^^J%</pre>
754	***************************************
755	$\{ \}$
756	\ProcessOptions\relax
757	\newcommand*{\CPROT@actualfont}{%
758	\f@family%
759	\f0series%
760	\f@shape%
761	\I@encoding%
762	5 \@onlypreamble\setfallbackfont%
764	\newcommand*{\CPROT@cpa@toload}{}
\CPROT@setprotcodes@font	This command you've seen before in the definition of \CPROT@setprotcodes@. There it
	was protected so it won't be expanded to early. That was important, to test for the
765	\newcommand*{\CPROT@setprotcodes@font}{%
766	\renewcommand*{\\CFKUI@cpa@toload}{\CFKUI@actualfont.cpa}% \CPROT@cpa@tolosd\J%
101	TIL TTERVISIS ( /01 101 AC bas 101 00 1/0

Former version used a general command beeing redefined any time directly from the CPA. Now the CPA defines a new command we can test for, so it will be loaded just once. The same code is found in the part for the fallback font. Using this code some time will be spared (I hope).

768	\expandafter\@ifundefined\expandafter{\CPROT@actualfont}{%
769	\input{\CPROT@cpa@toload}
770	<del>}</del> 0

As the fontencoding names contain numbers \csname and \endcsname must be used to call the actual command to setup character protruding. After that a adjustment may be needed.

771	\csname\CPROT@actualfont\endcsname%
772	\CPROT@adjustprotcodes{\font}%
773	\CPROT@packageinfo{%
774	File \CPROT@cpa@toload{} for the adjustment of\MessageBreak
775	the char protruding used. It seems to be _fit_\MessageBreak
776	to the font you requested.%
777	}%
778	}{%
779	\CPR0T@packageinfo@or@warning{%
780	I didn't find \CPROT@cpa@toload.\MessageBreak%
781	\space As a fallback I will try \CPROT@fallbackfont\f@encoding.cpa.%
782	}%
783	$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $
784	\IfFileExists{\CPROT@cpa@toload}{%
785	\expandafter\@ifundefined\expandafter{\CPROT@fallbackfont\f@encoding}{%
786	\input{\CPROT@cpa@toload}%
787	}-{} <sup>™</sup> / <sub>2</sub>
788	$\csname\CPROT@fallbackfont\f@encoding\endcsname%$
789	\CPROT@adjustprotcodes{\font}%
790	\CPROT@packageinfo{%
791	File \CPROT@cpa@toload{} used for the adjustment\MessageBreak
792	of the char protruding. It seems that it isn't the one\MessageBreak
793	you intended to use.%
794	3%
795	}{%
796	\CPROT@packageinfo@or@warning{%
797	Even the fallback font was not found. Maybe it's\MessageBreak
798	not the right font encoding. Currently "pdfcprot" MessageBreak
799	bundles only with cpa's for T1, OT1 and TS1 encoding.%
800	}%
801	}%
802	3%
803	}
804	\newif\ifCPROT@adjustprotcodes@
805	\newcommand*{\CPROT@adjustprotcodes}[1]{%
806	\CPROT@adjustprotcodes@false
807	\ifnum\pdftexversion > 14
808	\CPROT@adjustprotcodes@true
809	\else
810	\ifnum\pdftexversion = 14
811	\ifnum \expandafter'\pdftexrevision > 'g
812	\CPROT@adjustprotcodes@true
813	\fi
814	\fi
815	\fi
816	\ifCPROT@adjustprotcodes@
817	\@tempcnta=0
818	\loop
819	\ifcase\lpcode#1\@tempcnta\else
820	\CPROT@adjustprotcodes@@\lpcode{#1}\@tempcnta
821	\fi
822	\ifcase\rpcode#1\@tempcnta\else
823	\CPROT@adjustprotcodes@@\rpcode{#1}\@tempcnta
824	\fi

825	\advance\@tempcnta 1
826	\ifnum\@tempcnta < 256 \repeat
827	\fi
828	}
829	\def\CPROT@adjustprotcodes@@#1#2#3{%
830	\setbox0=\hbox{\the#2\char#3}%
831	% \setbox0=%
832	% \ifx#2\font\else#2\fi%
833	% \char#3}%
834	\@tempcntb=\wd0%
835	\multiply\@tempcntb #1#2#3%
836	\divide\@tempcntb \fontdimen6 #2%
837	#1#2#3=\@tempcntb%
838	}
839	
840	\newcommand*{\CPROT@setprotcodes}{%
841	%
842	\ifthenelse{\equal{\CPROT@setprotcodes@}{}}{%
843	}{%
844	\CPROT@setprotcodes@%
845	}%
846	}{}%
847	}
848	
849	%
850	\CPROT@setprotcodes\relax%
851	}
852	$\langle / package  angle$

## A.2 The Provided .CPA Files

### A.2.1 Palatino

\pplmnT1

\pplmsg@T1	$\langle pplmnT1 \rangle $ \expandafter \gdef \csname pplmnT1 \endcsname {%
854	$\langle pplmnOT1 \rangle \$
855	$\langle pp mnT2A \rangle \langle expandafter \rangle gdef \langle sname pp mnT2A \rangle endcsname {%}$
856	<pre>(pplmnOT2)\expandafter\gdef\csname pplmnOT2%</pre>
857	(*pplmnOT1   pplmnOT2)
858	<pre>\lpcode\font 92=500 % ''</pre>
859	\rpcode\font 34=500 % ''
860	\rpcode\font 123=300 %
861	\rpcode\font 124=200 %
862	$\langle /pplmnOT1   pplmnOT2 \rangle$
863	⟨∗pplmnT1   pplmnT2A⟩
864	<pre>\lpcode\font 16=500 % ''</pre>
865	\rpcode\font 17=500 % ''
866	\rpcode\font 21=300 %
867	\rpcode\font 22=200 %
868	$\langle /pplmnT1 \mid pplmnT2A \rangle$
869	$\langle *pplmnT2A \mid pplmnOT2 \rangle$
870	\rpcode\font\cyrdash=200 % "
871	<pre>\lpcode\font\cyrdash=200 % "*</pre>
872	$\langle /pplmnT2A \mid pplmnOT2 \rangle$
873	$\langle * pplmnT1 \mid pplmnT2A  angle$
874	% german quotation marks
875	\lpcode\font\quotedblbase=600
876	\rpcode\font\textquotedblleft=500
877	$\langle /pplmnT1 \mid pplmnT2A  angle$
878	$\langle * pplmnOT1 \mid pplmnT1 \mid pplmnT2A \mid pplmnOT2  angle$
879	<pre>% set the protrusion of ",","-" and "." a bit smaller</pre>
880	% than originally suggested by Han The Than
881	\rpcode\font'=650
882	\rpcode\font'\-=650
883	\rpcode\font'\.=650
884	% originial Setting from Han The Thans protcode.tex

885	\rpcode\font'\!=200
886	\rpcode\font'\;=500
887	\rpcode\font'\:=500
888	\rpcode\font'\?=200
889	<pre>\lpcode\font'\'=700</pre>
890	rpcode(font') = 700
891	(rpcode (Iont ()=50
892	
893	\rpcode\font`\A=\rpcode\font`\A
894	(pplmnll) \rpcode\font 196=50 % A umlaut
895	\rpcode\font \F=50
896	\rpcode\font'\K=50
897	\rpcode\font'\L=50
898	\rpcode\font'\T=50
899	\rpcode\font'\V=50
900	\rpcode\font'\W=50
901	\rpcode\font'\X=50
902	\rpcode\font'\Y=50
903	\rpcode\font'\k=50
904	\rpcode\font`\r=50
905	rpcode font' t=50
906	rpcode font' v=50
907	\rpcode\font'\w=50
908	rpcode font' x=50
909	\rpcode\font'\y=50
910	$\langle /!pplmnOT2 \rangle$
911	\lpcode\font`\(=50
912	$\langle *!pplmnOT2 \rangle$
913	\lpcode\font'\A=50
914	$pplmnT1 \setminus lpcode font 196=lpcode font 'A % A umlaut'$
915	\lpcode\font'\J=50
916	\lpcode\font'\T=50
917	\lpcode\font'\V=50
918	\lpcode\font'\W=50
919	\lpcode\font'\X=50
920	\lpcode\font'\Y=50
921	lpcode font' v=50
922	\lpcode\font`\w=50
923	\lpcode\font'\x=50
924	\lpcode\font'\y=50
925	$\langle /!pplmnOT2 \rangle$
926	$\langle * pplmnT2A \mid pplmnOT2 \rangle$
927	\rpcode\font\CYRA=50
928	\rpcode\font\CYRK=50
929	\rpcode\font\CYRT=50
930	\rpcode\font\CYRH=50
931	\rpcode\font\CYRU=50
932	\rpcode\font\CYRG=50
933	\rpcode\font\CYRZH=50
934	\rpcode\font\CYRL=50
935	\rpcode\font\CYRC=20
936	\rpcode\font\CYRSHCH=10
937	\rpcode\font\CYRHRDSN=50
938	\rpcode\font\CYRSFTSN=50
939	\rpcode\font\cvrk=50
940	\rpcode\font\cyrt=50
941	\rpcode\font\cyrb=50
942	$\rcode$ $font \cvru=50$
344	Theore I our least a co

943	\rpcode\font\cyrg=50
944	\rpcode\font\cyrzh=50
945	\rpcode\font\cyr1=50
946	\rpcode\font\cyrc=20
947	rpcodefont cyrshch=10
948	rpcodefont cyrhrdsn=50
949	rpcodefont cyrsftsn=50
950	$\line \font \CYRA=50$
951	$\line \font \CYRT=50$
952	$\line \font CYRH=50$
953	$\line \font CYRU=50$
954	$\line \font CYRZH=50$
955	$\line \font \CYRL=50$
956	$\line \font CYRU=50$
957	$\line \font CYRCH=50$
958	$\line \font CYRHRDSN=50$
959	$\line \font CYRD=50$
960	$\lpcode\font\cyra=50$
961	lpcode font cyrt=50
962	lpcode font cyrh=50
963	$\lpcode\font\cyru=50$
964	$\line \font\cyrzh=50$
965	lpcodefont cyr1=50
966	$\lpcode\font\cyru=50$
967	$\lpcode\font\cyrch=50$
968	$\lpcode\font\cyrhrdsn=50$
969	lpcode font cyrd=50
970	⟨/pplmnT2A   pplmnOT2⟩
971	$f_{\lambda}$
972	$\langle ppimnOTT   ppimnTT   ppimnTZA   ppimnOT2 \rangle$

#### \pplmnTS1

973	$\langle * pplmnTS1 \rangle$
974	\expandafter\gdef\csname pplmnTS1%
975	\rpcode\font 176=600 % \textdegree
976	}
977	(/pplmnTS1)