# The `pdfcprot.sty` Package.*

Carsten Schurig†, Tobias Schlemmer

June 10, 2005

**Abstract**

This package was written to provide the "normal" LaTeX 2ε 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

As the abstract stated this package exists to provide a simple user interface to the character protruding feature of pdfTeX[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

*This file has version v1.7a dated 2005/05/23.

†This package depends intensely on code originally from Hàn Thế Thành.

[1] You'll find pdfTeX 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 LaTeX 2ε 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 TeX, but that it is not without problem. As an example the LaTeX-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 LaTeX 2ε and one would have to wait for LaTeX3 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 LaTeX 2ε 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 LaTeX. Shortly after getting this information I found `protcode.tex` provided by Hàn Thế Thành.

That I transfered to a packaged for my own use and as some people showed interests on `de.comp.text.tex` I posted it there.[3] 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 pdfTeX (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 pdfTeX).

## 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 LaTeX 2ε 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 pdfTeX output to `DVI` and a command testing, whether you use `PDF` output or not are provided.[4]

# 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 pdfTeX to get any effect, but the package just gives a warning if you don't.

[2] It could be found on the web, but I don't remember where I got it.
[3] The first version was named `optrandausgleich.sty` and a second `pdftexfeats.sty`.
[4] 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 LaTeX 2ε 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

For the "normal" LaTeX 2ε user, who just wants to activate the character protruding one option is provided.

```
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 \rmfamily\mdseries\upshape and maybe \sffamily \mdseries\upshape using activate or activate=normal as package option. That is done because I don't want to bother the "normal" LaTeX 2ε-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 LaTeX 2ε 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. `rmfamily` and `sffamily` are valid values. Any other value will be supposed as a low level font name (see below). *This tag is mandatory.*

`series` This tag is used to chose the font series. Valid values are `mdseries` and `bfseries`. Any other value will be supposed as a low level font name (see below). *This tag is mandatory.*

`shape` The font shape is chosen using this tag. As for the other tags valid values are the high level LaTeX 2ε 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 2ε 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 protruding 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 level font selection.** The "normal" parameters for font selection are the high level LaTeX 2ε 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`.

```
\usepackage[activate=none]{pdfcprot}
\setupcharprotruding{encoding=T2A,family=rmfamily,series=elec,
                     shape=ui,size=normalsize,
                     textcomp=false}
\activatecharprotruding
```

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

```
\usepackage[activate]{pdfcprot}
\setupcharprotruding{family=rmfamily,series=bfseries,
                     shape=upshape,size=normalsize,
                     textcomp=false}
\activatecharprotruding
```

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 LaTeX 2ε 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 LaTeX 2ε 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 LaTeX 2ε 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.[5] 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 $2_\varepsilon$ 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" 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 TeX 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`).

[5] The commands used to determine which font is loaded (`\f@family`, `\f@series`, `\f@shape`, `\f@encoding`) return the font requested to TeX *not* the font actually used, so it seems. That means for instance if you request a bold Palatino font with `\bfseries`, TeX 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 pdfTEX.

First there's is an option to get a DVI output using pdfTEX.

`DVIoutput`

This may be useful if one can't switch completely to pdfTEX, because in this mode all the additional features of pdfTEX[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`.[7]

Further on there are two LATEX $2_\varepsilon$-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[8].

The command `\ifpdftex` is very similar but leads to a decision depending on the usage of pdfTEX or not. This is useful for the activation of special features of pdfTEX 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`.[9]

## 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.

[6] That's not completely true though. The supported type of images is in DVI mode the same as using `latex`.

[7] Most actual versions of packages having had problems may treat it correctly by now. But who has all packages installed in most recent versions?

[8] If you want a TEX `\if` switch, look for `ifpdf.sty` on CTAN written by Heiko Oberdiek.

[9] That is a suggestion of myself. I don't know what the authors of pdfTEX 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 LaTeX 2ε 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 pdfTeX have to be mentioned. Some code comes directly from them.

# References

[1] Hàn Thế Thành. *Micro-typographic extensions to the TeX 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 LaTeX 2ε 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   ⟨*package⟩
```

First the requirement of LaTeX 2ε[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 LaTeX 2ε-distribution `;->`.

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

[10] 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 LaTeX 2ε 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@`.

```
\newcounter{CPROT@family}
\newcounter{CPROT@family@temp}
\newcounter{CPROT@series}
\newcounter{CPROT@series@temp}
\newcounter{CPROT@shape}
\newcounter{CPROT@shape@temp}
\newcounter{CPROT@size}
\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}
```

`quiet` For providing the `quiet` option a (global) boolean is used. By default it is set to `true`, so there're little warnings (but it will be written as info to the logs).

```
\newboolean{CPROT@quiet}
\setboolean{CPROT@quiet}{true}
```

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

```
\newcounter{CPROT@temp@chars}
```

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

```
\newcommand*{\CPROT@lowlevel@encoding}{}
\newcommand*{\CPROT@lowlevel@family}{}
\newcommand*{\CPROT@lowlevel@series}{}
\newcommand*{\CPROT@lowlevel@shape}{}
```

`\CPROT@resetall` This is to reset all the counters to zero, so that a new `\setupcharprotruding`... command won't activate something requested with a former command.

```
\newcommand*{\CPROT@resetall}{%
  \setcounter{CPROT@family}{0}%
  \setcounter{CPROT@series}{0}%
  \setcounter{CPROT@shape}{0}%
  \setcounter{CPROT@size}{0}%
  \setboolean{CPROT@textcomp}{false}%
  \edef\CPROT@lowlevel@encoding{}%
}
```

`\CPROT@setprotcodes@` `\CPROT@setprotcodes@add` `\CPROT@setprotcodes@temp` These are internal macros regarding the creation of the actual command, to adjust the character protruding for the fonts wanted. `\CPROT@setprotcodes@` actually will contain the command, whereas `\CPROT@setprotcodes@temp` is used to store part of this command temporary. `\CPROT@setprotcodes@add` is called by `\setupcharprotruding` and `\setupcharprotrudingnumeral`.

```
\newcommand*{\CPROT@setprotcodes@}{}
\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}}%
```

Here the actual evaluation will take place starting with the font encoding.

```
    \CPROT@aac@encoding%
}
\newcommand*{\CPROT@setprotcodes@temp}{}
```

\CPROT@packageinfo
\CPROT@packagewarning
\CPROT@packageerror

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) and it's simplier to customise if I would want to do something other than normal.

```
\newcommand{\CPROT@packageinfo}[1]{%
  \PackageInfo{pdfcprot}{#1}%
  }
\newcommand{\CPROT@packagewarning}[1]{%
  \PackageWarning{pdfcprot}{#1}%
  }
\newcommand{\CPROT@packageerror}[1]{%
  \PackageError{pdfcprot}{#1}%
  }
```

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.

```
\newcommand{\CPROT@packageinfo@or@warning}[1]{%
  \ifthenelse{\boolean{CPROT@quiet}}{%
    \CPROT@packageinfo{#1}%
    }{%
    \CPROT@packagewarning{#1}%
    }%
  }%
```

encoding

```
\define@key{CPROT}{encoding}{%
  \ifthenelse{\equal{#1}{encodingdefault}}{%
    \edef\CPROT@lowlevel@encoding{\encodingdefault}% \encodingdefault will be set.
  }{% I assume a low level encodingname is given
    \renewcommand*{\CPROT@lowlevel@encoding}{#1}%
  }%
}
```

family

```
\define@key{CPROT}{family}{%
  \ifthenelse{\equal{#1}{rmfamily}}{%
    \setcounter{CPROT@family}{2}% \rmfamily will be set
    }{%
    \ifthenelse{\equal{#1}{sffamily}}{%
      \setcounter{CPROT@family}{1}% \sffamily will be set
      }{% I assume a low level fontname is given
      \setcounter{CPROT@temp@chars}{0}%
      \expandafter\@tfor\expandafter\@tempb\expandafter:\expandafter=#1\do{%
        \stepcounter{CPROT@temp@chars}%
        }%
      \ifthenelse{\value{CPROT@temp@chars}>4}{%
        \CPROT@packageinfo@or@warning{None of the keywords 'all',\MessageBreak
          'rmfamily' or 'sffamily' was found. I assume you gave a\MessageBreak
          low level font name but it is at least unusual that a\MessageBreak
          TeX-fontname has more than 4 characters. I assume you know\MessageBreak
          what you're doing and continue, but the selection of the\MessageBreak
          font may fail!%
          }%
        \CPROT@packageinfo{It seem's as you want to use a low level\MessageBreak
          command for font selection. -- You're for yourself.%
          }%
        }{}%
      \setcounter{CPROT@family}{-1}%
      \renewcommand*{\CPROT@lowlevel@family}{#1}%
      }%
    }%
  }
```

series

```
\define@key{CPROT}{series}{%
```

```
      \ifthenelse{\equal{#1}{mdseries}}{% medium series
        \setcounter{CPROT@series}{2}%
        }{%
        \ifthenelse{\equal{#1}{bfseries}}{% bold series
          \setcounter{CPROT@series}{1}%
          }{% here the low level interface is suspected
          \setcounter{CPROT@temp@chars}{0}%
          \expandafter\@tfor\expandafter\@tempb\expandafter:\expandafter=#1\do{%
            \stepcounter{CPROT@temp@chars}%
            }%
          \ifthenelse{\value{CPROT@temp@chars}>4}{%
            \CPROT@packageinfo@or@warning{None of the keywords 'all',\MessageBreak
              'mdseries' or 'bfseries' was found. I assume you gave a\MessageBreak
              low level font name but a series identifier should consist\MessageBreak
              of 4 characters at maximum. I assume you know what you're\MessageBreak
              doing and continue, but the selection of the font will\MessageBreak
              very likely fail!%
              }%
            \CPROT@packageinfo{It seem's as you seem want to use a low level\MessageBreak
              command for font selection. -- You're for yourself.%
              }%
            }{}%
          \setcounter{CPROT@series}{-1}%
          \renewcommand*{\CPROT@lowlevel@series}{#1}%
          }%
        }%
      }%
```

shape

```
    \define@key{CPROT}{shape}{%
      \ifthenelse{\equal{#1}{upshape}}{%
        \setcounter{CPROT@shape}{8}%
        }{%
        \ifthenelse{\equal{#1}{itshape}}{%
          \setcounter{CPROT@shape}{4}%
          }{%
          \ifthenelse{\equal{#1}{slshape}}{%
            \setcounter{CPROT@shape}{2}%
            }{%
            \ifthenelse{\equal{#1}{scshape}}{%
              \setcounter{CPROT@shape}{1}%
              }{% low lewel format
              \setcounter{CPROT@temp@chars}{0}%
              \expandafter\@tfor\expandafter\@tempb\expandafter:\expandafter=#1\do{%
                \stepcounter{CPROT@temp@chars}%
                }%
              \ifthenelse{\value{CPROT@temp@chars}>2}{%
                \CPROT@packageinfo@or@warning{None of the keywords 'all',\MessageBreak
                  'upshape', 'itshape', 'slshape' or 'scshape' was\MessageBreak
                  found. I assume you gave a low level font name but it\MessageBreak
                  is at least unusual that a shape identifier has more\MessageBreak
                  than 2 characters. I assume you know what you're doing\MessageBreak
                  and continue, but the selection of the font will very\MessageBreak
                  likely fail!%
                  }%
               \CPROT@packageinfo{It seem's as you seem want to use a low level\MessageBreak
                  command for font selection. -- You're for yourself.%
                  }%
                }{}%
              \setcounter{CPROT@shape}{-1}%
              \renewcommand{\CPROT@lowlevel@shape}{#1}%
              }%
            }%
          }%
        }%
      }
```

size

```
    \define@key{CPROT}{size}{%
      \ifthenelse{\equal{#1}{Huge}}{%
        \setcounter{CPROT@size}{512}%
        }{%
        \ifthenelse{\equal{#1}{huge}}{%
          \setcounter{CPROT@size}{256}%
          }{%
          \ifthenelse{\equal{#1}{LARGE}}{%
            \setcounter{CPROT@size}{128}%
            }{%
            \ifthenelse{\equal{#1}{Large}}{%
              \setcounter{CPROT@size}{64}%
              }{%
              \ifthenelse{\equal{#1}{large}}{%
                \setcounter{CPROT@size}{32}%
                }{%
                \ifthenelse{\equal{#1}{normalsize}}{%
                  \setcounter{CPROT@size}{16}%
                  }{%
                  \ifthenelse{\equal{#1}{small}}{%
                    \setcounter{CPROT@size}{8}%
                    }{%
                    \ifthenelse{\equal{#1}{footnotesize}}{%
                      \setcounter{CPROT@size}{4}%
                      }{%
                      \ifthenelse{\equal{#1}{scriptsize}}{%
                        \setcounter{CPROT@size}{2}%
                        }{%
                        \ifthenelse{\equal{#1}{tiny}}{%
                          \setcounter{CPROT@size}{1}%
                          }{% low lewel format
                          \CPROT@packageerror{%
                           None of the keywords 'all', 'normal' or any LaTeX2e\MessageBreak
                           font name was found. \MessageBreak
                           \space For more information how to use\MessageBreak
                           \string\setupcharprotrudingnumeral see the
                           pdfcprot manual.%
                           }%
                          }{}%
                        }%
                      }%
                    }%
                  }%
                }%
              }%
            }%
          }%
        }%
      }
    \define@key{CPROT}{textcomp}{%
      \ifthenelse{\equal{#1}{true}\or\equal{#1}{on}\or\equal{#1}{1}\or\equal{#1}{yes}}{%
        \setboolean{CPROT@textcomp}{true}%
        }{%
        \ifthenelse{\equal{#1}{false}\or\equal{#1}{off}\or\equal{#1}{0}\or\equal{#1}{no}}{%
          \setboolean{CPROT@textcomp}{false}%
          }{%
          \CPROT@packageerror{%
            ``#1'' is no setting for ``textcomp''. Use\MessageBreak
            one of 'true', 'on', 'yes', '1', or 'false', 'off', \MessageBreak
            'no', '0' instead.\MessageBreak
            }%
          }%
        }%
      }
    \define@key{CPROTnum}{encoding}{%
```

```
    \ifthenelse{\equal{#1}{0}}{%
      \renewcommand*{\CPROT@lowlevel@encoding}{}% No encoding will be set.
    }{%
      \ifthenelse{\equal{#1}{1}}{%
        \edef\CPROT@lowlevel@encoding{\encodingdefault}% \encodingdefault will be set.
      }{%
        \ifthenelse{\equal{#1}{encodingdefault}}{%
          \edef\CPROT@lowlevel@encoding{\encodingdefault}% \encodingdefault will be set.
        }{% I assume a low level encodingname is given
          \renewcommand*{\CPROT@lowlevel@encoding}{#1}%
        }%
      }%
    }%
  }
  \define@key{CPROTnum}{family}{%
    \ifthenelse{#1>0 \and #1<4}{%
      \setcounter{CPROT@family}{#1}%
      }{%
      \ifthenelse{#1=0}{%
        \CPROT@packageinfo{%
          As you selected '0' (that means 'none') for\MessageBreak
          selection of the font family, no font will\MessageBreak
          be set up for char protruding.%
          }%
        }{%
        \CPROT@packageerror{%
        Your given value '#1' to family in \string\setupcharprotrudingnumeral\MessageBreak
          is not valid. Sensible values are between 1 and 3 (inclusive).\MessageBreak
          For further information see the pdfcprot manual.%
          }%
        }%
      }%
    }
  \define@key{CPROTnum}{series}{%
    \ifthenelse{#1>0 \and #1<4}{%
      \setcounter{CPROT@series}{#1}%
      }{%
      \ifthenelse{#1=0}{%
        \CPROT@packageinfo{%
          As you selected '0' (that means 'none') for\MessageBreak
          selection of the font series, no font will\MessageBreak
          be set up for char protruding.%
          }%
        }{%
        \CPROT@packageerror{%
        Your given value '#1' to series in \string\setupcharprotrudingnumeral\MessageBreak
          is not valid. Sensible values are between 1 and 15 (inclusive).\MessageBreak
          For further information see the pdfcprot manual.%
          }%
        }%
      }%
    }
  \define@key{CPROTnum}{shape}{%
    \ifthenelse{#1>0 \and #1<16}{%
      \setcounter{CPROT@shape}{#1}%
      }{%
      \ifthenelse{#1=0}{%
        \CPROT@packageinfo{%
          As you selected '0' (that means 'none') for\MessageBreak
          selection of the font shape, no font will\MessageBreak
          be set up for char protruding.%
          }%
        }{%
        \CPROT@packageerror{%
         Your given value '#1' to shape in \string\setupcharprotrudingnumeral\MessageBreak
          is not valid. Sensible values are between 1 and 15 (inclusive).\MessageBreak
```

```
            For further information see the pdfcprot manual.%
            }%
          }%
        }%
      }
    \define@key{CPROTnum}{size}{%
      \ifthenelse{#1>0 \and #1<1024}{%
        \setcounter{CPROT@size}{#1}%
        }{%
        \ifthenelse{#1=0}{%
          \CPROT@packageinfo{%
            As you selected '0' (that means 'none') for\MessageBreak
            selection of the font size, no font will\MessageBreak
            be set up for char protruding.%
            }%
          }{%
          \CPROT@packageerror{%
            Your given value '#1' to size in \string\setupcharprotrudingnumeral\MessageBreak
            is not valid. Sensible values are between 1 and 1023 (inclusive).\MessageBreak
            For further information see the pdfcprot manual.%
            }%
          }%
        }%
      }
    \define@key{CPROTnum}{textcomp}[true]{%
      \setkeys{CPROT}{textcomp=#1}%
    }
```

activate

```
    \define@key{CPROTpackage}{activate}[normal]{%
      \ifthenelse{\equal{#1}{normal}}{%
        \CPROT@setupcharprotruding@normal%
        \ifthenelse{\equal{\CPROT@setprotcodes@}{}}{%
        }{% just activate char prot when something was set
          \activatecharprotruding[true]%
        }%
      }{%
        \ifthenelse{\equal{#1}{none}}{%
        }{%
          \CPROT@packageerror{%
            Value of activate can be 'normal' or 'none'.\MessageBreak%
            \space For more low level setup use \string\setupcharprotruding\MessageBreak%
            and see the manual%
            }%
          }%
        }%
      }%

    \define@key{CPROTpackage}{quiet}[true]{%
      \ifthenelse{\equal{#1}{true}\or\equal{#1}{on}\or\equal{#1}{yes}\or\equal{#1}{1}}{%
        \setboolean{CPROT@quiet}{true}%
        }{%
        \ifthenelse{\equal{#1}{false}\or\equal{#1}{off}\or\equal{#1}{no}\or\equal{#1}{0}}{%
          \setboolean{CPROT@quiet}{false}%
          }{%
          \CPROT@packageerror{%
            Value of quiet should be 'true' ('on', 'yes, '1') or\MessageBreak%
            'false' ('off', 'no', '0'). You didn't seem to use any\MessageBreak%
            of them.%
            }%
          }%
        }%
      }%
```

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

```
\newcommand{\CPROT@aac@encoding}{%
  \ifthenelse{\equal{\CPROT@lowlevel@encoding}{}}{%
      \setcounter{CPROT@family@temp}{\value{CPROT@family}}%
      \CPROT@aac@family{}%
    }{%
      \@for \CPROT@aac@encoding@first:=\CPROT@lowlevel@encoding \do {%
        \setcounter{CPROT@family@temp}{\value{CPROT@family}}%
        \expandafter\ifx\csname T@\CPROT@aac@encoding@first\endcsname\relax%
          \ifthenelse{\boolean{CPROT@quiet}}{%
         \CPROT@packagewarning{Encoding \CPROT@aac@encoding@first not defined\MessageBreak
              (see log file for more information)}
          }{}%
          \CPROT@packageinfo@or@warning{%
         You've requested char protruding for \CPROT@aac@encoding@first encoding\MessageBreak
          but \CPROT@aac@encoding@first encoding doesn't seem to be loaded.\MessageBreak
           Maybe you forgot a '\string\usepackage[\CPROT@aac@encoding@first]{fontenc}'.%
          }%
        \else
        \CPROT@aac@family{\protect\fontencoding{\CPROT@aac@encoding@first}\selectfont}%
        \fi
      }%
    }%
  }
% \changes{1.5}{2002/02/27}{added a some more protection for each font
% selection command (new \KOMAScript{} adds something to
% some of the font size selection commands)}
% \changes{1.7}{2004/06/28}{added parameter for using with encodings}
%    \begin{macrocode}
\newcommand{\CPROT@aac@family}[1]{%
  \ifthenelse{\value{CPROT@family@temp}>1}{%
    \protected@edef\CPROT@setprotcodes@temp{#1\protect\rmfamily}%
    \setcounter{CPROT@series@temp}{\value{CPROT@series}}%
    \CPROT@aac@series{\CPROT@setprotcodes@temp}%
    \addtocounter{CPROT@family@temp}{-2}%
    \CPROT@aac@family{#1}%
    }{%
    \ifthenelse{\value{CPROT@family@temp}>0}{%
      \protected@edef\CPROT@setprotcodes@temp{#1\protect\sffamily}%
      \setcounter{CPROT@series@temp}{\value{CPROT@series}}%
      \CPROT@aac@series{\CPROT@setprotcodes@temp}%
      \addtocounter{CPROT@family@temp}{-1}%
      \CPROT@aac@family{#1}%
    }{%
      \ifthenelse{\value{CPROT@family@temp}<0}{%
     \protected@edef\CPROT@setprotcodes@temp{#1\fontfamily{\CPROT@lowlevel@family}\selectfont}%
      \setcounter{CPROT@series@temp}{\value{CPROT@series}}%
      \CPROT@aac@series{\CPROT@setprotcodes@temp}%
    }{}%
      }%
    }%
  }
\newcommand{\CPROT@aac@series}[1]{%
  \ifthenelse{\value{CPROT@series@temp}>1}{%
    \protected@edef\CPROT@setprotcodes@temp{#1\protect\mdseries}%
    \setcounter{CPROT@shape@temp}{\value{CPROT@shape}}%
    \CPROT@aac@shape{\CPROT@setprotcodes@temp}%
    \addtocounter{CPROT@series@temp}{-2}%
    \CPROT@aac@series{#1}%
    }{%
    \ifthenelse{\value{CPROT@series@temp}>0}{%
      \protected@edef\CPROT@setprotcodes@temp{#1\protect\bfseries}%
      \setcounter{CPROT@shape@temp}{\value{CPROT@shape}}%
      \CPROT@aac@shape{\CPROT@setprotcodes@temp}%
      \addtocounter{CPROT@series@temp}{-1}%
      \CPROT@aac@series{#1}%
      }{%
```

```
      \ifthenelse{\value{CPROT@series@temp}<0}{%
        \protected@edef\CPROT@setprotcodes@temp{%
          #1\fontseries\expandafter{\CPROT@lowlevel@series}\selectfont}%
        \setcounter{CPROT@shape@temp}{\value{CPROT@shape}}%
        \CPROT@aac@shape{\CPROT@setprotcodes@temp}%
        }{}%
      }%
    }%
  }
\newcommand{\CPROT@aac@shape}[1]{%
  \ifthenelse{\value{CPROT@shape@temp}>7}{%
    \protected@edef\CPROT@setprotcodes@temp{#1\protect\upshape}%
    \setcounter{CPROT@size@temp}{\value{CPROT@size}}%
    \CPROT@aac@size{\CPROT@setprotcodes@temp}%
    \addtocounter{CPROT@shape@temp}{-8}%
    \CPROT@aac@shape{#1}%
    }{%
    \ifthenelse{\value{CPROT@shape@temp}>3}{%
      \protected@edef\CPROT@setprotcodes@temp{#1\protect\itshape}%
      \setcounter{CPROT@size@temp}{\value{CPROT@size}}%
      \CPROT@aac@size{\CPROT@setprotcodes@temp}%
      \addtocounter{CPROT@shape@temp}{-4}%
      \CPROT@aac@shape{#1}%
      }{%
      \ifthenelse{\value{CPROT@shape@temp}>1}{%
        \protected@edef\CPROT@setprotcodes@temp{#1\protect\slshape}%
        \setcounter{CPROT@size@temp}{\value{CPROT@size}}%
        \CPROT@aac@size{\CPROT@setprotcodes@temp}%
        \addtocounter{CPROT@shape@temp}{-2}%
        \CPROT@aac@shape{#1}%
        }{%
        \ifthenelse{\value{CPROT@shape@temp}>0}{%
          \protected@edef\CPROT@setprotcodes@temp{#1\protect\scshape}%
          \setcounter{CPROT@size@temp}{\value{CPROT@size}}%
          \CPROT@aac@size{\CPROT@setprotcodes@temp}%
          \addtocounter{CPROT@shape@temp}{-1}%
          \CPROT@aac@shape{#1}%
          }{%
          \ifthenelse{\value{CPROT@shape@temp}<0}{%
            \protected@edef\CPROT@setprotcodes@temp{%
              #1\fontshape\expandafter{\CPROT@lowlevel@shape}\selectfont}%
            \setcounter{CPROT@size@temp}{\value{CPROT@size}}%
            \CPROT@aac@size{\CPROT@setprotcodes@temp}%
            }{}%
          }%
        }%
      }%
    }%
  }
\newcommand{\CPROT@aac@size}[1]{%
  \ifthenelse{\value{CPROT@size@temp}>511}{%
    \protected@edef\CPROT@setprotcodes@temp{#1\protect\Huge}%
    \CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
    \addtocounter{CPROT@size@temp}{-512}%
    \CPROT@aac@size{#1}%
    }{%
    \ifthenelse{\value{CPROT@size@temp}>255}{%
      \protected@edef\CPROT@setprotcodes@temp{#1\protect\huge}%
      \CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
      \addtocounter{CPROT@size@temp}{-256}%
      \CPROT@aac@size{#1}%
      }{%
      \ifthenelse{\value{CPROT@size@temp}>127}{%
        \protected@edef\CPROT@setprotcodes@temp{#1\protect\LARGE}%
        \CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
        \addtocounter{CPROT@size@temp}{-128}%
```

```
            \CPROT@aac@size{#1}%
            }{%
            \ifthenelse{\value{CPROT@size@temp}>63}{%
              \protected@edef\CPROT@setprotcodes@temp{#1\protect\Large}%
              \CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
              \addtocounter{CPROT@size@temp}{-64}%
              \CPROT@aac@size{#1}%
              }{%
              \ifthenelse{\value{CPROT@size@temp}>31}{%
                \protected@edef\CPROT@setprotcodes@temp{#1\protect\large}%
                \CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
                \addtocounter{CPROT@size@temp}{-32}%
                \CPROT@aac@size{#1}%
                }{%
                \ifthenelse{\value{CPROT@size@temp}>15}{%
                  \protected@edef\CPROT@setprotcodes@temp{#1\protect\normalsize}%
                  \CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
                  \addtocounter{CPROT@size@temp}{-16}%
                  \CPROT@aac@size{#1}%
                  }{%
                  \ifthenelse{\value{CPROT@size@temp}>7}{%
                    \protected@edef\CPROT@setprotcodes@temp{#1\protect\small}%
                    \CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
                    \addtocounter{CPROT@size@temp}{-8}%
                    \CPROT@aac@size{#1}%
                    }{%
                    \ifthenelse{\value{CPROT@size@temp}>3}{%
                      \protected@edef\CPROT@setprotcodes@temp{#1\protect\footnotesize}%
                      \CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
                      \addtocounter{CPROT@size@temp}{-4}%
                      \CPROT@aac@size{#1}%
                      }{%
                      \ifthenelse{\value{CPROT@size@temp}>1}{%
                        \protected@edef\CPROT@setprotcodes@temp{#1\protect\scriptsize}%
                        \CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
                        \addtocounter{CPROT@size@temp}{-2}%
                        \CPROT@aac@size{#1}%
                        }{%
                        \ifthenelse{\value{CPROT@size@temp}>0}{%
                          \protected@edef\CPROT@setprotcodes@temp{#1\protect\protect\tiny}%
                          \CPROT@aac@textcomp{\CPROT@setprotcodes@temp}%
                          }{}%
                        }%
                      }%
                    }%
                  }%
                }%
              }%
            }%
          }%
        }%
      }
    \newcommand{\CPROT@aac@textcomp}[1]{%
      \ifthenelse{\boolean{CPROT@textcomp}}{%
        \@ifundefined{T@TS1}{%
          \CPROT@packageinfo@or@warning{%
            You've requested char protruding for TS1 encoding\MessageBreak
            but TS1 encoding doesn't seem to be loaded.\MessageBreak
            Maybe you forgot a '\string\usepackage{textcomp}'.%
            }%
          \setboolean{CPROT@textcomp}{false}%
          }{}%
        }{}%
      \ifthenelse{\boolean{CPROT@textcomp}}{%
        \protected@xdef\CPROT@setprotcodes@{%
          \CPROT@setprotcodes@{%
```

```
            #1\protect\CPROT@setprotcodes@font{%
              \fontencoding{TS1}\selectfont\protect\CPROT@setprotcodes@font%
              }%
            }%
          }%
        }{%
        \protected@xdef\CPROT@setprotcodes@{%
          \CPROT@setprotcodes@{%
            #1\protect\CPROT@setprotcodes@font%
            }%
          }%
        }%
      }
```

\activatecharprotruding

```
    \newcommand*{\activatecharprotruding}[1][true]{%
      \ifpdftex{%
        \ifthenelse{\equal{#1}{true}\or\equal{#1}{on}\or\equal{#1}{1}\or\equal{#1}{yes}}{%
          \CPROT@good@pdftex@version{%
            \global\pdfprotrudechars=2%
           }
          }{%
          \ifthenelse{\equal{#1}{compatibility}\or\equal{#1}{compatible}}{%
            \CPROT@good@pdftex@version{%
              \global\pdfprotrudechars=1%
              }%
            }{%
           \ifthenelse{\equal{#1}{false}\or\equal{#1}{off}\or\equal{#1}{0}\or\equal{#1}{no}}{%
              \global\pdfprotrudechars=0%
              }{%
              \CPROT@packageerror{%
                ``#1'' is no setting for \activatecharprotruding. Use\MessageBreak
                one of 'true', 'on', 'yes', '1', or 'false', 'off', \MessageBreak
                'no', '0', or 'compatible', 'compatibility' instead.%
                }%
              }{%
              \CPROT@packageinfo@or@warning{%
                You want to activate char protruding, but it does\MessageBreak
                look as your are NOT using pdftex. So I can't\MessageBreak
                activate it, as ``pdfcprot.sty'' only supports\MessageBreak
                pdftex.}%
              }%
            }%
          }%
        }{}%
      }
```

\setupcharprotruding

```
    \newcommand*{\setupcharprotruding}[1]{%
      \ifpdftex{%
        \CPROT@resetall{}%
        \@ifundefined{CPROT@save@KV@errx}{%
          \let\CPROT@save@KV@errx=\KV@errx%
          \renewcommand*{\KV@errx}[1]{%
            \CPROT@packageerror{option ##1 for \string\setupcharprotruding}{%
            You've tried to use the option ##1 with \string\setupcharprotruding.\MessageBreak
              But there is no option with that name.\MessageBreak
              See the manual of ``pdfcprot.sty'' for information
              about the usage of \string\setupcharprotruding.%
              }%
            }%
          \setkeys{CPROT}{#1}%
          \let\KV@errx=\CPROT@save@KV@errx%
          \let\CPROT@save@KV@errx=\relax%
          }{%
```

```
      \setkeys{CPROT}{#1}%
      }%
    \CPROT@setprotcodes@add{}%
    }{%
    \CPROT@packageinfo@or@warning{%
      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.}%
    }%
  }
\@onlypreamble\setupcharprotruding%
```

\setupcharprotrudingnumeral

```
\newcommand*{\setupcharprotrudingnumeral}[1]{%
  \ifpdftex{%
    \CPROT@resetall{}%
    \@ifundefined{CPROT@save@KV@errx}{%
      \let\CPROT@save@KV@errx=\KV@errx%
      \renewcommand*{\KV@errx}[1]{%
        \CPROT@packageerror{option ##1 for \string\setupcharprotrudingnumeral}{%
        You've tried to use the option ##1 with \string\setupcharprotrudingnumeral.\MessageBreak
          But there is no option with that name.\MessageBreak
          See the manual of ``pdfcprot.sty'' for information
          about the usage of \string\setupcharprotrudingnumeral.%
          }%
        }%
      \setkeys{CPROTnum}{#1}%
      \let\KV@errx=\CPROT@save@KV@errx%
      \let\CPROT@save@KV@errx=\relax%
      }{%
      \setkeys{CPROTnum}{#1}%
      }%
    \CPROT@setprotcodes@add{}%
    }{%
    \CPROT@packageinfo@or@warning{%
      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.}%
    }%
  }
\@onlypreamble\setupcharprotrudingnumeral
```

\ifpdftex

```
\newcommand{\ifpdftex}{%
  \ifx\pdftexversion\@undefined
    \expandafter\@secondoftwo
  \else
    \ifx\pdftexversion\relax
      \expandafter\expandafter\expandafter\@secondoftwo
    \else
      \expandafter\expandafter\expandafter\@firstoftwo
    \fi
  \fi
}
```

\CPROT@good@pdftex@version

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

```
      \CPROT@packageerror@bad@pdftex %
    \else
      \ifnum\pdftexversion = 14
        \ifnum \expandafter`\pdftexrevision < `f
          \CPROT@packageerror@bad@pdftex %
        \else
          #1
        \fi
      \else
        #1
      \fi
    \fi
  }
```

\ifpdfoutput This was taken from KOMA-Script and is provided for convenience.

```
  \newcommand*{\CPROT@ifpdfoutput}{%
    \ifcase 0%
            \ifx\pdfoutput\@undefined 1%
            \else
              \ifx\pdfoutput\relax    1%
              \else
                \ifcase\pdfoutput     1%
                \fi
              \fi
            \fi
            \space
      \expandafter\@firstoftwo
    \else
      \expandafter\@secondoftwo
    \fi
  }
  \@ifundefined{ifpdfoutput}{\let\ifpdfoutput\CPROT@ifpdfoutput}{%
    \CPROT@packageinfo{%
      \string\ifpdfoutput\space already defined.\MessageBreak
      That may happen when using KOMA-Script together\MessageBreak
      with ``pdfcprot.sty''. I have changed the definition\MessageBreak
      from Markus' KOMA-Script, so if you experience errors\MessageBreak
     try \string\let\string\ifpdfoutput\string\relax\space before loading ``pdfcprot.sty''.}%
    }

  \newcommand*{\CPROT@packageoptions}[1]{%
    \@ifundefined{CPROT@save@KV@errx}{%
      \let\CPROT@save@KV@errx=\KV@errx%
      \renewcommand*{\KV@errx}[1]{%
        \PackageError{pdfcprot}{unknown option ``#1''}{%
          You've tried to use the option ``#1''.\MessageBreak
          But there is no option with that name.\MessageBreak
          See the manual of ``pdfcprot.sty'' for information
          about known options.%
          }%
        }%
      \setkeys{CPROTpackage}{#1}%
      \let\KV@errx=\CPROT@save@KV@errx%
      \let\CPROT@save@KV@errx=\relax%
      }{%
      \setkeys{CPROTpackage}{#1}%
      }%
    }

  \DeclareOption{DVIoutput}{%
    \CPROT@packagewarning{%
      It is not a good idea to use the DVIoutput package
      option.\MessageBreak
      Use it only if you experience some using other
      packages.\MessageBreak
      And don't forget to file a bug report against these
      packages.\MessageBreak
```

```
      The best is to add just \string\pdfoutput=0\space at the beginning\MessageBreak
      of your preamble or even before \string\documentclass.
    }
    \ifpdftex{%
      \pdfoutput=0%
      \let\pdfoutput\undefined
      }{}%
    }
```

Define a standard command for undefined options. Actually this is just a redirection to a `\setkeys` command.

```
  \DeclareOption*{%
    \ifpdftex{%
      \expandafter\CPROT@packageoptions\expandafter{\CurrentOption}%
      }{}%
    }%
```

`\CPROI@fallbackfont` `\setfallbackfont` The internal command `\CPROT@fallbackfont` stores the font to be looked for, if no `CPA` is found for the actual font. By default it's Palatino. With `\setfallbackfont` it is possible to customise this, if required.

```
  \newcommand*{\CPROT@fallbackfont}{pplmn}
  \newcommand*{\setfallbackfont}[1]{\renewcommand*{\CPROT@fallbackfont}{#1}}
```

`T@setupcharprotruding@normal` When the package is loaded with option `activate` or `activate=normal` this command does the actual adjustment and by that defines which fonts will get character protruding with that options.

```
  \newcommand*{\CPROT@setupcharprotruding@normal}{%
    \setupcharprotrudingnumeral{family=3,series=1,shape=8,size=60,textcomp=1}%
    \setupcharprotrudingnumeral{family=2,series=2,shape=12,size=60,textcomp=1}%
    \setupcharprotrudingnumeral{family=1,series=2,shape=14,size=60,textcomp=1}%
    }
```

`tupcharprotrudingforencoding` This command

```
  \newcommand*{\setupcharprotrudingforencoding}[1]{%
    \setupcharprotrudingnumeral{encoding={#1},family=3,series=1,shape=8,size=60,textcomp=1}%
    \setupcharprotrudingnumeral{encoding={#1},family=2,series=2,shape=12,size=60,textcomp=1}%
    \setupcharprotrudingnumeral{encoding={#1},family=1,series=2,shape=14,size=60,textcomp=1}%
    }
  \@onlypreamble\setupcharprotrudingforencoding
```

`pdfcprot.cfg`

```
  \InputIfFileExists{pdfcprot.cfg}
            {\typeout{**************************************^^J%
                      * Local config file pdfcrpot.cfg used *^^J%
                      **************************************}}
            {}

  \ProcessOptions\relax
  \newcommand*{\CPROT@actualfont}{%
    \f@family%
    \f@series%
    \f@shape%
    \f@encoding%
    }
  \@onlypreamble\setfallbackfont%
  \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 correct font.

```
  \newcommand*{\CPROT@setprotcodes@font}{%
    \renewcommand*{\CPROT@cpa@toload}{\CPROT@actualfont.cpa}%
    \IfFileExists{\CPROT@cpa@toload}{%
```

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).

```
      \expandafter\@ifundefined\expandafter{\CPROT@actualfont}{%
        \input{\CPROT@cpa@toload}
        }{}
```

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.

```
      \csname\CPROT@actualfont\endcsname%
      \CPROT@adjustprotcodes{\font}%
      \CPROT@packageinfo{%
        File \CPROT@cpa@toload{} for the adjustment of\MessageBreak
        the char protruding used. It seems to be _fit_\MessageBreak
        to the font you requested.%
        }%
      }{%
      \CPROT@packageinfo@or@warning{%
        I didn't find \CPROT@cpa@toload.\MessageBreak%
        \space As a fallback I will try \CPROT@fallbackfont\f@encoding.cpa.%
        }%
      \renewcommand*{\CPROT@cpa@toload}{\CPROT@fallbackfont\f@encoding.cpa}%
      \IfFileExists{\CPROT@cpa@toload}{%
        \expandafter\@ifundefined\expandafter{\CPROT@fallbackfont\f@encoding}{%
          \input{\CPROT@cpa@toload}%
          }{}%
        \csname\CPROT@fallbackfont\f@encoding\endcsname%
        \CPROT@adjustprotcodes{\font}%
        \CPROT@packageinfo{%
          File \CPROT@cpa@toload{} used for the adjustment\MessageBreak
          of the char protruding. It seems that it isn't the one\MessageBreak
          you intended to use.%
          }%
        }{%
        \CPROT@packageinfo@or@warning{%
          Even the fallback font was not found. Maybe it's\MessageBreak
          not the right font encoding. Currently ``pdfcprot''\MessageBreak
          bundles only with cpa's for T1, OT1 and TS1 encoding.%
          }%
        }%
      }%
    }

  \newif\ifCPROT@adjustprotcodes@
  \newcommand*{\CPROT@adjustprotcodes}[1]{%
    \CPROT@adjustprotcodes@false
    \ifnum\pdftexversion > 14
      \CPROT@adjustprotcodes@true
    \else
      \ifnum\pdftexversion = 14
        \ifnum \expandafter`\pdftexrevision > `g
          \CPROT@adjustprotcodes@true
        \fi
      \fi
    \fi
    \ifCPROT@adjustprotcodes@
      \@tempcnta=0
      \loop
        \ifcase\lpcode#1\@tempcnta\else
          \CPROT@adjustprotcodes@@\lpcode{#1}\@tempcnta
        \fi
        \ifcase\rpcode#1\@tempcnta\else
          \CPROT@adjustprotcodes@@\rpcode{#1}\@tempcnta
        \fi
```

```
          \advance\@tempcnta 1
        \ifnum\@tempcnta < 256 \repeat
       \fi
     }
   \def\CPROT@adjustprotcodes@@#1#2#3{%
     \setbox0=\hbox{\the#2\char#3}%
   %  \setbox0=\hbox{%
   %    \ifx#2\font\else#2\fi%
   %    \char#3}%
     \@tempcntb=\wd0%
     \multiply\@tempcntb #1#2#3%
     \divide\@tempcntb \fontdimen6 #2%
     #1#2#3=\@tempcntb%
     }

   \newcommand*{\CPROT@setprotcodes}{%
     \ifpdftex{%
       \ifthenelse{\equal{\CPROT@setprotcodes@}{}}{%
         }{%
         \CPROT@setprotcodes@%
         }%
       }{}%
     }

   \AtBeginDocument{%
     \CPROT@setprotcodes\relax%
     }
```
⟨/package⟩

## A.2 The Provided .CPA Files

### A.2.1 Palatino

\pplmnT1
\pplmnOT1

```
⟨pplmnT1⟩\expandafter\gdef\csname pplmnT1\endcsname{%
⟨pplmnOT1⟩\expandafter\gdef\csname pplmnOT1\endcsname{%
⟨pplmnT2A⟩\expandafter\gdef\csname pplmnT2A\endcsname{%
⟨pplmnOT2⟩\expandafter\gdef\csname pplmnOT2\endcsname{%
⟨*pplmnOT1 | pplmnOT2⟩
  \lpcode\font 92=500  % ‘‘
  \rpcode\font 34=500  % ’’
  \rpcode\font 123=300 % --
  \rpcode\font 124=200 % ---
⟨/pplmnOT1 | pplmnOT2⟩
⟨*pplmnT1 | pplmnT2A⟩
  \lpcode\font 16=500  % ‘‘
  \rpcode\font 17=500  % ’’
  \rpcode\font 21=300 % --
  \rpcode\font 22=200 % ---
⟨/pplmnT1 | pplmnT2A⟩
⟨*pplmnT2A | pplmnOT2⟩
  \rpcode\font\cyrdash=200 % "---
  \lpcode\font\cyrdash=200 % "--*
⟨/pplmnT2A | pplmnOT2⟩
⟨*pplmnT1 | pplmnT2A⟩
% german quotation marks
  \lpcode\font\quotedblbase=600
  \rpcode\font\textquotedblleft=500
⟨/pplmnT1 | pplmnT2A⟩
⟨*pplmnOT1 | pplmnT1 | pplmnT2A | pplmnOT2⟩
  % set the protrusion of ","."-" and "." a bit smaller
  % than originally suggested by Han The Than
  \rpcode\font‘\,=650
  \rpcode\font‘\-=650
  \rpcode\font‘\.=650
  % originial Setting from Han The Thans protcode.tex
```

```
    \rpcode\font`\!=200
    \rpcode\font`\;=500
    \rpcode\font`\:=500
    \rpcode\font`\?=200
    \lpcode\font`\`=700
    \rpcode\font`\'=700
    \rpcode\font`\)=50
⟨*!pplmnOT2⟩
    \rpcode\font`\A=\rpcode\font`\A
⟨pplmnT1⟩  \rpcode\font 196=50 % A umlaut
    \rpcode\font`\F=50
    \rpcode\font`\K=50
    \rpcode\font`\L=50
    \rpcode\font`\T=50
    \rpcode\font`\V=50
    \rpcode\font`\W=50
    \rpcode\font`\X=50
    \rpcode\font`\Y=50
    \rpcode\font`\k=50
    \rpcode\font`\r=50
    \rpcode\font`\t=50
    \rpcode\font`\v=50
    \rpcode\font`\w=50
    \rpcode\font`\x=50
    \rpcode\font`\y=50
⟨/!pplmnOT2⟩
    \lpcode\font`\(=50
⟨*!pplmnOT2⟩
    \lpcode\font`\A=50
⟨pplmnT1⟩  \lpcode\font 196=\lpcode\font`\A % A umlaut
    \lpcode\font`\J=50
    \lpcode\font`\T=50
    \lpcode\font`\V=50
    \lpcode\font`\W=50
    \lpcode\font`\X=50
    \lpcode\font`\Y=50
    \lpcode\font`\v=50
    \lpcode\font`\w=50
    \lpcode\font`\x=50
    \lpcode\font`\y=50
⟨/!pplmnOT2⟩
⟨*pplmnT2A | pplmnOT2⟩
    \rpcode\font\CYRA=50
    \rpcode\font\CYRK=50
    \rpcode\font\CYRT=50
    \rpcode\font\CYRH=50
    \rpcode\font\CYRU=50
    \rpcode\font\CYRG=50
    \rpcode\font\CYRZH=50
    \rpcode\font\CYRL=50
    \rpcode\font\CYRC=20
    \rpcode\font\CYRSHCH=10
    \rpcode\font\CYRHRDSN=50
    \rpcode\font\CYRSFTSN=50
    \rpcode\font\cyrk=50
    \rpcode\font\cyrt=50
    \rpcode\font\cyrh=50
    \rpcode\font\cyru=50
```

```
    \rpcode\font\cyrg=50
    \rpcode\font\cyrzh=50
    \rpcode\font\cyrl=50
    \rpcode\font\cyrc=20
    \rpcode\font\cyrshch=10
    \rpcode\font\cyrhrdsn=50
    \rpcode\font\cyrsftsn=50
    \lpcode\font\CYRA=50
    \lpcode\font\CYRT=50
    \lpcode\font\CYRH=50
    \lpcode\font\CYRU=50
    \lpcode\font\CYRZH=50
    \lpcode\font\CYRL=50
    \lpcode\font\CYRU=50
    \lpcode\font\CYRCH=50
    \lpcode\font\CYRHRDSN=50
    \lpcode\font\CYRD=50
    \lpcode\font\cyra=50
    \lpcode\font\cyrt=50
    \lpcode\font\cyrh=50
    \lpcode\font\cyru=50
    \lpcode\font\cyrzh=50
    \lpcode\font\cyrl=50
    \lpcode\font\cyru=50
    \lpcode\font\cyrch=50
    \lpcode\font\cyrhrdsn=50
    \lpcode\font\cyrd=50
⟨/pplmnT2A | pplmnOT2⟩
}%
⟨/pplmnOT1 | pplmnT1 | pplmnT2A | pplmnOT2⟩
```

\pplmnTS1

```
⟨*pplmnTS1⟩
\expandafter\gdef\csname pplmnTS1\endcsname{%
  \rpcode\font 176=600 % \textdegree
}
⟨/pplmnTS1⟩
```