

interfaces

interfaces with keys for other packages

FC

2010/12/27 – version 2.6 – release – pgf v2.1 / pgf v2.0 compatible

Abstract

`interfaces` provides a small number of user friendly macros to access features implemented in frequently used packages. Most of the macros are based on `pgfkeys`, hence the “key=value” syntax.

`interfaces` is mainly based on `pgfkeys`¹ and `scrfile` for its different modules are loaded \AfterPackage. ε-TEx is required of course... Each interface is actually made as a separate module, which is loaded automatically after its *master package*. Adding `\usepackage{interfaces}` into your document preamble will make all modules active (*ie.* they will be loaded or not, depending on your document preamble). If you prefer not to use all modules, you can load them separately as a package, for example, type: `\usepackage[interfaces-titlesec]{interfaces}` *after* `\usepackage{titlesec}`. A better way is to add `\usepackage[interfaces-base]{interfaces}` to your preamble, and then: `\LoadInterface{titlesec,tocloft}` indifferently before or after `\usepackage{titlesec,tocloft}`. Another possibility to restrict the modules is to give them as options for the main package: `\usepackage[titlesec,tocloft]{interfaces}`.

`interfaces-base` is a mandatory “module”. It loads `interfaces-scrfile`, `interfaces-etoolbox`² and `interfaces-pgfkeys`: those are the *system-layer* of `interfaces`.

Example

```
\pagestyle{document}%
  rules=.5pt,
  foot/left/odd=\arabic{page},
  foot/right/even=\arabic{page}/\pageref{LastPage},
  head/font=\color[gray]{.85}\bfseries\small,
  head/left/even=\leftmark,
  head/left=\thefirstmarks{MiniSection},% interfaces-marks
}
\pagestyle{plain}%
  % for chapters...
  pagestyle=document,% preloaded settings
  head=,
  foot/left/odd = \roman{page},
  norules, head/rule=.4pt
  head/center=This is the title,
}
\sectionformat{section}{hang}%
  label=\arabic{section},
  labelsep=,
  font=\bfseries\Large,
  bookmark={bold,color=blue},% interfaces-bookmark
  top=6pt plus3pt minus2pt,
  bottom=3pt plus2pt,
  break=\marksthecs{section}{lastsectiontoc},% interfaces-hypbmsec
  mark=\marksthe{section}{#1},% interfaces-marks
}
\sectionformat{subsection}{hang}%
  label=, labelsep=,
  font=\bfseries,
  bookmark=italic,
  top*=1.2,% a 120% scaling
  bottom+=0pt plus4pt,% add to previous value
  mark=\markright{#1},
}
\sectionformat{subsubsection}{hang}%
  class=straight[\subparagraph],
  shape=block,
  label=,
  top=3pt plus2pt, bottom=1pt,
  font=\bfseries\sffamily,
  bookmark=off,% or false or none
  break=\allowbreak,
  mark=\marksthe{MiniSection}{#1},% see interfaces-marks
}
\bottomtitles=.2\textheight
\tocsetup{
  depth=3, title=Contents, pagenumbers=on,
  pagestyle=plain,% first page style: tocloftpagestyle
  title/top=1.5ex plus.5ex,
  title/bottom=0pt,
  section/indent=1em,
  section/number/width=1.5em,
  section/font+=\larger\bfseries,% append to previous setting
  section/leaders,section/dotsep,
  subsection/indent=2em,
  subsection/number/width=2.5em,
  subsubsection/font=\scriptsize,
  subsection/leaders,subsection/dotsep,% default values
  subsubsection/pagenumbers=off,% no page numbers
  subsubsection/dotsep=% no leader
  columns=2,% same as twocolumns
  columns/sep=.5em,
  columns/rule/width=1pt,columns/rule/color=blue,
  label=toc,% gives a label to the t.o.c.
  bookmark={text=Contents,bold},% interfaces-bookmark
}
\tableofcontents
\mainmatter \pagestyle{document}
```

This documentation is produced with the DocStrip utility.

- To get the package, run: `etex interfaces.dtx`
- To get the documentation (run thrice): `pdflatex interfaces.dtx`
- To get the index, run: `makeindex -s gind.ist interfaces.idx`

The .dtx file is embedded into this pdf file thank to embedfile by H. Oberdiek.

1. `pgfkeys`: [CTAN:graphics/pgf/base/generic/pgf/utilities](#)

2. All macros in `interfaces-etoolbox` are defined with `\providecommand` except `\removefromstart` which is used by `\repeatcell`.

Contents

1 Interface L^AT_EX	3	21 Implementation	26
2 Interface TikZ	3	21.1 Interfaces	26
3 Interface titlesec: \sectionformat	4	21.1.1 Requirements	
4 Interface hyperref	6	21.1.2 The package options / list of interfaces	
5 Interface bookmark	6	21.2 Interfaces-base	26
6 Interface hypbmsec	7	21.2.1 Interfaces patches	
7 Interface tocloft: \tocsetup	8	21.2.2 Helper macros	
8 Interface fancyhdr: \pagesetup	10	21.2.3 Numbers	
9 Interface marks	11	21.2.4 Units and math units	
10 Interface makecell: \repeatcell	12	21.2.5 Default color	
10.1 Conversion to scaled points	13	21.3 Interface L^AT_EX	31
10.2 Variable length cell	13	21.3.1 \changefont	
10.3 Empty rows	14	21.4 Interface TikZ	32
10.4 color palette	14	21.4.1 Catcode verifications	
10.5 tabularcalc-like tables (with dcolumn package) and pgfmath	14	21.4.2 List of pgf/TikZ packages and libraries	
10.6 Double entry array: the Body Mass Index	15	21.4.3 pgfkeys definitions for \usetikz	
11 Interface enumitem	16	21.5 Interface titlesec: \sectionformat	35
12 Interface truncate	17	21.5.1 titlesec modifications	
13 Interface appendix	18	21.5.2 \sectionformat	
14 Interface embedfile: \embedthisfile	18	21.5.3 Keys for \sectionformat	
15 Interface environ	18	21.5.4 \bottomtitles	
16 Interface umrand	18	21.6 Interface hyperref	41
17 Interface pgfkeys	19	21.7 Interface bookmark	42
17.1 /.blank	19	21.7.1 star keys* definitions	
17.2 /.is private	20	21.7.2 ref and pageref keys	
17.3 /.store in cs, /.estore in cs	20	21.8 Interface hypbmsec	45
17.4 Boolean keys: /.true, /.false, /.on /.off	20	21.8.1 interfaces-hypbmsec	
17.5 /.is multiple choice	20	21.9 Interface fancyhdr: \pagesetup	47
17.6 /.is enum	20	21.9.1 \pagesetup	
17.7 /.store as code	20	21.9.2 Keys for \pagesetup	
17.8 /.store as code in cs	21	21.10 marks	61
17.9 /.store as color	21	21.11 Interface tocloft: \tocsetup	62
17.10 /.store as color in cs	21	21.12 Interface makecell: \repeatcell	68
17.11 /.store as number	22	21.12.1 countvalues	
17.12 /.store as number in cs	22	21.12.2 \repeatcell	
17.13 /.store as glue	22	21.12.3 colortbl: patch for \CT@cellc	
17.14 /.store as dimen	22	21.13 Interface enumitem	74
17.15 /.store as glue in cs	22	21.14 Interface truncate	75
17.16 /.store as dimen in cs	22	21.15 Interface appendix	77
17.17 /.estore as dimen in cs	22	21.16 Interface embedfile: \embedthisfile	77
18 Interface etoolbox	22	21.17 Interface environ	78
18.1 New macros provided with \providetcommand	23	21.18 Interface umrand	78
19 Interface scrlfile	23	21.19 Interface scrlfile	78
19.1 Star forms for \BeforeFile and \AfterFile	23	21.20 Interface pgfkeys	79
19.2 Allowing \BeforeFile and \AfterFile when \@input is invoked	23	21.21 Interface etoolbox	91
20 List of corrected or patched commands	23	22 History	93
20.1 LaTeX	24	[2010/12/27 v2.6]	93
20.2 bookmark	24	[2010/12/07 v2.5]	93
20.3 tocloft	24	[2010/11/26 v2.4]	93
20.4 colortbl	24	[2010/11/21 v2.3]	93
20.5 hypbmsec	25	[2010/11/16 v2.2]	94
20.6 enumitem	25	[2010/11/15 v2.1]	94
20.7 pgfkeys	25	[2010/10/31 v2.0]	94
	24	[2010/10/27 v1.2]	94
	24	[2010/10/20 v1.0]	94
24 README	24	23 Index	95

1 Interface LATEX

```
\changefont{ encoding |enc = changes \fontencoding
            family |fam = changes \fontfamily
            series |ser = changes \fontseries
            shape |sha = changes \fontshape
            size |size* |size+ |size- = changes the font size (default unit is pt)
            baselineskip |baselineskip* + - |skip = changes the \baselineskip for the font
            linespread |linespread* + - |spread = changes the line spread (a real value) }
```

Example: adds 5pt to the font size (pt is added if not specified) and scale the linespread by 130%:

```
\changefont{ size+ = 5, spread* = 1.3} (\selectfont is included in the macro).
```

If a key is not specified, the current value remains.

Finally, `\changefont` is compatible with `hyperref` thank to `\pdfstringdefDisableCommands`.

2 Interface TikZ ([pgfmanual](#))

```
\usetikz { basic =..., loads tikz with libraries calc, arrows, scopes and backgrounds
           all =..., loads tikz with all packages, pgf-modules and TikZ-libraries
           any other key =..., loads tikz and the specified package, module or library
                         using \usepackage, \usepgfmodule or \usetikzlibrary }
```

`\usetikz` is a facility provided to load pgf, tikz, pgf-modules, and tikz-libraries, without the need to know if you must use `\usepackage` or `\usepgfmodule` or `\usetikzlibrary`.

Keys `shorten>` and `shorten<` have been set as aliases to `shorten >` and `shorten <` (original with space) and keys `shorten<>` and `shorten <>` have been set to style `shorten >` and `shorten <`.

`\usetikz` does not parse options. If you are used to load pgf with option (f.ex. `version=0.96`), please use the syntax:

```
\PassOptionsToPackage{version=0.96}{pgf}
```

MEMO

TikZ/pgf package	pgfmodules	TikZ libraries	
pgf	matrix	arrows	decorations.pathreplacing
pgfmath	plot	chains	decorations.shapes
pgfcalendar		er	decorations.text
pgfplots		scopes	decorations.markings
pgfpages		snakes	decorations.fractals
tikz		shapes	decorations.footprints
		shapes.misc	positioning,trees
		shapes.geometric	automata
		shapes.arrows	patterns
		shapes.multipart	fadings
		shapes.callouts	shadows
		shapes.symbols	through
		shapes.gates.logic.US	backgrounds
		shapes.gates.logic.IEC	petri
		decorations	fit
		decorations.pathmorphing	mindmap
			folding
			plothandlers
			plotmarks
			calc
			matrix

sorry for the blank cell, but I didn't get out with `colortbl` and the `\multicolumn` entry...

3 Interface ☞ titlesec : \sectionformat

```
\sectionformat{\section} { class=..., page|top|straight [\"sup-level]
                     shape=..., hang|block|display|runin|leftmargin|rightmargin|
                     drop|wrap|frame|runinifempty
                     font|format=..., label=..., labelsep|sep=..., 
                     left=..., right=..., top=..., bottom=..., 
                     before=..., after=..., 
                     mark=..., break=..., thesection=..., }
```

`\sectionformat` is a `pgfkeys`-command for replacement of both `\titleformat` and `\titlespacing`. If a key is not given, its value remains the same while with `\titleformat` or `\titlespacing`, you must specify all values in a rigid order... Moreover, `\sectionformat` can be used concurrently with `\titleformat` and `\titlespacing` (although `\sectionformat` is deemed more comfortable, this allows to quickly modify some files written without `interfaces`.)

Keys corresponding to T_EX code (`font`, `before` or `after`) are associated with starred keys and `+` keys: they append their content to the previous value. Ex: `font+={\larger}` adds `\larger` to the previous settings.

For dimensions or skips, the `+` key adds, the `-` key subtract and the `*` key scales.

~~~~~ MEMO ~~~~

<code>class</code>	name of the class	<code>page</code> =like standard <code>\part</code> <code>top</code> =like standard <code>\chapter</code> <code>straight</code> =like standard <code>\section</code> etc. example: <code>\sectionformat\subchapter{class=straight[\chapter]}</code>
<code>shape</code>	shape	<code>hang</code> =hanging label (like <code>\section</code>) <code>block</code> =paragraph without formatting (for centered titles) <code>display</code> =like standard <code>\chapter</code> <code>runin</code> =like standard <code>\paragraph</code> <code>leftmargin</code> =left margin <code>rightmargin</code> =right margin <code>drop</code> =wraps the text around the title (the first paragraph should be longer than the title, otherwise, they overlap) <code>wrap</code> =like <code>drop</code> but the space reserved for the title is variable <code>frame</code> =like <code>display</code> +puts the title in frame <code>runinifempty</code> = <code>hang</code> but no label if the title is empty
<code>font=format</code> <code>label</code> <code>sep=labelsep</code>	format for title title label space between label and title	applied to the label and the title ex: <code>\arabic{section}</code>
<code>left</code> <code>right</code> <code>top</code> <code>bottom</code>	left margin right margin vertical space before vertical space after	a glue: horizontal space before a glue: horizontal space after a glue: vertical space before a glue: vertical space after
<code>mark</code> <code>break</code> <code>thesection</code>	macro with 1 parameter parameterless macro toc and ref labels	ex <code>\sectionmark</code> : redefines <code>\...mark#1...</code> (the argument is the title) ex <code>\sectionbreak</code> to be executed just before the section command redefines <code>\thesection</code> <code>\thesection</code> , <code>\thesubsection</code> etc.. are used to print references and the label in the table of contents
<code>before</code> <code>after</code>	code before title body code after title body	

If the `bookmark` package is loaded, with `interfaces-bookmark`, the following keys are defined for each section:

<code>bookmark/bold</code>	bold bookmark text	the text of the bookmark for that sectionning command will be bold
<code>bookmark/italic</code>	italic bookmark text	the text of the bookmark for that sectionning command will be in <i>italic</i>
<code>bookmark/color</code>	colored bookmark text	the text of the bookmark for that sectionning command will be colorized
<code>bookmark/off</code>	disable the bookmark	no bookmark for that sectionning command alternatively you may set <code>bookmark/level</code> to a huge value...
<code>bookmark/style</code>	use a <i>style</i>	the <i>style</i> must be previously declared with <code>\bookmarkdefinestyle</code>
<code>bookmark/level</code>	level of the bookmark	defines <code>\toclevel@<sectionname></code> , see <code>bookmark</code> documentation

The syntax: `\sectionformat\subsection{bookmark={bold,italic,color=green,on,view=FitB}}` also works.

→ Key `bookmark` is an example of a key defined with the handler `/.is multiple choice`.

The starred keys `bold*`, `italic*`, `color*` and `off*` are also provided. For example:

```
\sectionformat\section{  
    after=\sectionformat\subsection{bookmark/color*=red}  
}
```

makes the bookmarks corresponding to the first subsection in each section to be red. If you ascertain that there is no bookmark between the one corresponding to the section and the one corresponding to its first subsection, then this is the same as: “`after=\bookmarksetup{color*=red}`”.

EXAMPLE

<pre>\sectionformat\section[hang]{% font=\bfseries\scshape, label=\Roman{chapter}\ \textbullet\ \arabic{section}, labelsep=\widthof{\textbullet}, bottom=1ex plus.4ex minus.2ex, top+=3pt, mark=\markright{\#1},% => this defines \def\sectionmark{\markright{\#1}} break=\goodbreak, % => this defines \def\sectionbreak{\goodbreak} thesection=\Roman{chapter}.\arabic{section}, % redefines \def\thesection{...} bookmark={bold,color=blue}}</pre>	this is the same as: <code>shape = hang</code>
---	--

`titlesec` provides *extended settings* for numberless (*i.e.* starred) sections with the syntax:

```
\titleformat{name=\section,numberless}{...}
```

One swallow does not a summer make, `interfaces-titlesec` does not provide this feature in a key=value stuff but just type:

```
\sectionformat\section*{[ key=value ]}
```

to give to your `\section*` command a special format. Keys `label` and `labelsep` are not ignored, this means that `\section*` can be defined in a way it produces a numbered section! Albeit, this would be weird because the `\section` counter is not updated for starred sections...

Presently, `\sectionformat` does not implement the *extended settings* of `titlesec` for odd and even pages.

However, it is always possible to use `\sectionformat` concurrently with `\titleformat` and `\titlespacing`. A better way to use different settings for odd and even pages is the use of `\ifthispageodd` from the KOMA-Script classes, or the features provided by `changepage.sty`, which is almost the same: actually `titlesec` test oddity of the `page` counter (*i.e.* TeX `\count` register nr 0) while oddity of `\pageref` should be tested... Hence the decision I came not to implement different settings for odd and even pages in `\sectionformat`.

`\bottomtitles{<glue|percentage of \textheight>}`

This globally sets the value of `\bottomtitlespace` (see `titlesec` documentation):

<code>\bottomtitles{.2}</code>	sets <code>\bottomtitlespace</code> to <code>.2\textheight</code>
<code>\bottomtitles=.2\textheight</code>	is the same

You can globally set a negativ value: `\bottomtitles = -1` to allow bottom titles.

4 Interface ⇔ `hyperref`

```
\raisedhyperdef[⟨raised dimen⟩]{⟨category⟩}{⟨name⟩}{⟨text⟩}
```

Package `hyperref` provides `\hyperdef` and `\Hy@raisedlink` but no easy way to insert a “raised link” with `\hyperdef` (or `\hypertarget`).

`\raisedhyperdef` can be used exactly like `\hyperdef`, but the pdf-anchor will be raised of the specified dimension.

5 Interface ⇔ `bookmark`

A few new keys appear with `interfaces` for the command `\bookmarksetup`:

```
\bookmarksetup { bold*=..., will make the bookmark text bold, but only for the very next bookmark
                italic*=..., idem for italic
                color*=..., idem for color: the previous color will be restored after the next bookmark
                disable=..., will disable the bookmarks (the same as \bookmarksetup{depth=-2})
                disable*=..., will disable only the very next bookmark }
```

Furthermore the following code:

```
\sectionformat\section{bookmark={bold,color=blue}}
\bookmarksetup{color*=red}
\section{A lovely red, bold bookmark here!}
\section{And the following bookmark is blue and bold}
```

will produce a bookmark with attributes: `color=red, bold=true`. In other words, the starred keys of `\bookmarksetup` overwrite the “default” settings of `\sectionformat` for the very next bookmark.

Note that this code requires `titlesec`, `bookmark`, `interfaces-titlesec` and `interfaces-bookmark`.

Three new keys are added to the `\bookmark` command:

```
\bookmark{ pageref = the destination of the bookmark is the page referenced
           ref = the destination of the bookmark is the anchor associated to the reference
           atend = the bookmark will be inserted at the end of the document }
```

I admit those keys will not be used very often... but imagine you want to put a bookmark at the end, which points to the table of contents, then:

```
\bookmark[ref=TableOfContents,atend]{Back To Contents}
```

Provided that the label `TableOfContents` has been defined, for example with `\tocsetup`, if `tocloft` and `interfaces-tocloft` are loaded:

```
\tocsetup{label=TableOfContents} \tableofcontents
```

The `atend` key requires of course that the package `bookmark` was loaded with the `[atend]` option (otherwise, `bookmark.sty` gives a warning).

6 Interface `\hypbmsec`

The features provided by `interfaces-hypbmsec` are now independant of `titlesec` and `interfaces-titlesec`.

Package `hypbmsec` provides extended sectionning commands with the syntax:

```
\section{out}[toc]{title}      or      \section[toc](out){title}
```

(`out`) is used for the bookmark (outline)

Where: `[toc]` for the table and contents and the marks

`{title}` to typeset the section title in the document

Internally, `hypbmsec` defines `\hbs@bmstring` (the string used for the bookmark entry) and `\hbs@tocstring` (the string used for the table of contents and the marks).

`interfaces` takes advantage of this and defines:

<code>\lastsectiontitle</code>	expands to the title of the last <code>\section</code>
<code>\lastsectionbookmark</code>	expands to the text of the bookmark for the last <code>\section</code>
<code>\lastsectiontoc</code>	expands to the entry of the last <code>\section</code> in the table of contents
<code>\lastsectionanchor</code>	expands to the anchor name of the last <code>\section</code> if <code>hyperref</code> is used

Macros for each sectionning command are defined:

<code>\lastparttitle</code>	<code>\lastchaptertitle</code>	
<code>\lastsectiontitle</code>	<code>\lastsubsectiontitle</code>	<code>\lastsubsubsectiontitle</code>
<code>\lastparagraphtitle</code>	<code>\last subparagraphtitle</code>	

with their `...bookmark` and `...toc` counterparts.

Assignment to those macros is automatic. More precisely, `\lastsectiontitle` is defined **after** `\sectionbreak` and **before** `\sectionmark`: this means that in the fields `after` or `mark` of `\sectionformat`, `\lastsectiontitle` has been updated yet. While in the fields `before`, and `break`, `\lastsectiontitle` is not yet updated. (For chapters, this is the same if `titlesec` formats the chapter titles, *i.e.* if `\sectionformat\chapter` or `\titleformat\chapter` has been invoked. Otherwise, `\lastchaptertitle`, `\lastchapterbookmark` and `\lastchaptertos` are set at `\stepcounter{chapter}`).

Values are automatically reset to `\relax` when a section of a superior level is created.

This feature has been implemented essentially to make the use of a `\mark` command at `\sectionbreak` easier. Mark at `\sectionbreak` is important to get the correct behaviour for marks described in `interfaces-marks` (page 11 under `\ifmarksequal`). With this feature, when a new `\section` is created, `\lastsubsection` is reset to `\relax` therefore, `\marksthecs{\subsection}{\lastsubsection}` will not mark anything.

Another possibility is shown in this document: the header is clickable and has been defined with:

```
\sectionformat\section{
    mark=\marksthe{section}{#1}
    \marksthecs{sectionanchor}{lastsectionanchor}
}
\pagesetup{
    head/right=\hyperref{}{}\{\thefirstmarks{sectionanchor}\}
                {\thefirstmarks{section}}
}
```

Setting the keys overwrites the previous value, unless the `+` key (append code/add for dimensions), the `-` key (subtract) or the starred-key`*` (scale dimensions) is used. `\tocsetup` settings apply to `\tableofcontents` and also to `\shorttableofcontents` (package `shorttoc`). `\tocsetup` can be invoked more than once in a document...

```
\tocsetup{ depth = \setcounter{tocdepth} This is a LATEX rather than tocloft's ...
           With interfaces you can change the t.o.c. depth anywhere inside your
           document, just like bookmarks depths (two runs are required)
           title|contentsname = redefines \contentsname
           title/top|title/top+|title/top-|title/top* = vertical space before the title (set|add|subtract|scale)
           title/bottom|title/bottom+ etc. = vertical space after the title
           title/font|title/font+|title/format = the title font (set|append code)
           title/color|title/font/color = the color of the title (may also be defined in the key title/font)
           title/after|title/after+ = hook to execute just after the title (\cftaftertoctitle...)
           parskip|parskip+|parskip-|parskip* |parsep = modifies (or add to) \cftparskip
           dotsep = modifies \cftdotsep if no dimen is specified, mu is assumed, otherwise, conver-
             sion to  $\mu \approx 1em/18$  is supplied. Default value is  $4.5\mu$  and blank value is the same as
             \cftnodots
           dot = modifies \cftdot, the dot used for leaders produced by
             \cftdotfill
           pagestyle = page style for the first page (\tocloftpagestyle)
           mark = the mark command: \cftmarktoc
             example: \chaptermark\contentsname
           before|before*|after|after* = hooks to execute before or after the command \tableofcontents
           twocolumns|columns|multicol = twocolumns=true is the same as columns=2 Makes a multicolumn
             table of contents. Package multicol is required!
           columns/sep|columns/sep+|columnsep etc. = sets (or add) a \dimen to \columnsep
           columns/rule|columns/rule+|columnseprule etc. = sets (or add) a \dimen to \columnseprule
           columns/color|columnseprulecolor = modifies \columnseprulecolor
           multicolsep|multicolsep+ etc. = modifies (or add) a \glue to \multicolsep
             \multicolsep is the space added before and after multicol
             environment. Its default value is 0pt plus 4pt minus 3pt.
           multicols/begin|twocolumns/before = hooks just after \begin{multicols} ...
           multicols/end|twocolumns/after = ... and before \end{multicols}
           label = the label for reference to the table of contents
           ref = the reference printed by \ref when referencing the table of contents
             (default is \contentsname. In any case, \nameref prints \contentsname).
           bookmark/text = the bookmark text for the table of contents (if empty, no bookmark is set)
           bookmark/bold|bookmark/italic|bookmark/color = the bookmark style for the table of contents
           bookmark/level = the bookmark level: either a number or section or chapter
             Default is chapter, or section if the class does not provide chapters. }
```

`bookmark` key and sub-keys are available only if the package `bookmark` is loaded with `interfaces-bookmark`.

The following keys apply to each section levels. Section level names are : `part` / `chap` or `chapter` / `sec` or `section` / `subsec` or `subsection` / `subsubsec` or `subsubsection` / `para` or `paragraph` / `subpara` or `subparagraph` / `fig` or `figure` / `subfig` or `subfigure` / `tab` or `table` / `subtab` or `subtable`. ¶ may be one of those names (see `tocloft` documentation).

```
\tocsetup{ ¶/top|¶/top+|¶/skip|¶/beforeskip = modifies (or add to) the vertical \glue before an entry
           ¶/indent|¶/indent+|¶/indent-|¶/indent* = modifies (or add etc.) the indentation of the entry
           ¶/font|¶/font+ = the font of the ¶ section title
           ¶/color|¶/font/color = the color of the ¶ section title (overwrites linkcolor if hyper-linked)
           ¶/number/width|number/width+ etc. |numwidth = modifies the space allowed to typeset the title numbers
           ¶/number/before|number/before+|presnum = code before the section number (see pseudo code below)
           ¶/number/after|number/after+|aftersnum = code after the section number (see pseudo code below)
           ¶/title/before|title/before+|aftersnum = code before the title (ie. after the section number box)
           ¶/leader|¶/leaders = modifies the leader: the default is \cftdotfill{\cftdotsep}
           ¶/dotsep = the space (in mu) between dots in the leader. Default is \cftdotsep
             and blank value is the same as \cftnodots
           ¶/pagefont|¶/page/font|¶/page/number/font = modifies the font used to typeset the page number
           ¶/page/number/color|¶/page/number/font/color = the color of the page numbers for this section (overwrites linkcolor
             if the page number if hyper-linked)
           ¶/page/after|¶/pagenumber/after|afterpnum = macro expanded after the page number is typeset }
```

~~~~~ MEMO ~~~~~

---

numbered entries are typeset with this pseudo code:

```
{\cftXfont {\cftXpresnum SNUM\cftXaftersnum\hfil} \cftXaftersnumb TITLE}%
{\cftXleader}{\cftXpagefont PAGE}\cftXafterpnum\par
```

and for not numbered entries:

```
{\cftXfont TITLE}{\cftXleader}{\cftXpagefont PAGE}\cftXafterpnum\par
```

---

\lofsetup and \lotsetup behave the same.

\listofsetup{<list-name>}{<key=value>}

This command is provided to setup the parameters of user-defined lists created with tocloft's \newlistof and printed with \listofX (where X is the name given to the list). The keys are the same as ~\tocsetup, and the sub-keys correspond to the entries. In fact:

- \tocsetup is a shortcut for \listofsetup{toc}
- \lofsetup is a shortcut for \listofsetup{lof} and
- \lotsetup is a shortcut for \listofsetup{lot}

Example:

```
\newlistof[section]{answer}{ans}{List of Answers}
\newlistentry[answer]{subanswer}{ans}{1} } preferably in
\listofsetup{ans}{ } the preamble
    depth=2, % can be changed inside the document body
    title=List of Answers of the section,% modifies the title
    title/font=\Huge,
    answer/font=\Large\bfseries, % font spec.
    subanswer/font=\large\itshape,
    answer/dotsep=3mu,
    subanswer/dotsep=, % no dot
    subanswer/pagenumbers=off,
}
\addcontentsline{ans}{answer}{\numberline{1}First answer}
\addcontentsline{ans}{subanswer}{\numberline{1.1}First sub-answer}
\listofanswer
```

## 8 Interface fancyhdr : \pagestyle{}

|                                                  |                            |                                                                                         |
|--------------------------------------------------|----------------------------|-----------------------------------------------------------------------------------------|
| <code>\pagestyle[&lt;pagestyle-name&gt;]{</code> |                            |                                                                                         |
| norule norules                                   |                            | set <code>\headrulewidth</code> and <code>\footrulewidth</code> to 0pt                  |
| rule rule/width[* + -]=...,                      | rule/font=...,             | set <code>\headrulewidth</code> and <code>\footrulewidth</code> to a given value        |
| rule/color=...,                                  | footrule=...,              | To change the rule font/color                                                           |
| headrule=...,                                    |                            | redefines <code>\headrule</code> and <code>\footrule</code> (no value restores default) |
| head/rule/width=...,                             | foot/rule/width=...,       |                                                                                         |
| head/rule/color=...,                             | foot/rule/color=...,       | for head and foot rules separately                                                      |
| head/rule/font=...,                              | foot/rule/font=...,        | <i>idem</i>                                                                             |
| head=...,                                        | foot=...,                  | header and footer ( <code>\fancyhead</code> / <code>\fancyfoot</code> )                 |
| left=...,                                        | center=...,                |                                                                                         |
| head/left=...,                                   | head/center=...,           | right=..., by position                                                                  |
| foot/left=...,                                   | foot/center=...,           | head/right=..., for one side documents                                                  |
| head/left/even=...,                              | head/left/odd=...,         | foot/right=..., <i>idem</i>                                                             |
| etc.                                             | etc.                       | for two side documents                                                                  |
| foot/right/even=...,                             | foot/right/odd=...,        |                                                                                         |
| inner=...,                                       | head/inner=...,            | foot/inner=..., inner is an alias for right/even, left/odd                              |
| outer=...,                                       | head/outer=...,            | foot/outer=..., outer is an alias for left/even, right/odd                              |
| offset=...,                                      | foot/offset=...,           | changes <code>\fancyhfoffset</code>                                                     |
| head/offset=...,                                 | right/offset=...,          |                                                                                         |
| left/offset=...,                                 |                            | for two side documents                                                                  |
| head/left/even/offset=...,                       | head/left/odd/offset=...,  |                                                                                         |
| etc.                                             | etc.                       | for two side documents                                                                  |
| foot/right/even/offset=...,                      | foot/right/odd/offset=..., |                                                                                         |
| inner/offset=...,                                | head/inner/offset=...,     | inner is an alias for right/even, left/odd                                              |
| outer/offset=...,                                | foot/outer/offset=...,     | outer is an alias for left/even, right/odd                                              |
| pagestyle=...,                                   |                            | load the settings of another page style                                                 |
|                                                  |                            | }                                                                                       |

`\pagestyle*` [<pagestyle-name>] {key = value}

`\pagestyle` resets all headers and footers, rules etc. and modifies the current (active) page style depending on the key=value pairs. `\pagestyle[<page-style>]` defines (or redefines) a page style, starting from default values. When a key is specified with no value, the default is used (f. ex. `head/rule` will be `.4pt` and `foot/rule` will be `0pt`).

`\pagestyle*` does not reset current values: keys that are not specified remains the same.

`\pagestyle*` [<page-style>] defines or redefines <page-style>; the normal form starts from scratch, while the starred form `\pagestyle*` keeps previous values if they are not overwritten, and if the <page-style> exists, of course.

`foot/left/odd` is the same as `odd/foot/left` etc. but `twoside` option is not checked... Even pages are ignored in `oneside` documents: this is `LATeX`, not `fancyhdr` nor `interfaces-fancyhdr`.

As usual, for dimensions or glues, `+`, `-` and `*` keys are provided for addition, subtraction and scaling . And for code (font keys) the `+` and `*` keys are provided: both append their content to the previous value.

## 9 Interface marks

`interfaces-marks` provides a few macros to make easier the use of  $\varepsilon$ -TeX \marks registers.  $\varepsilon$ -TeX defines 32 768 marks registers, making the use of them far more comfortable than “old” L<sup>A</sup>T<sub>E</sub>X \markboth, \markright, \leftmark and \rightmark. Who gets the correct – I mean *expected* – result with such commands ? Not me, I can say !

```
\marksthe{\{named-mark\}}{\{content\}}
\marksthecs{\{named-mark\}}{\{cs-name\}}
```

`\marksthe{\{section\}}{\{content\}}` will mark `<content>` into a mark register referenced by the control sequence `\marks@{<section>}`. **<content> is not expanded** (You can expand before `\marksthe` if you need it).

If the mark `\marks@{<section>}` is not defined, then a new mark register is allocated (with etex-\newmarks).

`\marksthecs{\{subsubsection\}}{\{cs-name\}}` will similarly mark the unexpanded content of `\cs-name` by means of the marks register `\marks@{subsubsection}`. `<cs-name>` is really the ***name of the control sequence***, not the control sequence itself. If `\cs-name` is undefined or `\relax`, nothing is marked, making `\marksthecs` suitable to use with `\lastsectiontitle` etc.. at `\sectionbreak` (see `interfaces-hypbmsec` p. 7).

```
\thefirstmarks{\{named-mark\}}EXPANDABLE
\thebotmarks{\{named-mark\}}EXPANDABLE
\thetopmarks{\{named-mark\}}EXPANDABLE
```

`\thefirstmarks{\{chapter\}}` expands to `\firstmarks\marks@{chapter}` (this is expandable in exactly one step of expansion), *ie.* either the content of the first invocation of `\marksthe{\{chapter\}}` on the current page, if `\marks` was used on the current page, or the last invocation of `\marks` if no marks occurs on the current page.

`\thebotmarks{\{chapter\}}` expands to `\botmarks\marks@{chapter}` *ie.* the content of the last invocation of `\marksthe{\{chapter\}}` (the most recent `\marks`).

`\thetopmarks{\{chapter\}}` expands to `\topmarks\marks@{chapter}` and this is the content of `\botmarks` at the time TeX shipped out the last page.

If there is no mark or if the mark register does not exist, expansion is null (*ie.* nothing is done).

Those commands can be used in headers and footers, f.ex:

```
\sectionformat{section}
    mark=\marksthe{section}{\arabic{section}\hspace{.4em}\textendash\hspace{.4em}\#1}
\pagesetup{head/right=\thefirstmarks{section}}
```

```
\gettmarks\firstmarks|\botmarks|\topmarks{\{named-mark\}}{\control-sequence}
\getthefirstmarks{\{named-mark\}}{\control-sequence}
\getthebotmarks{\{named-mark\}}{\control-sequence}
\getthetopmarks{\{named-mark\}}{\control-sequence}
```

You may wish to retrieve the values of `\firstmarks`, `\botmarks` or `\topmarks` into a macro. This is the purpose of these commands: `\control-sequence` is defined as a parameterless macro whose replacement text is the content of the specified marks register.

If the marks register `\marks@{named-mark}` does not exist, the meaning of `\control-sequence` is **undefined**.

### `\ifmarksequal{\firstmarks}{\topmarks}{named-mark}{\true}{\false}`

Pretty often, we want to compare the botmarks with firstmarks or the topmarks, to adapt the header and/or footer in case those marks are equal or different, *i.e.* in case the page contains a new section title or not:

`\ifmarksequal` expands conditionnally code `{\true}` only if the specified marks register to compare are equal, otherwise `{\false}`.

- `\firstmarks=\botmarks` ⇔ there is at most one section title on the current page;
- `\topmarks=\botmarks` ⇔ there is no section title on the current page;
- `\firstmarks=\topmarks` ⇔ the last section title continues on the current page.

Those assertions are true only if you mark the section title at `\sectionmark` and at `\sectionbreak`: this is easy when using `interfaces-hypbmsec`: `\sectionformat\subsection{break=\marksthecs{subsection}{lastsubsectiontoc}}`.

If the marks register `\marks@{named-mark}` does not exist, the `{\false}` part is expanded.

## 10 Interface ↗ `makecell` : `\repeatcell`

`makecell` provides `\eline`, `\nline`, `\Xrows` to fill a table with empty cells, or repeated rows.

`interfaces` extends the possibility with `\repeatcell`. Although it is of general use, `\repeatcell` should be used softly in order to keep the `.tex` document intelligible (see the examples below).

While repeating cells, `\repeatcell` can also make calculations with:

- `\numexpr`,
- `$\pgfmath$` ( $\sim \usepackage{pgfmath}$  required in preamble)
- `$$\FPeval$` ( $\usepackage{fp}$  required in preamble) ⇒ use `round` to display the results!

Inside formulas `\row` expands to the current row, `\column` to the current column number (`\row` and `\column` are 1-based and relative to `\repeatcell`).

### `\countvalues{<comma separated list>}`

`\countvalues{<comma separated list>}` expands to the number of values in the list. This is also the number of steps a `LATEX` `\@for` loop would do on the list: in other words, **empty values are counted**. `\countvalues` is provided to allow you to adjust automatically the number of columns of a tabular, depending on your data, with the `*` operator: `\begin{tabular}{*{\countvalues{list}}{c}}`.

`\countvalue` is purely expandable in one expansion step. The `<comma separated list>` is expanded once, exactly like in `LATEX` `\@for` (*hint*: you can put `\@empty` before, to avoid expansion).

|                                                                            |                                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\repeatcell[&lt;alignment&gt;]{&lt;columns&gt;}{&lt;text&gt;}</code> | ← number of columns or comma separated list of values                                                                                                                                                                                                                                            |
|                                                                            | <code>text</code> =the text to be inserted (repeated) in each cell                                                                                                                                                                                                                               |
|                                                                            | <code>font</code> =the font inside the cell                                                                                                                                                                                                                                                      |
|                                                                            | <code>rows</code> =the number of rows (default is 1 row) or a comma separated list of value:<br><code>{one,two,three,four,etc.}</code> Empty values <b>are not</b> ignored.                                                                                                                      |
|                                                                            | <code>cr end</code> =the end-of-row sequence (default is <code>\backslash</code> but you may want <code>\crrc\rhline</code> or<br><code>\tabularnewline...</code> )                                                                                                                              |
|                                                                            | <b>end of row sequence is never inserted after the last row.</b>                                                                                                                                                                                                                                 |
| <code>text/row1 text/row2 text/row3 etc.</code>                            | If you want to change the text on the specified rows                                                                                                                                                                                                                                             |
| <code>text/col1 text/col2 text/col3 etc.</code>                            | If you want to change the text on the specified columns                                                                                                                                                                                                                                          |
| <code>font/row1 font/row2 font/row3 etc.</code>                            | If you want to change the font on the specified rows                                                                                                                                                                                                                                             |
| <code>font/col1 font/col2 font/col3 etc.</code>                            | If you want to change the font on the specified columns                                                                                                                                                                                                                                          |
| <code>cr/row1 cr/row2 cr/row3 etc.</code>                                  | If you want to change the end-of-line sequence on the specified rows                                                                                                                                                                                                                             |
| <code>transpose</code>                                                     | Do you wish to invert rows and columns ? Then <code>transpose=true</code> ☺.                                                                                                                                                                                                                     |
| <code>any other key</code>                                                 | can be used to make calculations using <code>\numexpr</code> or <code>\$\pgfmathparse\$</code><br>or <code>\$\$\FPeval\$</code> . Pay attention to put your formula inside curly brackets<br><code>{\\$formula\\$}</code> if it contains a comma <code>,</code> or an equal <code>=</code> sign! |

☞ If `[<alignment>]` is specified, then each cell is placed into a `\makecell[<alignment>]` command.

☞ If both `text/row<i>` and `text/col<i>` are specified for the same cell, then `makecell \diaghead` is used to display the values in a diagonally divided cell.

☞ If you want to see (in the `.log` file) the code introduced into the tabular (or array) by `\repeatcell` type  
`\let\itf@message=\message` before `\repeatcell`.

**Formulas:** if you say: `x = $sin(10*\column)$` for example, then the formula is evaluated in each cell with `\pgfmath`. The result is stored into the control sequence `\x` and can be used in `text` or in any other following formula. The order in which the computed keys are given determines the order of evaluation... *of course!*

If you specify the number of columns (*resp.* of rows) as an integer, then `\repeatcell` works exactly as if you gave it a list of integers  $\{1, 2, 3, 4, \dots, p\}$ .

If you specify the rows as a comma separated list of values, the following sequences are also admissible in `text`:

- `\R\row` refer to the value of the list for the current row. `\r` is the same (unless you defined a formula named `r`).
- `\R{\langle n \rangle}` to the value of the list for row  $\{\langle n \rangle\}$

If the number of rows is specified as an integer, then `\R\row`, `\r` and `\row` are equal (to the current row index).

Similarly if you specify the columns as a comma separated list of values, the following sequences are provided:

- `\C\column` refer to the value of the list for the current column. `\c` is the same, unless you defined a formula named `c`.
- `\C{\langle m \rangle}` to the value of the list for column  $\{\langle m \rangle\}$

If the number of columns is specified as an integer, then `\C\row`, `\c` and `\col` are equal (to the current column index).

If the values in the lists are numeric, `\R`, `\C`, `\r` and `\c` can be used in formulas as well, *of course...*

~~~~~ A few examples are better than a 10 pages description ! ~~~~~

10.1 ↗ Conversion to scaled points

```
\def\listofunits{ pt , cm , mm , bp , dd , pc , in , cc ,sp }
\begin{tabu}{|>{\bfseries}|*9{X[mc]}|} \hline
\thead{units}
& \repeatcell[c]\listofunits{ text = \c, font=\bfseries } \\ \hline
\thead{value in\\ scaled\\ points}
& \repeatcell\listofunits{%
    sp =\number\dimexpr 1\c\relax,
    text=\numprint[sp]\sp} \\ \hline
\end{tabu}
```

| units | pt | cm | mm | bp | dd | pc | in | cc | sp |
|------------------------------|-----------|--------------|------------|-----------|-----------|------------|--------------|------------|------|
| value in
scaled
points | 65 536 sp | 1 864 679 sp | 186 467 sp | 65 781 sp | 70 124 sp | 786 432 sp | 4 736 286 sp | 841 489 sp | 1 sp |

10.2 ↗ Variable length cell

```
\newcommand\FAMILY[3]{%
\begin{tabu}{to .4\linewidth|*3{X[mc]}|} \hline
Mother & Father & Children \\ \hline
#1 & #2 & \makecell{\repeatcell[#3]{text=\c, transpose}} \\ \hline
\end{tabu}}
\FAMILY{Johanne}{Georg}{Priscilla, John, Tom}
\FAMILY{Nut}{Geb}{Isis, Osiris, Seth, Nephthys}
```

| Mother | Father | Children | Mother | Father | Children |
|---------|--------|--------------------------|--------|--------|------------------------------------|
| Johanne | Georg | Priscilla
John
Tom | Nout | Geb | Isis
Osiris
Seth
Nephthys |

10.3 † Empty rows

```
\begin{tabu}{*6{|c|}} \hline
\repeatcell{6}{text/col1=\row, rows=4, end=\\\hline} \\ \hline
\end{tabu}
```

| | | | | | |
|---|--|--|--|--|--|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |

```
\begin{tabu}{*4{|c|}} \hline
\repeatcell{6}{text/col1=\row, rows=4, end=\\\hline, transpose} \\ \hline
\end{tabu}
```

| | | | |
|---|---|---|---|
| 1 | 2 | 3 | 4 |
| | | | |
| | | | |
| | | | |
| | | | |

10.4 † color palette

```
\begin{tabular}{*4{c}}
\repeatcell{8}{transpose=true, rows=4,
red      = {$\$round(\c/8*255,0)$$},
greeni   = \red,           bluei    = \red,
greenii  = {$\$round(255-\red,0)$$}, blueii   = \red,
greeniii = \red,           blueiii  = {$\$round(255-\red,0)$$},
greeniv  = \greenii,       blueiv   = \blueiii,
text/row1=\cellcolor[RGB]{\red,\greeni,\bluei}(\red,\greeni,\bluei),
text/row2=\cellcolor[RGB]{\red,\greenii,\blueii}(\red,\greenii,\blueii),
text/row3=\cellcolor[RGB]{\red,\greeniii,\blueiii}(\red,\greeniii,\blueiii),
text/row4=\cellcolor[RGB]{\red,\greeniv,\blueiv}(\red,\greeniv,\blueiv)}}
\end{tabular}
```

| | | | |
|---------------|---------------|---------------|---------------|
| (32,32,32) | (32,223,32) | (32,32,223) | (32,223,223) |
| (64,64,64) | (64,191,64) | (64,64,191) | (64,191,191) |
| (96,96,96) | (96,159,96) | (96,96,159) | (96,159,159) |
| (128,128,128) | (128,127,128) | (128,128,127) | (128,127,127) |
| (159,159,159) | (159,96,159) | (159,159,96) | (159,96,96) |
| (191,191,191) | (191,64,191) | (191,191,64) | (191,64,64) |
| (223,223,223) | (223,32,223) | (223,223,32) | (223,32,32) |
| (255,255,255) | (255,0,255) | (255,255,0) | (255,0,0) |

Note: `makecell` interact very badly with `colortbl`. Therefore, with `\cellcolor` or `\rowcolor`, the optional `[(alignment)]` parameter of `\repeatcell` does not give good results... `colortbl` remain a rather rough package, and deserve a finer implementation... in future.

10.5 † tabularcalc-like tables (with `dcolumn` package) and `pgfmath`

```
\def\listoffx{ -4 , -2 , 0 , 2.25 , 7 }
$\$ \begin{tabu}{|c<\bottopstrut|*5{|D..{5.5}|}} \hline
\rowfont{\large\color{blue}}
x & \repeatcell{\listoffx{ text=\C\column }} & \hline
f(x)=2x-3 & \repeatcell{\listoffx{ fx = \$2*\c-3\$ , text=\fx }} & \hline
x^2 & \repeatcell{\listoffx{ xsquare = \$\C\column^2\$ , text = \xsquare }} & \hline
h(x) = \sqrt{x^2+1} & \repeatcell{\listoffx{ x=\C\column, hx = \sqrt{pow(\x,2)+1} , text = \hx }} & \hline
\end{tabu} \$\$
```

| x | -4 | -2 | 0 | 2.25 | 7 |
|-------------------------|---------|---------|---------|---------|---------|
| $f(x) = 2x - 3$ | -11.0 | -7.0 | -3.0 | 1.5 | 11.0 |
| x^2 | - 16.0 | -4.0 | 0.0 | 5.0625 | 49.0 |
| $h(x) = \sqrt{x^2 + 1}$ | 4.12310 | 2.23606 | 1.00000 | 2.46221 | 7.07106 |

$\$ \text{pgfmath} \$$ computations

| x | -4 | -2 | 0 | 2.25 | 7 |
|-------------------------|-----------|----------|----------|---------|----------|
| $f(x) = 2x - 3$ | -11.00000 | -7.00000 | -3.00000 | 1.50000 | 11.00000 |
| x^2 | 16.00000 | 4.00000 | 0.00000 | 5.06250 | 49.00000 |
| $h(x) = \sqrt{x^2 + 1}$ | 4.12311 | 2.23607 | 1.00000 | 2.46221 | 7.07107 |

And the same with $\$ \text{FPeval} \$$ (fp package)

10.6 Double entry array: the Body Mass Index

| height in cm | 190 | 180 | 170 | 160 | 150 | 140 | 130 | |
|--------------|------|------|------|------|------|------|------|-----------------|
| weight in kg | | | | | | | | |
| 30 | 8.3 | 9.3 | 10.4 | 11.7 | 13.3 | 15.3 | 17.8 | denutrition |
| 45 | 12.5 | 13.9 | 15.6 | 17.6 | 20.0 | 23.0 | 26.6 | skinniness |
| 50 | 13.9 | 15.4 | 17.3 | 19.5 | 22.2 | 25.5 | 29.6 | normal |
| 55 | 15.2 | 17.0 | 19.0 | 21.5 | 24.4 | 28.1 | 32.5 | obesity |
| 60 | 16.6 | 18.5 | 20.8 | 23.4 | 26.7 | 30.6 | 35.5 | serious obesity |
| 65 | 18.0 | 20.1 | 22.5 | 25.4 | 28.9 | 33.2 | 38.5 | morbid obesity |
| 70 | 19.4 | 21.6 | 24.2 | 27.3 | 31.1 | 35.7 | 41.4 | |
| 80 | 22.2 | 24.7 | 27.7 | 31.3 | 35.6 | 40.8 | 47.3 | |
| 90 | 24.9 | 27.8 | 31.1 | 35.2 | 40.0 | 45.9 | 53.3 | |
| 95 | 26.3 | 29.3 | 32.9 | 37.1 | 42.2 | 48.5 | 56.2 | |
| 100 | 27.7 | 30.9 | 34.6 | 39.1 | 44.4 | 51.0 | 59.2 | |
| 110 | 30.5 | 34.0 | 38.1 | 43.0 | 48.9 | 56.1 | 65.1 | |
| 120 | 33.2 | 37.0 | 41.5 | 46.9 | 53.3 | 61.2 | 71.0 | |

| height in cm \ weight in kg | 30 | 45 | 50 | 55 | 60 | 65 | 70 | 80 | 90 | 95 | 100 | 110 | 120 | |
|-----------------------------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| height in cm | 190 | 8.3 | 12.5 | 13.9 | 15.2 | 16.6 | 18.0 | 19.4 | 22.2 | 24.9 | 26.3 | 27.7 | 30.5 | 33.2 |
| 180 | 9.3 | 13.9 | 15.4 | 17.0 | 18.5 | 20.1 | 21.6 | 24.7 | 27.8 | 29.3 | 30.9 | 34.0 | 37.0 | |
| 170 | 10.4 | 15.6 | 17.3 | 19.0 | 20.8 | 22.5 | 24.2 | 27.7 | 31.1 | 32.9 | 34.6 | 38.1 | 41.5 | |
| 160 | 11.7 | 17.6 | 19.5 | 21.5 | 23.4 | 25.4 | 27.3 | 31.3 | 35.2 | 37.1 | 39.1 | 43.0 | 46.9 | |
| 150 | 13.3 | 20.0 | 22.2 | 24.4 | 26.7 | 28.9 | 31.1 | 35.6 | 40.0 | 42.2 | 44.4 | 48.9 | 53.3 | |
| 140 | 15.3 | 23.0 | 25.5 | 28.1 | 30.6 | 33.2 | 35.7 | 40.8 | 45.9 | 48.5 | 51.0 | 56.1 | 61.2 | |
| 130 | 17.8 | 26.6 | 29.6 | 32.5 | 35.5 | 38.5 | 41.4 | 47.3 | 53.3 | 56.2 | 59.2 | 65.1 | 71.0 | |

The same with transpose

Note that fp computation can be time consuming... depending on the formula and the number of cells to fill in. While preparing first such a table, give to the computed keys a constant value, and plug the formulas only when you are satisfied with the displayed tabular...

11 Interface ☈ enumitem : \setlist

```
\setlist [environment,level,environment,level,etc.]{  
    label = the label to be used  
    font = the label font  
    topsep = vertical space before the list  
    partopsep = vertical space  
    parsep = value of \parskip inside the list  
    itemsep = vertical space between two items  
    leftmargin = the list left margin  
    rightmargin = the list right margin  
    listparindent = the value of \parindent inside the list  
    labelwidth = the width of the label box  
    labelsep = horizontal space between label and following text  
    itemindent = the indentation of the items }  
\setlist* [<environment,level,environment,level,etc.>]{key=value}
```

Enumerated lists, description lists or itemize lists can be defined by `enumitem` with the command `\newlist`.

For example: `\newlist{mylegal}{enumerate}{10}`

The list can be customized with `\setlist`, for example:

```
\setlist[mylegal,1]{label*=\arabic*.}  
\setlist[mylegal,2]{label*=\arabic*.}  
\setlist[mylegal,3]{label*=\roman*)}
```

With `interfaces-enumitem` you may specify more than one level at a time in `\setlist`, (and even more than one environment at a time):

```
\setlist[mylegal,1,2]{label*=\arabic*.}  
\setlist[mylegal,3]{label*=\roman*)}
```

A list can also be **modified** with `\setlist*`: keys that are not specified remain the same.

```
\setlist*[mylegal,1,2,3]{align=left, leftmargin=*}
```

or

```
\setlist*[mylegal]{align=left, leftmargin=*}
```

\interitem

Just like $\text{\textit{AMS}}$ `\intertext` in math formulæ, `\interitem` allows to put some non intended text inside a list environment. Example:

```
\begin{enumerate}[label=\roman*)], ,parsep=0pt]  
    \item First item in the enumerate envir  
    \interitem or (intertext)  
    \item Second item  
    \end{enumerate}
```

The example give in real world:

- i) First item in the enumerate envir
- or: (intertext)
- ii) Second item

And the following follows...

12 Interface ↳ truncate

The package `truncate` by Donal Arseneau provides the command:

```
\truncate[⟨TruncateMarker⟩]{⟨width⟩}{⟨text⟩}
```

to trunc the given `⟨text⟩` to the specified `⟨width⟩`, by means of `⟨TruncateMarker⟩` if the the truncation occurs.

```
\trunctext[⟨TruncateMarker⟩]{⟨width⟩}{⟨text⟩}
```

`\trunctext` is the same as `\truncate` but it leaves `{text}` to its natural width if it has not to be truncated (*i.e.* if $\langle width \rangle \leq$ width of `⟨text⟩`). Example:

```
truncate: \truncate{<5cm>}{Hello world!}
trunctext: \trunctext{<5cm>}{Hello world!}
```

Hello word!

Hello world!

This corresponds to the `[fit]` option of the `truncate` package.

```
\htruncadjust{⟨total width⟩}[⟨x1, x2, ..., xn⟩]{⟨text1, text2, ..., textn⟩}
```

This macro will measure the width of `text1,text2,...,textn` and adjust them for truncation:

| text | width | adjusted width | γ (truncated?) | Returned width |
|-------------------|-------|---|--|---|
| text ₁ | w_1 | $w'_1 = \min\left(w_1, \frac{x_1}{\sum x_i} \times \langle total\ width \rangle\right)$ | $\gamma_1 = \begin{cases} 1 & \text{if } w'_1 < w_1 \\ 0 & \text{otherwise} \end{cases}$ | $\omega_1 = w'_1 + \gamma_1 \cdot \Phi \cdot \frac{x_1}{\sum \gamma_i x_i}$ |
| text ₂ | w_2 | $w'_2 = \min\left(w_2, \frac{x_2}{\sum x_i} \times \langle total\ width \rangle\right)$ | $\gamma_2 = \begin{cases} 1 & \text{if } w'_2 < w_2 \\ 0 & \text{otherwise} \end{cases}$ | $\omega_2 = w'_2 + \gamma_2 \cdot \Phi \cdot \frac{x_2}{\sum \gamma_i x_i}$ |
| text ₃ | w_3 | $w'_3 = \min\left(w_3, \frac{x_3}{\sum x_i} \times \langle total\ width \rangle\right)$ | $\gamma_3 = \begin{cases} 1 & \text{if } w'_3 < w_3 \\ 0 & \text{otherwise} \end{cases}$ | $\omega_3 = w'_3 + \gamma_3 \cdot \Phi \cdot \frac{x_3}{\sum \gamma_i x_i}$ |
| text ₄ | w_4 | $w'_4 = \min\left(w_4, \frac{x_4}{\sum x_i} \times \langle total\ width \rangle\right)$ | $\gamma_4 = \begin{cases} 1 & \text{if } w'_4 < w_4 \\ 0 & \text{otherwise} \end{cases}$ | $\omega_4 = w'_4 + \gamma_4 \cdot \Phi \cdot \frac{x_4}{\sum \gamma_i x_i}$ |
| | | $\Phi = \text{freespace} = \langle total\ width \rangle - \sum w'_i$ | | $\sum \omega_i \leq \langle total\ width \rangle$ |

Another example of a tabular made with $\sim\backslash\text{repeatcell}$

The widths can be accessed with `\htruncwidth{⟨i⟩}` and then:

```
\trunctext[⟨TruncateMarker⟩]{\htruncwidth{⟨i⟩}}{⟨texti⟩}.
```

x_i can be any real number provided that $\sum_i x_i \leq 16\,383 = \text{\maxdimen}$ in points.

Measuring the widths occur inside a `\hbox`, thus `text1,...,textn` can contain macros, non expandable material etc.. For example:

`\htruncadjust{4cm}[2,1]{Hello,world}` will prepare the widths so that their sum does not exceed 4 centimeters, and $w_1 = 2 \times w_2$.

`\htruncadjust{4cm}[2,1,2]{The,World,Is flat}` will prepare the widths so that their sum does not exceed 4 centimeters, and $w_1 = w_3 = 2 \times w_2$.

If the optional parameters x_i are not given, then they are equal to the natural width of `{text1,...,textn}`. In other words, `text1,...,textn` are truncated to a width proportional to their natural widths. (And the sum of their widths will not exceed `⟨total width⟩`).

This macro has been designed to cleverly truncate the text of headers, when they contain titles on both sides... Usually then, there are only two pieces of text whose width have to be adjusted for truncation...

```
\htruncadjust{3cm}{first quite long,second short,third} % natural widths
\trunctext{\htruncwidth{1}}{first quite long}
\trunctext{\htruncwidth{2}}{second short}
\trunctext{\htruncwidth{3}}{third}
```

→ first quit... second... t... (39.05698pt, 33.65436pt, 12.64691pt, total=85.35825pt).

```
\htruncadjust{3cm}[3,2,2]{first quite long,second short,third} % first is 3/2 of the two others
```

→ first quite...secon...third (41.01503pt, 27.34372pt, 17.00078pt, total=85.35953pt).

13 Interface ↗ appendix and the appendices environment

`appendix` provides a way to write appendices inside an environment. This feature is suitable for short appendices like the one included in this document. `interfaces-appendix` gives an easier control on some parameter for the appendix with standard (optional) key names.

```
\begin{appendices}[%  
    section = section | chapter,      alternatively, the key chapter is provided  
    title = the appendix title,       if left blank, title is removed: [page] option of appendix  
    intoc = true | false,            put an entry in t.o.c. ?  
    intoc/title = the entry in t.o.c.] if blank, no title in t.o.c., if no value: the appendix title  
    appendix content ...  
\end{appendices}
```

14 Interface ↗ embedfile : \embedthisfile

```
\embedthisfile  
\embedthisfile*
```

`\embedthisfile` will embed into the final .pdf file, the current .tex file and all .tex files that are included (with `\include`) in it. If `\includeonly` is used, the only files embedded (*ie.* attached) to the .pdf file are the ones specified by `\includeonly`.

`\embedthisfile*` embeds only the main file.

This macro could disappear from `interfaces-embedfile` if it is added to the main package... Depending on Mr Oberdieck moods ☺.

15 Interface ↗ environ

`environ` defines `\Renewenviron` but I found out it was usefull to have `\ReNewEnviron` (note the case!).

Therefore: `\let\ReNewEnviron\RenewEnviron` is made by `interfaces` (`\AfterPackage{environ}`).

16 Interface umrand

`umrand` defines the macro `\ifnotempty` which is already defined with a different meaning in `etextools`. `interfaces` provides a turn around to avoid redefinition.

For this turn around, `interfaces` must be loaded before `umrand` (because we use `\BeforePackage{umrand}`). This is – really – the only exception!

17 Interface pgfkeys (⇨ pgfmanual)

pgfkeys's the base of all... wonderfull and inimitable! ☺

The following handlers allows to build fairly quickly a key=value interface for commands in almost any case:

~~~~~ SUMMARY ~~~~~

| | |
|--|--|
| <code>./blank</code> | Replaces blank value by another value. |
| <code>/is private</code> | Throws an error if the key is used in an assignment |
| <code>/store in cs ./estore in cs</code> | Like <code>/store in</code> but with <code>\csname...\endcsname</code> |
| <code>/true ./false</code> | <code>/is choice</code> shortcut for boolean keys |
| <code>/on ./off</code> | |
| <code>/is multiple choice</code> | Like <code>/is choice</code> but more than one choice can be selected at a time |
| <code>/is enum</code> | <code>/is choice</code> shortcut for enumerated keys
also provides the <code>/key/.number</code> key to get the numeric index of the selected choice. |
| <code>/store as code</code> | Stores the given value in the key itself. <code>+</code> and <code>*</code> keys both append code. |
| <code>/estore as code</code> | The same but stores the <i>expanded value</i> . |
| <code>/store as code in cs</code> | Stores the given value in a cs-name rather than in the key. <code>+</code> and <code>*</code> keys both append code. |
| <code>/estore as code in cs</code> | The same but stores the <i>expanded value</i> . |
| <code>/store as color</code> | Stores the color in the key itself. |
| <code>/store as color command</code> | The same but stores the color <i>command</i> . |
| <code>/store as color in cs</code> | Stores the color in a cs-name rather than in the key. |
| <code>/store as color command in cs</code> | The same but stores the color <i>command</i> . |
| <code>/store as number</code> | Stores an integer in the key (<code>-</code> <code>+</code> and <code>*</code> keys are provided to add, subtract and multiply). |
| <code>/estore as number</code> | |
| <code>/store as number in cs</code> | The same, but stores into a cs-name rather than in the key. |
| <code>/estore as number in cs</code> | |
| <code>/store in counter cs</code> | The same, but stores into a TeX counter. |
| <code>/store as glue</code> | Stores a glue expression in the key (<code>-</code> <code>+</code> and <code>*</code> keys are provided to add, subtract or scale). |
| <code>/estore as glue</code> | The same, but stores the <i>computed value</i> of the glue, not its expression. |
| <code>/store as glue in cs</code> | The same, but stores in a given cs-name rather than in the key itself. |
| <code>/estore as glue in cs</code> | |
| <code>/store in skip cs</code> | The same, but stores the glue inside a TeX <code>\skip</code> register. |
| <code>/store as dimen</code> | Store a dimen expression in the key (with <code>-</code> <code>+</code> and <code>*</code> keys to add, subtract or scale). |
| <code>/estore as dimen</code> | The same, but stores the <i>computed value</i> of the dimen, not its expression. |
| <code>/store as dimen in cs</code> | The same, but stores in a given cs-name rather than in the key itself. |
| <code>/estore as dimen in cs</code> | |
| <code>/store in dimen cs</code> | Stores the dimension into a TeX <code>\dimen</code> register rather than in a macro. |
| <code>/alias</code> | Creates an alias for the key. |
| <code>/alias*</code> | The same, but also creates the alias for the <code>+</code> , <code>-</code> and <code>*</code> keys if they exist. |

The command `\pgfkeysmeaning` is provided (with `\providetcommand`): it expands to the meaning of the given pgfkey.

17.1 † Handler definition: `./blank`

A new handler `./blank` is defined to allow automatic replacement of empty or blank values by the “default” value specified:

```
\pgfkeys{/base/path/my key/.code=The key #1,%  
        /base/path/my key/.default=has its default value!,%  
        /base/path/my key/.blank=was left blank!}  
\pgfkeys{/base/path/my key,      produces The key has its default value!  
        /base/path/my key=}       produces The key was left blank!
```

The following make blank value behave the same as default :

```
\pgfkeys{/base/path/my key/.blank=\pgfkeysnovalue}
```

And If you wish to forbid blank values you can add:

```
\pgfkeys{/base/path/my key/.value required}
```

With `interfaces-pgfkeys` it is also possible to give a `./default` or/and a `./blank` value to a handler, or on the other hand to say that a handler must have `(./value required)` or does not suffer `(./value forbidden)` a value. (this feature was not implemented in `pgfkeys.code.tex`).

17.2 † Handler definition: `./is private`

This handler is mainly addressed to package writers. A private key does not suffer any assignment: this is a package constant:

```
\pgfkeys{/base/path/@my key/.is private=mypackage}
```

Then any assignment to `@my` key will throw an error, explaining that `@my` key is private to the package `mypackage`.

Similarly, there is a `./is private to class` handler (for `\ClassError` instead of `\PackageError`).

17.3 † Handler definition: `./store in cs` and `./estore in cs`

A new handler `./store in cs` is defined to store the value of a key in a named control sequence:

```
\pgfkeys{%
  /base/path/my key/.store in cs=some word,% will define : \"some word"
}
```

17.4 † Handlers definition: `./true`, `./false`, `./on` and `./off`

A key of type `./true` or `./false` can take the only values `true` or `false` and its default value is `true` (unless `./default=false` is explicitly given to `\pgfkeys`).

Similarly, a key of type `./on` or `./off` can take the only values `on` or `off` and its default value is `on` (unless `./default=off` is explicitly given to `\pgfkeys`).

These are shortcuts and a special case of `./is choice` keys:

```
\pgfkeys{
  my key/.true = code if the value is true,
  my key/.false = code if the value is false,
  my key2/.on = code if the value is on,
  my key2/.off = code if the value is off }
```

17.5 † Handler definition: `./is multiple choice`

A key defined with `./is multiple choice` behaves like a `./is choice` key but more than one choice can be selected at a time:

```
\pgfkeys{/path/.cd,
  format/.is multiple choice,
  format/bold/.code= ...,
  format/italic/.code= ...,
  format/color/.code= ... }
```

Possible values for key `format` are restricted to subkeys: `bold`, `italic` and `color`. But unlike simple `./is choice` keys, it is possible to use:

```
\pgfkeys{/path/format={bold,italic,color=...}}
```

this is the same as:

```
\pgfkeys{/path/format=bold,/path/format=italic,path/format={color=...}}
```

The `bookmark` key of $\sim \text{\textcolor{red}{sectionformat}}$ (in `interfaces-titlesec`) is a good example.

17.6 † Handler definition: `./is enum`

A key defined with `./is enum` behaves like a `./is choice` key: its possible values are limited to sub-keys. The name of the selected sub-key is stored into the key.

```
\pgfkeys{/path/.cd,
  enumKey/.is enum={one,two,three}
}
```

`enumKey` can be `one` or `two` or `three` and nothing else.

```
\pgfkeys{ enumKey=two }
```

assigns two to the value of the key: `\pgfkeysvalueof{enumKey}` is two.

`\pgfkeysvalueof{enumKey/.number}` is 1 (zero based index)

17.7 † Handlers definition: `./store as code`, `./estore as code`

This handler stores the value given to the key in the key itself. Furthermore, a starred key `*` and a `+` keys are defined such that they append their value to the stored key.

```
\pgfkeys{/test./store as code,
          /test=some code,
          /test+= the following follows... }
```

Defines the key `/test` so that `\pgfkeysvalueof{/test}` expands to `some code`. `/test+` appends “the following follows” to the previous value of the key. (`/test*` does the same).

If you prefer not to have star keys `*`, or `+` keys, or only one of these, just say, for example:

```
\pgfkeys{/test./store as code=+}
```

This feature is very important for `interfaces`: almost any `font` keys are defined with the handler `./store as code`.

A note for programmers: `./store as code` defines three keys: `test`, `test+` and `test*`. (The two former are the same). The “append code” keys (`+` and `*`) call the *main key* `test`. This is handy because if you append code to the *main key* (with `pgfkeys ./add code` or `./append code`) then this code will be expanded also in case the `+` or `*` keys are called.

There is also a `./estore as code` handler which expands the given code before storing it in the key.

17.8 † Handlers definition: `./store as code in cs=<cs-name>`, `./estore as code in cs=<cs-name>`

This handler works like `./store as code`: the content is stored in the `cs-name` given as value for the handler, but not inside the key itself (useful for key=value interfaces to other packages).

The `+` and `*` keys are provided: both append their value to the content of the given control sequence.

Typically, `/font` keys are defined this way for interface with other packages.

There is also a `./estore as code in cs` handler which expands the given code before storing it in the `<cs-name>..`

17.9 † Handler definition: `./store as color`

When specifying a color, did you say:

| | |
|-----------------------------------|---------------------------------|
| <code>color=\color{yellow}</code> | <code>color=\normalcolor</code> |
| <code>color=blue</code> | <code>color=[rgb]{1,0,0}</code> |

In any case, `./store as color` will transform the value as follow:

| | |
|---------------------------|--|
| <code>color=yellow</code> | <code>color=default</code> defined by <code>interfaces</code> if not already |
| <code>color=blue</code> | <code>color=[rgb]{1,0,0}</code> defined <code>\AtBeginDocument</code> |

In any case, `./store as color command` will transform the value as follow:

| | |
|---------------------------------|---------------------------------------|
| <code>color=\normalcolor</code> | <code>color=\color{yellow}</code> |
| <code>color=\color{blue}</code> | <code>color=\color[rgb]{1,0,0}</code> |

Please! Dont write: `\def\myBlue{blue}` `color=\myBlue`.

Use `\colorlet{myBlue}{blue}` instead, or `\definecolor`.

As for code, the color is stored inside the key.

17.10 † Handler definition: `./store as color in cs=<cs-name>`

The same as `./store as color`, but the color is stored in the given control sequence, rather than in the key itself.

There is also a `./store as color command in cs` handler. It is used to set the value to `\columnseprulecolor` by the key `columns/rule/color` of $\sim\backslash\text{tocsetup}$.

17.11 † Handlers definition: `./store as number`, `./estore as number`**17.12 † Handlers definition: `./store as number in cs`, `./estore as number in cs`****17.13 † Handler definition: `./store as glue`**

This is quite the same as `./store as code`, but the given code is surrounded by `\glueexpr...``\relax...` and is stored inside the key.

Furthermore:

the `+` key adds to the length: `\glueexpr old value + < argument > \relax`.
 the `-` key subtracts: `\glueexpr old value - < argument > \relax`
 the `*` key scales: `\glueexpr old value * < argument > \relax`

The `<argument>` is **not expanded** but the content of the key is ready to be used in place of a length. Scaling of two forms are allowed: *fractionnal* and *real*. *Real* scaling is converted to a fraction of 65536:

```
\pgfkeys{/test/.store as glue,
          /test=12pt plus5pt,
          /test*=.5,
          /test+=\parskip,
          /test*=4/6
}
\pgfkeysmeaning{/test}
```

→ macro:->`\glueexpr \glueexpr \glueexpr 12pt plus5pt\relax *32768/65536\relax +\glueexpr \parskip \relax \relax *4/6\relax`

The value of `\parskip` is not expanded at `\pgfkeys` time, but only when the value of the key is used (for an assignment...)

There is also a handler called `./estore as glue` if you really want the value to be expanded (with `\the`) at the time of `\pgfkeys` setting.

17.14 † New handler: `./store as dimen`

Just like `./store as glue` but with `\dimexpr` instead of `\glueexpr`.

There is also a `./estore as dimen` handler, for immediate expansion.

17.15 † New handler: `./store as glue in cs=<cs-name>`

This is like `./store as glue` but the the value is assigned to a macro whose name is given, rather than in the key itself. Expansion *does not* occur `\pgfkeys` time: the macro contains the glue definition in `\glueexpr...``\relax` expressions.

`+` (to add) `-` (to subtract) and `*` (to scale) keys are provided.

There is also a `./estore as glue in cs` handler, and expansion occurs when the key is set.

17.16 † New handler: `./store as dimen in cs=<cs-name>`

Just like `./store as glue in cs` but with `\dimexpr` instead of `\glueexpr`.

17.17 † New handler: `./estore as dimen in cs=<cs-name>`

This handler as a particularity: if the value is given without dimension, `pt` is assumed (that is... points).

This is the way the key `size` is defined for the command `\changefont` (in `interfaces-LaTeX`). `size` defines `\f@size`, the macro used by L^AT_EX to store the font size (and `size+` adds, `size-` subtracts, `size*` scales). But `\f@size` is stored in points, without the mention of the dimension: `pt`.

With its default units in `pt`, `./estore as dimen in cs` is the handler of the situation:

18 Interface ↗ etoolbox - etoolbox-like commands

May be they will be added to `etoolbox.sty` one day or another... and disappear from `interfaces` ...

18.1 † New macros provided with \providetcommand

| | |
|---|--|
| <code>\removefrom {hook}{<code>}</code> | <code>\removefromstart{hook}{<code>}</code> |
| <code>\gremovefrom{hook}{<code>}</code> | <code>\gremovefromstart{hook}{<code>}</code> |
| <code>\eremovefrom{hook}{<code>}</code> | <code>\eremovefromstart{hook}{<code>}</code> |
| <code>\xremovefrom{hook}{<code>}</code> | <code>\xremovefromstart{hook}{<code>}</code> |

`\removefrom` removes the first occurrence of `{<code>}` from a `<hook>`. The command is robust.

`{<code>}` should not contain groups: `{...}` but parameter characters are allowed, which need not to be doubled. The match for `{<code>}` inside the replacement text of `<hook>` is category-code sensitiv.

If the first occurrence of `{<code>}` into `<hook>` has been removed

then `\ifin@` is set to `\iftrue`

otherwise `\ifin@` is set to `\iffalse` (`<hook>` does not contain `{<code>}`).

this allows loops of the form:

`\in@true`

`\@whilesw\ifin@\fi{\removefrom{hook}{<code>}}`

☞ `\eremovefrom` and `\xremovefrom` expand `{<code>}` before trying to remove it from `<hook>`. Only `{<code>}` is expanded, the current replacement text of `<hook>` is not.

☞ `g` and `x` versions make the assignment to `<hook>` global.

☞ `\removefromstart` remove `{<code>}` only if it appears at the very beginning of `<hook>`.

19 Interface ↗ scrlfile

19.1 † Star forms for \BeforeFile and \AfterFile

`scrlfile` provides `\BeforeFile` and `\AfterFile` in order to execute some code when a file is loaded. The hooks are automatically reset to `\relax` after the file has been loaded, in other words, they are expanded at most one only time. This can be avoided by using `\BeforeFile*` and `\AfterFile*`, for example:

```
\BeforeFile*{\jobname.toc}{\begin{multicols}{2}}
\AfterFile*{\jobname.toc}{\end{multicols}}
```

The hooks `before` and `after` will be executed **every time** `\jobname.toc` is loaded.

This feature is used in `\tocsetup{twocolumns}`.

19.2 † Allowing \BeforeFile and \AfterFile when \@input is invoked

The hooks `\BeforeFile` and `\AfterFile` are executed only when the file is loaded with `\InputIfFileExists`: indeed this macro is redefined by `scrlfile`.

`interfaces` redefine `\@input` in order to execute those hooks also in case `\@input` is used: this is useful because `.toc` files are input with `\@input` rather than with `\InputIfFileExists`.

20 List of corrected or patched commands

Here is the complete list of patched commands in other packages.

The modification occurs only if the given module of `interfaces` is loaded. For example, `\addcontentsline` is modified only if packages `titlesec` and `bookmark` are loaded, *and if* `interfaces-titlesec` and `interfaces-bookmark` are loaded.

20.1 † `LATEX`

| | Patched command | Type | Importance | Module | ref. page |
|---|--|------|-------------------------------------|------------------|-----------|
| ❶ | <code>\@pkgextension</code>
<code>\@clsextension</code>
<code>\@ifpackageloaded</code>
<code>\@ifclassloaded</code> | | minor
optimisation or correction | interfaces-LaTeX | 78 |

Those commands are replaced by their equivalent in the package `ltxcmds`:

```
\ltx@pkgextension \ltx@clsextension \ltx@ifpackageloaded \ltx@ifclassloaded
```

This is because `LATEX` defines those commands `\@onlypreamble`, while the equivalent macros in `ltxcmds` can be used inside document as well.

| | Patched command | Type | Importance | Module | ref. page |
|---|-------------------------------|----------------|---------------------------------|--|-----------|
| ❷ | <code>\addcontentsline</code> | <i>prepend</i> | important
but no side effect | interfaces-titlesec
interfaces-bookmark | 40 |

`\addcontentsline` does a lot a thing and in particular it expands the `\bookmark` command (if the `bookmark` package is loaded, otherwise it's `hyperref`). As long as `\sectionformat` can drive the attributes of the bookmark for each section, it is necessary to expand `\bookmarksetup` with the right options before the section-bookmark is written.

This is achived by `\itf@tt1@BKM`, placed at the beginning of `\addcontentsline`.

| | Patched command | Type | Importance | Module | ref. page |
|---|----------------------|------|---------------------------------|--------------------|-----------|
| ❸ | <code>\@input</code> | | important
but no side effect | interfaces-scrfile | 79 |

The `\input` primitive (same as `LATEX` control sequence `\@@input`) is surrounded by `\scr@load@hook{before}... \scr@load@hook{after}`:

Original:

```
\def\@input#1{\IfFileExists{#1}
  { \@@input\@filef@und
    {\typeout{No file #1.}}%
  }%
```

Modified:

```
\renewcommand*\@input[1]{\IfFileExists{#1}
  { \scr@load@hook{before}{#1}%
    \@@input\@filef@und
    \scr@load@hook{after}{#1}%
    {\typeout{No file #1.}}%
  }%
```

20.2 † `bookmark` \`\bookmark`

| | Patched command | Type | Importance | Module | ref. page |
|---|------------------------|--------------|------------|---------------------|-----------|
| ❹ | <code>\bookmark</code> | <i>modif</i> | important | interfaces-bookmark | 42 |

The `*` keys for `\bookmarksetup` (`italic*` , `color*` , `bold*` , and `disable*`) are local to the group, but govern only the very next bookmark.

After the bookmark has been written, previous values must be restored at the end of each group until the same group level as the one in which the `*` option has been set is reached (if this group is the main group, level 0, then a global assignment is made).

Well... this is done at `\write: \immediate\write{the bookmark entry}` is replaced by `\itf@BKM@write{the bookmark entry}` to check first:

- 1) is the bookmark disable ?
- 2) is it necessary to restore previous values after write (*i.e.* `\aftergroup`) ?

20.3 † `tocloft` \`\newlistof`

| | Patched command | Type | Importance | Module | ref. page |
|---|-------------------------|------|---------------------------------|--------------------|-----------|
| ❺ | <code>\newlistof</code> | | important
but no side effect | interfaces-tocloft | |

20.4 † `colortbl`

| | Patched command | Type | Importance | Module | ref. page |
|---|------------------------|--------------|---------------------------------|----------------------|-----------|
| ❻ | <code>\CT@cellc</code> | <i>corr.</i> | important
but no side effect | interfaces-maketable | 74 |

`colortbl` modifies the preamble of tabulars: it put each cell in a box, sets the color and then displays the box. The example shown page 14 does not work without a minor modification: the color specification must be expanded short before their usage.

Original:

```
\def\CT@cellc#1[#2]#3{%
  \expandafter\gdef\expandafter\CT@cell@color\expandafter{%
    \expandafter\CT@color#1{[#2]}{#3}%
    \global\let\CT@cell@color\relax
  }{}}
```

Modified:

```
\def\CT@cellc#1[#2]{% #1 is either \@firstofone or \@gobble, [#2]#3 = color spec
  \xdef\CT@cell@color{\noexpand\CT@color#1{{#2}}}{#3}%
  \global\let\noexpand\CT@cell@color\relax}%
}
```

Remember `arydshln` must be loaded after `colortbl` (this is not due to `interfaces`, but to the fact that both redefine `\@classz` from the `array` package).

20.5 † `hypbmsec`

| | Patched command | Type | Importance | Module | ref. page |
|---|---|--------------|---------------------------------|--|-----------|
| 7 | <code>\hbs@do</code>
<code>\hbs@process</code> | <i>redef</i> | important
but no side effect | <code>interfaces-hypbmsec</code>
<code>interfaces-titlesec</code> | 45 |

In order to get `\lastsectiontitle`, `\lastsectionbookmark` and `\lastsectiontoc` updated at the right place (*i.e.* `after \sectionbreak` and `before \sectionmark`), some `interfaces`-macros have to be executed:

- at the beginning of `\hbs@do` → to get the word `section` or `subsection etc..`
- at the beginning of `\hbs@process` → to store the next section title, bookmark and toc

20.6 † `enumitem`

| | Patched command | Type | Importance | Module | ref. page |
|---|-----------------------|--------------|-------------------------------------|----------------------------------|-----------|
| 8 | <code>\setlist</code> | <i>redef</i> | minor
optimisation or correction | <code>interfaces-enumitem</code> | 74 |

`\setlist` is redefined by `interfaces-enumitem` in order to extend the syntax.

20.7 † `pgfkeys.code.tex`

| | Patched command | Type | Importance | Module | ref. page |
|---|--|--------------|------------|---------------------------------|-----------|
| 9 | <code>\pgfkeys@unpack</code>
<code>\pgfkeys@case@three</code> | <i>redef</i> | important | <code>interfaces-pgfkeys</code> | 80 |

`\pgfkeys@unpack` is modified to handle the case of the `./blank` handler: this is here that the value is conditionnally compared against `./default`, `./blank` or `./value required`.

`\pgfkeys@case@three` is modified to allow `handlers` to have `./default`, `./blank` or `./value required` values.

`\pgfkeys@exp@call`: missed a pair of braces around the value in `\pgfkeysalso`. Thus the code could break if the value contains an equal sign or a comma. Fixed.

21 Implementation

21.1 Interfaces

The package namespace is `itf@`.

```

1 <*package>
2 \NeedsTeXFormat{LaTeX2e}[2005/12/01]
3 \ProvidesPackage{interfaces}
4   [2010/12/27 v2.6 - pgfkeys interfaces - pgf 2.0 et 2.1 (FC)]

```

21.1.1 Requirements

```
5 \RequirePackage{interfaces-base}
```

21.1.2 The package options / list of interfaces

```

6 \DeclareOption{LaTeX}{\AtEndOfPackage{\LoadInterface{LaTeX}}}
7 \DeclareOption{marks}{\AtEndOfPackage{\LoadInterface{marks}}}
8 \def\itf@tp#1{%
9   \DeclareOption{#1}{\AtEndOfPackage{\LoadInterface{#1}}}%
10 }% \itf@tp
11 \itf@tp{titlesec}
12 \itf@tp{hyperref}
13 \itf@tp{bookmark}
14 \itf@tp{fancyhdr}
15 \itf@tp{tocloft}
16 \itf@tp{makecell}
17 \itf@tp{enumitem}
18 \itf@tp{hypbmsec}
19 \itf@tp{environ}
20 \itf@tp{truncate}
21 \itf@tp{appendix}
22 \itf@tp{embedfile}
23 \if\relax@\optionlist{interfaces.sty}\relax % no option given
24   \ExecuteOptions\@declaredoptions          % all interfaces are active
25 \fi
26 \ProcessOptions
27 
```

21.2 Interfaces-base

```

28 <*base>
29 \ProvidesPackage{interfaces-base}
30   [2010/12/27 v2.6 Interfaces core system layer (FC)]
31 \RequirePackage{etex,ltxcmds,etoolbox,pgfkeys,scrlfile,infwarerr}
32 \AtEndOfPackage{%
33   \RequirePackage{interfaces-etoolbox}%
34   \RequirePackage{interfaces-scrlfile}%
35   \RequirePackage{interfaces-pgfkeys}%
36   \RequirePackage{interfaces-tikz}%
37   \RequirePackage{interfaces-umrand}% There is a \BeforePackage here !
38 }% AtEndOfPackage

```

\LoadInterface Mechanic of `interfaces`: each module is loaded \AfterPackage.

```

39 \newrobustcmd*\LoadInterface[1]{%
40   \@for\itf@tp:=#1\do{%
41     \expandafter\ifx\csname\ifcsname\itf@tp\endcsname
42       \itf@tp\else relax\fi\endcsname\marks
43     \RequirePackage{interfaces-marks}%
44     \else\expandafter\ifx\csname\ifcsname\itf@tp\endcsname
45       \itf@tp\else relax\fi\endcsname\LaTeX
46     \RequirePackage{interfaces-LaTeX}%
47   \else

```

```

48      \expandafter\itf@LoadInterface\expandafter{\itf@tp}%
49      \fi\fi}%
50 }% \LoadInterface
51 \@onlypreamble\LoadInterface
52 \newrobustcmd*\itf@LoadInterface[1]{\ltx@ifpackageloaded{interfaces-#1}
53   {}%
54   {\AfterPackage*{#1}{\RequirePackage{interfaces-#1}}}%
55 }% \itf@LoadInterface

56 \newrobustcmd*\itf@Check@Package[1]{\itf@pgfkeys\ltx@ifpackageloaded{#1}}
57   {}%
58   {\PackageError{interfaces-#1}%
59     {The package ‘#1’ must be loaded before ‘interfaces-#1’%
60      \MessageBreak Please refer to ‘interfaces.pdf’ document%
61      \MessageBreak for correct usage of the ‘interfaces’ package}\@ehd
62   }%
63 }% \itf@Check@Package

```

`\itf@AfterPackage` This is `\AfterPackage*` but it works inside document. `\ltxcmds` provides `\ltx@ifpackageloaded` to check if a package is loaded and – unlike L^AT_EX’s `\@ifpackageloaded` – this macro is not restricted to the document preamble.

```

64 \newrobustcmd*\itf@AfterPackage[1]{%
65   \csname\ltx@ifpackageloaded{#1}{\ltx@secondoftwo}{\AfterPackage}\endcsname{#1}%
66 }% \itf@AfterPackage

```

21.2.1 Interfaces patches

First a warning for loading `arydshln` after `colortbl`. Those packages does not interact properly.

```

67 \AfterPackage{colortbl}{%
68   {\ltx@ifpackageloaded{arydshln}%
69     {\@PackageWarningNoLine{interfaces-base}%
70       {The package arydshln should be loaded after colortbl%
71       \MessageBreak because both redefine \string@classz}}%
72   }%
73 }% AfterPackage colortbl

```

`\itf@preapptocmd` The need for this macro has rised to make `\itf@patch` simpler.

This is also more secure to do `\pretocmd` and `\apptocmd` in the same time, for the result (success or failure) is always the same for both.

```

74 \newrobustcmd*\itf@preapptocmd[5]{%
75   \pretocmd#1{#2}%
76   {\apptocmd#1{#3}%
77     {#4}%
78     {#5}}%
79   {#5}%
80 }% \itf@preapptocmd

```

`\itf@patch` `\itf@patch` is used as a prefix to `\pretocmd`, `\apptocmd` and `\patchcmd`.

It’s main purpose is to keep the information (`\PackageInfo`) in the case `\usepackage{interfaces}` has been put in a `\dump` format.

`\PackageInfo` is then deferred `\AtBeginDocument` time...

```

81 \def\itf@patch{@makeother\#\futurelet\itf@tp\itf@patch@}%
82 \def\itf@patch@{\csname itf@patch@%
83   \ifx\itf@tp\patchcmd inside\else%
84   \ifx\itf@tp\itf@preapptocmd inside\else%
85   boundary\fi\fi%
86   \endcsname%
87 }% \itf@patch@
88 \def\itf@patch@boundary#1#2#3#4#5{%
89   \catcode‘\#=6%
90   \let\itf@tp#2%

```

```

91   #1\itf@tp{#3}%
92   {\let#2\itf@tp
93     \AtBeginDocument{#4}% in order to keep this info even after \dump
94   }
95   {#5}%
96 }% \itf@patch@boundary (for \pretocmd and \apptocmd)
97 \def\itf@patch@inside#1#2#3#4#5#6{%
98   \catcode`\#=6%
99   \let\itf@tp#2%
100  #1\itf@tp{#3}{#4}%
101  {\ifx#1\patchcmd \itf@patch@inside@{#1}{#3}{#4}\fi
102    \let#2\itf@tp
103    \AtBeginDocument{#5}%
104  }
105  {#6}%
106 }% \itf@patch@inside (for \patchcmd)
107 \def\itf@patch@inside@#1#2#3{%
108   \loop
109     \let\@tempa \itf@tp
110     #1\itf@tp{#2}{#3}{}{}%
111     \unless\ifx \atempa\itf@tp
112     \repeat
113 }% \itf@patch@inside@

```

21.2.2 Helper macros

```

114 \newskip\itf@sk
115 \newmuskip\itf@musk
116 \newtoks\itf@tk
117 \newcount\itf@cnt

```

\itf@strip@at \itf@strip@at removes all until @ while \itf@strip@dot removes the fractional part.

\itf@strip@dot

```

118 %% H E L P E R M A C R O S
119 \begingroup\makeatother
120 \csgdef{\itf@strip@at}{#1@{}}
121 \endgroup
122 \def\itf@strip@dot{#1.#2}@nil{#1}

```

\itf@expandonceifcs This macro is used to expand or not a comma separated list.

USAGE: \itf@expandonceifcs\macrotest

\macrotest is expanded once and its first token τ is placed into a temporary macro \itf@tp.

If \macrotest equals \itf@tp: \macrotest is composed of that only token τ , otherwise we leave and do nothing.

Then we test \noexpand τ against \relax with \ifcat. If false τ cannot be a macro then we leave.

Then we test if τ is expandable: \ifx τ =\expandonce τ , τ is not expandable and we leave. (may be \def τ { τ } but it's not *expandable*!).

Otherwise, τ is expandable and \macrotest contains a single expandable token. Thus \macrotest shall be expanded once: \let\macrotest= τ .

```

123 \def\itf@expandonceifcs#1{\expandafter\itf@exp@ndonceifcs\expandafter#1#1Q\@nnil}
124 \def\itf@exp@ndonceifcs#1#2#3@nnil{%
125   \def\itf@tp{#2}%
126   \ifx\itf@tp#1%
127     \ifcat\relax\noexpand#2%
128       \expandafter\expandafter\expandafter\remove@to@nnil
129         \expandafter\ifx\expandafter#2#2\@nnil\else\@nnil\let#1#2\fi
130     \fi\fi
131 }% \itf@exp@ndonceifcs

```

21.2.3 Numbers

`\itf@number` `\itf@number` tries to get a number from its argument, and stores the result into the TeX count register given as first argument. Same syntax as L^AT_EX `\@defaultunits`.

A number is an integer that can be assigned to a count register, may be followed by one or more `\relax`.

```

132 %% G E T A N U M B E R (if possible)
133 %% USAGE: \itf@number\count<text>\relax\@nnil (like \@defaultunits)
134 \def\itf@number#1#2\relax\@nnil{%
135   \relax is to mimick the syntax of \@defaultunits
136   \afterassignment\itf@number@#1\#2\relax\itf@number@%
137 }
138 \def\itf@number@#1\relax#2\itf@number@{%
139   \ifblank{#1}{%
140     \csname ltx@\ifx@\let@token\itf@number@@%
141       first\else second\fi oftwo\endcsname}%
142   \ltx@secondoftwo
143 }% \itf@number@
144 \def\itf@number@@#1{\futurelet@\let@token\itf@number@@@}
145 \def\itf@number@@@{%
146   \csname \ifx@\let@token\relax \itf@number@@%
147   \else remove@to@nnil%
148   \fi \endcsname
149 }% \itf@number@@

```

21.2.4 Units and math units

`\itf@setlength` To be able to write: `\pgfkeys{key=\widthof{some text}}` (package `calc`), `interfaces-base` defines `\itf@setlength`:

```
150 \protected\def\itf@setlength#1#2{#1\glueexpr#2\relax}
```

`\itf@units@scale` For multiplication of units: `\glueexpr...\relax*\itf@units@scale<scaling factor>`!

The scaling factor can be a fraction or a real number. Real numbers are approximated by a fraction of 65536.

```

151 %% glue * 1.5 = glue * 98302 / 65536
152 %% glue * 3/2 = glue * 3/2
153 \def\itf@units@scale#1!{\itf@units@scale@normalize#1\@nnil\@nil}
154 \def\itf@units@scale@normalize#1/#2/#3@nil{%
155   \ifx#2\@nnil \number\dimexpr#1pt\relax/65536
156   \else#1/#2
157   \fi
158 }% \itf@units@scale@normalize
159 \def\itf@units@scaleNoCalc#1!{\itf@units@scale@normalizeNoCalc#1\@nnil\@nil}
160 \let\itf@units@scale@normalizeNoCalc \itf@units@scale@normalize

```

`\itf@setlength@calc` The version for the `calc` package. `calc` removes stretch and shrink components of glues when a scaling operation is done using `\real` or `\ratio`. This limitation is lifted for `key=value` assignments.

`\itf@units@scale@normalize@calc` The version of `\itf@units@scale@normalize` when using the `calc` package.

`\ratio` is used.

```

161 \protected\def\itf@setlength@calc#1#2{\begingroup
162   \let\calc@multiply@by@real \itf@calc@multiply@by@real
163   \let\calc@Adimen \itf@calc@Askip \let\calc@Bdimen \itf@calc@Bskip
164   \setlength{#1}{#2}%
165   \expandafter\endgroup\expandafter#1\the#1\relax
166 }% \itf@setlength@calc
167 \def\itf@units@scale@normalize@calc#1/#2/#3@nil{%
168   \ifx#2\@nnil 1*\ratio{\dimexpr#1pt\relax}\p@
169   \else 1*\ratio{\dimexpr#1pt\relax}/{\dimexpr#2pt\relax}%
170   \fi
171 }% \itf@units@scale@normalize@calc
172 \def\itf@calc@multiply@by@real#1{%
173   \calc@Bdimen\glueexpr\calc@B*\number\dimexpr#1pt\relax/65536\relax

```

```

174   \calc@B\calc@Bdimen
175 }% \itf@calc@multiply@by@real
176 \AfterPackage*{calc}{%
177   \newskip\itf@calc@Askip \newskip\itf@calc@Bskip
178   \let\itf@setlength \itf@setlength@calc
179   \let\itf@units@scale@normalize \itf@units@scale@normalize@calc
180 }% AfterPackage calc

```

\itf@strip@mu This is the equivalent of L^AT_EX's `\strip@pt` but for math units:

```

181 %% U N I T S F O R D I M E N S
182 \def\itf@strip@mu{\expandafter\itf@rem@mu\the}
183 {%
184 \@makeother\mathbin\@makeother\U
185 \lowercase{%
186   \gdef\itf@rem@mu#1.#2MU{\#1\ifnum#2>\z@.\#2\fi}%
187 }{%

```

\itf@units@pttomu 1*mu* is exactly 1*em*/18 where the *em* is taken from `\textfont2`.
\itf@units@mutopt Thus the scaling: `\fontdimen6\textfont2 * 100/1800`.

```

188 \% \skip = \itf@units@mutopt \muskip
189 \def\itf@units@pttomu#1{\gluetomu\glueexpr
190   \glueexpr#1\relax*1800%
191   \expandafter\itf@strip@dot\the\dimexpr
192     \fontdimen6\textfont2*100\relax.\@nil
193   \relax
194 }% \itf@units@pttomu
195 \def\itf@units@mutopt#1{\glueexpr
196   \mutoglue\muexpr#1\relax*
197   \expandafter\itf@strip@dot\the\dimexpr
198     \fontdimen6\textfont2*100\relax.\@nil/1800
199   \relax
200 }% \itf@units@mutopt

```

\itf@units@mu Forces a dimension to be expressed in terms of math units (*mu*), convert if needed. `\cftdotsep` in `\tocloft` is expressed in math units.

```

201 \% \muskip = \itf@units@mu{ \skip or \muskip or number}
202 {%
203 \@makeother\P\@makeother\T\@makeother\S\@makeother\B
204 \@makeother\C\@makeother\M\@makeother\D\@makeother\I\@makeother\N
205 \lowercase{%
206 \gdef\itf@units@mu#1#2{\begingroup
207   \edef\itf@tp{#2}%
208   \atfor\x:={PT}{pt}{PC}{pc}{IN}{in}{BP}{bp}%
209     {CM}{cm}{MM}{mm}{DD}{dd}{CC}{cc}{SP}{sp}\do{%
210     \expandtwoargs\in@\x\itf@tp\ifin@\@break@tfor\fi
211   }%
212   \expandafter\endgroup
213   \ifin@
214     #1=\itf@units@pttomu{#2}%
215   \else
216     \@defaultunits#1 #2mu\relax\@nnil
217   \fi
218 }% \itf@units@mu
219 }%

```

\itf@units@pt Conversely, forces a unit to be expressed in terms of points. Convert (from *mu*) if needed.

```

220 \% \skip = \itf@units@pt( \muskip or \skip or number )
221 {%
222 \@makeother\mathbin\@makeother\U
223 \lowercase{%
224 \gdef\itf@units@pt#1#2{\begingroup
225   \expandtwoargs\in@{MU}{#2}%
226   \ifin@\else\expandtwoargs\in@{mu}{#2}\fi

```

```

227   \expandafter\endgroup
228   \ifin@
229     #1=\itf@units@mutopt{#2}%
230   \else
231     \@defaultunits#1 #2pt\relax\@nnil
232   \fi
233 }}}} \itf@units@pt

```

21.2.5 Default color

A trick about `\normalcolor`, which refers to `\default@color`:

`\default@color` is not expressed in terms of a color model. Therefore, it is not possible to convert it into a universally correct parameter for the `\color` command. That's why, for example, `\bookmark` ignore blank parameter for `color`: because it can do nothing with `\normalcolor`!

Solution: if a color named “default” has been defined in the preamble, then it is used as the default color. Otherwise, we define, `\AtBeginDocument` a color named “default”

If `xcolor` is loaded, an assignment: `\colorlet{default}{.}` is made `\AtBeginDocument`. If only `color` is loaded, then `\color{default}` means `\color{\default@color}` (this syntax is not admissible with `xcolor`, but it works with `color`).

```

234 \AtBeginDocument{%
235   \ltx@ifpackageloaded{xcolor}
236   { \@ifundefinedcolor{default}
237     {\colorlet{default}{.}}
238     {} }%
239   \ltx@ifpackageloaded{color}
240   { \@ifundefined{\string\color @default}
241     {\expandafter\def\csname\string\color @default\endcsname{\default@color}}
242     {} }%
243   {} }%
244 }

245 </base>

```

21.3 ⇔ Interface LATEX

```

246 <*LaTeX>
247 \ProvidesPackage{interfaces-LaTeX}
248   [2010/12/27 v2.6 pgfkeys macros for LaTeX (FC)]
249 \RequirePackage{interfaces-base}
250 \itf@pgfkeys

```

21.3.1 \changefont

`\changefont` First use of `\pgfkeys` handlers `./store as glue in cs`, `./estore as dimen in cs...` The handlers create the `+` – and `*` keys automatically. As far as the `+` – and `*` keys invoke the main key, it is possible to `./append code` to the main key, without taking care of the `+` – and `*` keys.

The `./alias*` handler creates the aliases for the main key and the `+` – and `*` keys in one shot.

```

251 \newrobustcmd*\changefont[1]{\itf@pgfkeys
252   \pgfqkeys{/itf/LaTeX/changefont}{#1}\selectfont
253 }% \changefont
254 \AfterPackage*{hyperref}
255   {\pdfstringdefDisableCommands{\let\changefont\gobble}%

256 \pgfqkeys{/itf/LaTeX/changefont}{%
257   encoding/.code=\fontencoding{#1},encoding/.value required,%
258   enc/.alias=encoding,%
259   family/.code=\fontfamily{#1},family/.value required,%
260   fam/.alias=family,%
261   series/.code=\fontseries{#1},series/.value required,%
262   ser/.alias=series,%

```

```

263   shape/.code=\fontshape{#1},shape/.value required,%
264     sha/.alias=shape,%
265   size/.estore as dimen in cs=f@size,%
266   size/.append code=\set@fontsize\f@linespread\f@size\f@baselineskip,%
267     siz/.alias*=size,%
268   baselineskip/.estore as glue in cs=f@baselineskip,%
269   baselineskip/.append code=\set@fontsize\f@linespread\f@size\f@baselineskip,%
270     skip/.alias*=baselineskip,%
271   @linespread/.estore as dimen,%
272   @linespread/.append code=
273     \pgfkeysgetvalue{/itf/TeX/changefont/@linespread}\pgfkeys@temp
274     \ifx\pgfkeys@temp\empty \let\pgfkeys@temp \p@ \fi
275     \edef\f@linespread{\expandafter\rem@pt\pgfkeys@temp}%
276     \set@fontsize\f@linespread\f@size\f@baselineskip,%
277   @linespread/.initial=\p@,%
278   linespread/.code=@defaultunits\itf@sk #1pt\relax@nnil
279     \pgfkeysalso{/itf/TeX/changefont/@linespread=\itf@sk},%
280   linespread+/.code=@defaultunits\itf@sk #1pt\relax@nnil
281     \pgfkeysalso{/itf/TeX/changefont/@linespread+=\itf@sk},%
282   linespread-/.code=@defaultunits\itf@sk #1pt\relax@nnil
283     \pgfkeysalso{/itf/TeX/changefont/@linespread-=\itf@sk},%
284   linespread*/.alias=@linespread*,%
285     spread/.alias*=linespread,%
286 }% \pgfkeys definitions for \changefont
287 \itf@AfterPackage{color}{%
288   \pgfkeysifdefined{/itf/TeX/changefont/color/.@cmd}{}
289     {\pgfkeys{/itf/TeX/changefont/color/.code=\color{#1}}}}%
290 }% AfterPackage color
291 \itf@AfterPackage{xcolor}{%
292   \pgfkeysifdefined{/itf/TeX/changefont/color/.@cmd}{}
293     {\pgfkeys{/itf/TeX/changefont/color/.code=\color{#1}}}}%
294 }% AfterPackage xcolor

```

295 </TeX>

21.4 Interface TikZ ([pgfmanual](#))

```

296 {*tikz}
297 \ProvidesPackage{interfaces-tikz}
298   [2010/12/27 v2.6 load tikz libraries easily (FC)]
299 \RequirePackage{interfaces-base}
300 \itf@pgfkeys

```

21.4.1 Catcode verifications

Some languages (french) changes the catcode of punctuations symbols: : and ! are active, and this is bad for tikz calculations.

`\tikzset{every picture/.style=@makeother...}` solves the problem.

```

301 %% Some catcode checking
302 \AfterPackage*{tikz}{%
303   \AtEndPreamble{%
304     \tikzset{every picture/.add style={%
305       /utils/exec={\@makeother\! \@makeother\:}}}}%
306 }%
307 }%

```

21.4.2 List of pgf/TikZ packages and libraries

Record “modules” that must be loaded with `\usepackage`:

```

308 \newcommand*\alltikzpackages{%
309   tikz,pgf,pgfcore,pgfcalendar,pgffor,pgfkeys,pgfpages,pgfrcs,xxcolor,%
310   pgfplots,pgfsys,

```

```

311   pgfpict2e,fp%%,%           + fp required for fixedpointarithmetic
312 %%   tikzexternal,% needs special setup
313 %%   pgfmath,% pgfmath should not be loaded twice (\pgfmathdeclarefunction)
314 }% \alltikzpackages

```

`\allpgfmodules` Record modules that must be loaded with `\usepgfmodule`:

```

315 \newcommand*\allpgfmodules{datavisualization,%
316 decorations,%
317 matrix,%
318 oo,%
319 parser,%
320 plot,%
321 shapes,%
322 snakes%
323 %%sorting% bug in pgfmodulesorting.code.tex
324 }% \allpgfmodules

```

`\alltikzlibraries` Record modules that must be loaded with `\usetikzlibrary`.

If a module is not found in any of the three lists `\alltikzpackage`, `\allpgfmodules`, and `\alltikzlibraries` then `\usetikzlibrary` is used by default.

```

325 \newcommand*\alltikzlibraries{3d,%
326 arrows,%
327 automata,%
328 backgrounds,%
329 calc,%
330 calendar,%
331 chains,%
332 circuits,%
333 circuits.ee,%
334 circuits.ee.IEC,%
335 circuits.logic,%
336 circuits.logic.CDH,%
337 circuits.logic.IEC,%
338 circuits.logic.US,%
339 datavisualization,%
340 datavisualization.3d,%
341 datavisualization.barcharts,%
342 datavisualization.formats.functions,%
343 datavisualization.polar,%
344 datavisualization.sparklines,%
345 decorations,%
346 decorations.footprints,%
347 decorations.fractals,%
348 decorations.markings,%
349 decorations.pathmorphing,%
350 decorations.pathreplacing,%
351 decorations.shapes,%
352 decorations.text,%
353 er,%
354 %%external,% needs special setup
355 fadings,%
356 fit,%
357 fixedpointarithmetic,%
358 folding,%
359 fpu,%
360 graphs,%
361 graphs.basic,%
362 intersections,%
363 lindenmayersystems,%
364 matrix,%
365 mindmap,%
366 patterns,%
367 petri,%
368 plothandlers,%

```

```

369 plotmarks,%
370 positioning,%
371 scopes,%
372 shadings,%
373 shadows,%
374 shapes,%
375 shapes.arrows,%
376 shapes.callouts,%
377 shapes.gates.logic.IEC,%
378 shapes.gates.logic.US,%
379 shapes.geometric,%
380 shapes.misc,%
381 shapes.multipart,%
382 shapes.symbols,%
383 snakes,%
384 spy,%
385 svg.path,%
386 through,%
387 topaths,%
388 trees,%
389 turtle%
390 }% \alltikzlibraries

```

21.4.3 pgfkeys definitions for \usetikz

```

391 \pgfqkeys{/itf/tikz}{%
392   basic/.code={%
393     \usetikzlibrary{calc,% for coordinate calculations
394                 arrows,% for arrows
395                 backgrounds,% for framed
396                 scopes,% for scopes and shortcut : { [scope] ... }
397                 shapes,% for ellipse shapes of nodes
398     }%
399   },%
400   basic/.value forbidden,% \alltikzpackages
401   base/.alias=basic,%
402   all/.code={%
403     \expandafter\RequirePackage\expandafter{\alltikzpackages}%
404     \expandafter\usepgfmodule\expandafter{\allpgfmodules}% see pgfcorescopes.code.tex
405     \expandafter\usetikzlibrary\expandafter{\alltikzlibraries}% from: tikz.code.tex
406   },%
407   .unknown/.code=\itf@usetikzcode\pgfkeyscurrentname
408 }% \pgfkeys for \usetikz

```

\usetikz

```

409 \newrobustcmd*\usetikz[1]{%
410   \RequirePackage{tikz}%
411   \ifblank{#1}%
412     {\pgfqkeys{/itf/tikz}{all}}%
413     {\pgfqkeys{/itf/tikz}{#1}}%
414   \itf@pgfkeys% overwritten by \input primitive otherwise
415   \itf@usetikzsetup
416 }% \usetikz

```

\itf@usetikzsetup

```

417 \def\itf@usetikzsetup{%
418   \pgfkeysifdefined{/tikz/shorten}{}{%
419     \pgfkeys{/tikz/shorten/.alias=tikz/shorten >}%}
420   \pgfkeysifdefined{/tikz/shorten<}{}{%
421     \pgfkeys{/tikz/shorten</.alias=tikz/shorten <}%}
422   \pgfkeysifdefined{/tikz/shorten <>}{}{%
423     \pgfkeys{/tikz/shorten <>/style={/tikz/shorten >={##1},/tikz/shorten<={##1}}}}%
424   \pgfkeysifdefined{/tikz/shorten <>}{}{%

```

```
425     \pgfkeys{/tikz/shorten<>/.alias=tikz/shorten <>}}%
426 }%
```

\itf@usetikzcode For unknown keys: when a library is referenced.

```
427 \newrobustcmd*\itf@usetikzcode[1]{\begingroup
428   \edef\x{%
429     \noexpand\in@{,#1}{,\alltikzpackages,}%
430     \noexpand\ifin@ \noexpand\usepackage{#1}%
431     \noexpand\else
432       \noexpand\in@{,#1}{,\allpgfmodules,}%
433       \noexpand\ifin@ \noexpand\usepgfmodule{#1}%
434       \noexpand\else \noexpand\usetikzlibrary{#1}\% \usetikzlibrary by default
435       \noexpand\fi
436     \noexpand\fi
437   }\expandafter\endgroup\x
438 }% \itf@usetikzcode
439 
```

21.5 Interface titlesec : \sectionformat

```
440 {*titlesec}
441 \ProvidesPackage{interfaces-titlesec}
442 [2010/12/27 v2.6 pgfkeys macros for titlesec (FC)]
443 \RequirePackage{interfaces-base}
444 \RequirePackage{auxhook}\% <fix for \ttlp@append (odd/even pages)>
445 \RequirePackage{getttitlestring}
446 \itf@Check@Package{titlesec}
```

21.5.1 titlesec modifications

```
447 %% titlesec "patches"
448 \robustify\titleclass \robustify\titleformat \robustify\titlespacing
449 \AddLineBeginMainAux{%
450   \string\providecommand\string\ttlp@append[2]\{}%
451 }
452 \BeforeFile*{ttlkeys.def}\%
453   \expandafter\edef\csname itf@ttlkeys.def\endcsname{%
454     \catcode32 \the\catcode32\relax
455     \catcode37 \the\catcode37\relax
456     \endlinechar\the\endlinechar\relax\%
457     \catcode37 14\relax % percent=comment (inside .dtx files...)
458     \catcode32 9\relax % ignore spaces
459     \endlinechar\m@ne % and line feeds
460 }
461 \AfterFile*{ttlkeys.def}\{\csname itf@ttlkeys.def\endcsname\}
```

\ttlh@runinifempty A “new” shape which behaves like **hang** but no paragraph break is inserted if the title is empty. Hence the name: **runinifempty**.

```
462 %-----%
463 % runinifempty => like hang but no paragraph break (like runin) if title is empty
464 \def\ttlh@runinifempty{%
465   \csname ttlh@\ifx\ttl@savetitle\@empty runin\else hang\fi\endcsname
466 }% \ttlh@runinifempty
```

21.5.2 \sectionformat

```
\sectionformat \sectionformat\subsection* [<shape>]{<key=value>}
467 \newrobustcmd*\sectionformat[1]{\ifstar
468   {\itf@sectionformat{#1}{/*}}
469   {\itf@sectionformat{#1}{}}\%
470 }% \sectionformat
471 \def\itf@sectionformat#1#2{\@testopt
```

```

472     {\expandafter\expandafter\expandafter%
473      \itf@sectionf@rmat\expandafter\gobble\string#1@nil{#2}}{}%
474 }% \sectionformat
475 \def\itf@sectionf@rmat#1@nil#2[#3]{\itf@pgfkeys@ifblank{#3}%
476   {\pgfqkeys{/itf/titlesec/#1#2}%
477    {/itf/titlesec/@setup={#1}{#2},#4,/itf/titlesec/@update={#1}{#2}}}%
478   {\pgfqkeys{/itf/titlesec/#1#2}%
479    {/itf/titlesec/@setup={#1}{#2},#4,shape={#3},/itf/titlesec/@update={#1}{#2}}}%
480 }% \itf@ttl@sectionformat

```

`\itf@ttl@setup` Macros used to retrieve the values set by `\titleformat` and/or `\titlespacing`. This allows to use
`\itf@ttl@getformat` `\sectionformat` concurrently with `\titleformat` and `\titlespacing` provided by `titlesec`.

```

\itf@ttl@getspacing 481% private keys @setup and @update
482 \pgfqkeys{/itf/titlesec}{%
483   @setup/.code 2 args=\itf@ttl@setup[#1]{#2},%
484   @update/.code 2 args=\expandafter\itf@ttl@update
485   \csname#1\endcsname{/itf/titlesec/#1}{#2}%
486 }
487 \newrobustcmd*\itf@ttl@setup[2]{%
488   \pgfkeysifdefined{/itf/titlesec/#1#2/shape}{}%
489   {\itf@ttl@newsection{#1}{#2}}%
490   \ifcsundef{ttls@#1#2}{}% no previous spacing settings
491   {\expandafter\expandafter\expandafter%
492    \itf@ttl@getspacing\csname ttls@#1#2\endcsname{/itf/titlesec/#1#2}}%
493   \ifcsundef{ttlf@#1#2}{}% no previous format settings
494   {\ifttl@explicit
495    \expandafter\expandafter\expandafter%
496    \itf@ttl@getformat\csname ttlf@#1#2\endcsname{/itf/titlesec/#1#2}%
497    \else
498    \expandafter\expandafter\expandafter%
499    \itf@ttl@getformat\csname ttlf@#1#2\endcsname{/itf/titlesec/#1#2}%
500    \fi}%
501 }% \itf@ttl@setup
502 \def\itf@ttl@setkey#1#2{\begingroup
503   \def\x{\pgfkeysvalueof{#1}}\def\y{#2}%
504   \expandafter\endgroup\ifx\x\y\else
505     \pgfkeyssetvalue{#1}{#2}\fi
506 }% \itf@ttl@setkey
507 \def\itf@ttl@getspacing#1#2#3#4#5#6{%
508   \itf@ttl@setkey{#6/left}{#1}%
509   \itf@ttl@setkey{#6/right}{#2}%
510   \itf@ttl@setkey{#6/top}{#3}%
511   \itf@ttl@setkey{#6/bottom}{#4}%
512   \itf@ttl@setkey{#6/indent first}{#5}%
513 }% \itf@ttl@getspacing
514 \def\itf@ttl@getformat#1#2{%
515   \csname ltx@\ifx\pgfkeysvalueof#1first\else second\fi oftwo\endcsname
516   {\itf@ttl@get@format{}}
517   {\itf@ttl@get@format{#1}{#2}}%
518 }% \itf@ttl@getformat
519 \def\itf@ttl@get@format#1#2#3#4#5#6#7{%
520   \if\relax\detokenize{#1}\relax\else
521     \pgfkeysEsetvalue{#7/shape}{\expandafter\itf@strip@at\string#1}%
522   \fi
523   \itf@ttl@setkey{#7/font}{#2}%
524   \itf@ttl@setkey{#7/label}{#3}%
525   \itf@ttl@setkey{#7/sep}{#4}%
526   \itf@ttl@setkey{#7/before}{#5}%
527   \itf@ttl@setkey{#7/after}{#6}%
528 }% \itf@ttl@get@format

```

`\itf@ttl@update` This macros plugs the values set before with `\pgfkeys` into the special macros `\ttlf@<section>` (for format) and `\ttls@<section>` (for spacing).

```
529 \newrobustcmd*\itf@ttl@update[3]{%
```

```

530   \ifblank{#3}
531     {\ttl@getkeys{#1}{titlesec}}
532     {\ttl@getkeys{name=#1, numberless}{titlesec}}%
533   \pgfkeysgetvalue{#2#3/shape}\itf@tp
534   \ifcsundef{ttlh@\itf@tp}
535     {\begingroup \makeatletter
536      \InputIfFileExists{\itf@tp.tss}{}{%
537        \PackageError{titlesec}{Unknown shape}%
538          {Shapes are defined in files with extension tss\MessageBreak
539            Either you have misspelled the shape\MessageBreak
540            or there is no a \itf@tp.tss file}}%
541      \endgroup
542    }%
543   \ifttl@explicit \begingroup
544     \pgfkeysgetvalue{#2#3/before}\pgfkeys@temp
545     \toks@\expandafter{\pgfkeys@temp}%
546     \edef\x{\endgroup\def\noexpand\itf@tp##1{\the\toks@}}%
547     \x \pgfkeyslet{#2#3/before@cmd}\itf@tp
548   @temptokena{\csdef{ttl@#1}{%
549     \the\@temptokena{%
550       \pgfkeysvalueof{#2#3/shape@cs}%
551       {\pgfkeysvalueof{#2#3/font}}%
552       {\pgfkeysvalueof{#2#3/label}}%
553       {\pgfkeysvalueof{#2#3/sep}}%
554       {\pgfkeysvalueof{#2#3/before@cmd}##1}%
555       {\pgfkeysvalueof{#2#3/after}}}}%
556     \else\@temptokena{\csdef{ttl@#1}{%
557       \the\@temptokena{%
558         \pgfkeysvalueof{#2#3/shape@cs}%
559         {\pgfkeysvalueof{#2#3/font}}%
560         {\pgfkeysvalueof{#2#3/label}}%
561         {\pgfkeysvalueof{#2#3/sep}}%
562         {\pgfkeysvalueof{#2#3/before}}%
563         {\pgfkeysvalueof{#2#3/after}}}}%
564     \fi
565     \csname ttl@compat\endcsname
566     \ttl@spacing@ii
567       {\pgfkeysvalueof{#2#3/indent first}}%
568       {\pgfkeysvalueof{#2#3/left}}%
569       {\pgfkeysvalueof{#2#3/top}}%
570       {\pgfkeysvalueof{#2#3/bottom}}%
571       [\pgfkeysvalueof{#2#3/right}]%
572 }% \itf@ttl@update

```

`\itf@ttl@titleclass` This macro calls `\titleclass` to either:

- change the class of an existing sectionning command;

or:

- create a new sectionning command.

```

573 %-----%
574 %%  t i t l e c l a s s
575 \newrobustcmd*\itf@ttl@titleclass{\ifstar
576   {\itf@ttl@titleclass@i*}
577   {\itf@ttl@titleclass@i{}}}%
578 }% \itf@ttl@titleclass
579 \def\itf@ttl@titleclass@i#1#2[#4]#5@nil{\ifblank{#4}
580   {%
581     \expandafter\titleclass\csname#2\endcsname{#3}%
582   }
583   {%
584     \expandafter\titleclass\csname#2\endcsname{#3}[{#4}]% #4 = sup-level
585     \if\relax#1\relax % not starred
586       \expandafter\expandafter\expandafter%
587         \itf@ttl@titleclass@ii\expandafter@gobble\string#4\relax{#2}%
588   }%
589 \fi

```

```

589    }%
590    \expandafter\providecommand\csname l@#2\endcsname[2]{}}% no entry in toc
591 }% \itf@ttl@titleclass@i
592 \def\itf@ttl@titleclass@ii#1\relax#2{%
593   \ifcsundef{c@#2}
594     {\newcounter{#2}[#1]}% creates counter for #1
595     {\@addtoreset{#2}{#1}}%
596   \csxdef{the#2}{%
597     \expandafter\noexpand\csname the#1\endcsname
598     .\noexpand\@arabic\expandafter\noexpand\csname c@#2\endcsname
599   }% \the#2
600   \ifcsundef{l@#1}{}{%
601     \expandafter\providecommand\csname l@#2\endcsname[2]{%
602       \csname l@#1\endcsname{##1}{##2}}%
603     }% \l@#2 = \l@newsection is (by default) the same as \l@sup-level
604   }%
605 }% \itf@ttl@titleclass@ii

```

\itf@ttl@createsection If \sectionformat is invoked with an unknown sectionning command (*i.e.* an undefined macro as first argument) then \itf@ttl@createsection is expanded in order to:

- 1) create the new sectionning command with \titleclass,
- 2) create the pgfkeys keys for this section,
- 3) do some additionnal setup if `interfaces-hypbmsec` and/or `interfaces-tocloft` are loaded.

```

606 %-----%
607 %% Creating new sections
608 \def\itf@ttl@createsection#1{%
609   \expandafter\titleclass\csname#1\endcsname{straight}[*]%
610   \itf@AfterPackage{hypbmsec}{%
611     \AfterPreamble{%
612       \csletcs{hbs@#1}{#1}%
613       \csxdef{#1}{\noexpand\hbs@do\expandafter\noexpand\csname hbs@#1\endcsname}%
614     }%
615   }%
616   \itf@AfterPackage{interfaces-tocloft}{%
617     \itf@cft@newsection{toc}{#1}%
618   }%
619   \itf@AfterPackage{interfaces-hypbmsec}{%
620     \expandafter\let\csname last#1title\endcsname\relax
621     \expandafter\let\csname last#1bookmark\endcsname\relax
622     \expandafter\let\csname last#1toc\endcsname\relax
623   }%
624 }% \itf@ttl@createsection

```

21.5.3 Keys for \sectionformat

\itf@ttl@newsection This macro creates the pgfkeys keys for each sectionning command.

The pgfkeys keys are not created at once, but only when \sectionformat is invoked first.

```

625 %-----%
626 %% Defining keys for each section
627 \def\itf@ttl@newsection#1#2{%
628   #1=section, #2=/* or empty
629   \ifcsundef{#1}{%
630     \itf@ttl@createsection{#1}%
631   }%
632   \ifblank{#2}{%
633     \pgfqkeys{/itf/titlesec/#1}{%
634       these keys do not exist for starred sections
635       class/.code=\itf@ttl@titleclass{#1}##1[]\@nil,% with \@addtoreset
636       class/.value required,% if sup-level is given
637       class*/.code=\itf@ttl@titleclass*{#1}##1[]\@nil,% without \@addtoreset
638       class*/.value required,% even if sup-level is given
639       shape/.initial=,% initial shapes set by titlesec.sty for not numbered sections.
640       the#1/.store in cs=the#1,%
641     }%
642   }%
643 }

```

```

639         the#1/.default=\pgfkeysvalueof{/itf/titlesec/#1/label},%
640     }
641     % starred section: unless titlesec, we do not put the default shape to hang,
642     %\pgfkeysifdefined{/itf/titlesec/#1/shape}{but to the shape of the not starred
643     % {\pgfkeysgetvalue{/itf/titlesec/#1/shape}\itf@tp}%
644     % \def\itf@tp{hang}}%
645     \pgfkeyslet{/itf/titlesec/#1#2/shape}\itf@tp%
646   }%
647 \pgfqkeys{/itf/titlesec/#1#2}{%
648   shape/.value required,%
649   shape@cs/.initial=%
650     \csname ttlh@\pgfkeysvalueof{/itf/titlesec/#1#2/shape}\endcsname,% constant
651     shape@cs/.is private=interfaces-titlesec,% constant
652   font/.store as code,%
653     format/.alias*=font,%
654   label/.store as code,%
655   sep/.estore as glue,%
656     labelsep/.alias*=sep,%
657   before/.store as code,%
658     before@cmd/.is private=interfaces-titlesec,%
659   after/.store as code,%
660   indent first/.true=\pgfkeyssetvalue{/itf/titlesec/#1#2/indent first}{\@ne},%
661   indent first/.false=\pgfkeyssetvalue{/itf/titlesec/#1#2/indent first}{\z@},%
662     indentfirst/.alias=indent first,%
663     indent first=true,%
664   left/.estore as glue,%
665   top/.estore as glue,%
666   bottom/.estore as glue,%
667   right/.estore as glue,%
668   break/.code=\ifblank{##1}%
669     {\cslet{\#1break}{\empty}}
670     {\protected\csdef{\#1break}{##1}},%
671   break*/.code=\protected\csappto{\#1break}{##1},%
672   break+/.alias=break*,%
673   mark/.code=\ifblank{##1}%
674     {\cslet{\#1mark}{\gobble}}
675     {\protected\long\csdef{\#1mark}####1{##1}},%
676   mark/.default=,%
677 }% \pgfkeys
678 \ifblank{##2}{% no default marks for stared (numberless) \section*
679 \itf@AfterPackage{interfaces-marks}{% requires interfaces-marks
680   \pgfkeys{/itf/titlesec/#1/mark/.default=\marksthe{##1}{##1}}%
681 }% AfterPackage interfaces-mark
682 \itf@AfterPackage{interfaces-bookmark}{% requires interfaces-bookmark
683 \pgfqkeys{/itf/titlesec/#1}{% does not exist for starred sections
684   bookmark/.is multiple choice,%
685   bookmark/level/.estore in cs=toclevel@##1,% depth of the bookmark
686     bookmark/level/.default=\ifcsvvoid{ttll@##1}{99}{\csuse{ttll@##1}},%
687     bookmark/level/.blank=\ifcsvvoid{toclevel@##1}{%
688       \ifcsvvoid{ttll@##1}{99}{\csuse{ttll@##1}}{\csuse{toclevel@##1}},%
689     bookmark/toclevel/.alias=bookmark/level,%
690   bookmark/bold/.code=\ifblank{##1}%
691     {\pgfkeyslet{/itf/titlesec/#1/bookmark/bold}{\empty}}
692     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/bold}{bold=##1},}%,%
693     bookmark/bold/.default=true,%
694   bookmark/bold*/.code=\ifblank{##1}%
695     {\pgfkeyslet{/itf/titlesec/#1/bookmark/bold*}{\empty}}
696     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/bold*}{bold*=##1},}%,%
697     bookmark/bold*/.default=true,%
698   bookmark/italic/.code=\ifblank{##1}%
699     {\pgfkeyslet{/itf/titlesec/#1/bookmark/italic}{\empty}}
700     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/italic}{italic=##1},}%,%
701     bookmark/italic/.default=true,%
702   bookmark/italic*/.code=\ifblank{##1}%
703     {\pgfkeyslet{/itf/titlesec/#1/bookmark/italic*}{\empty}}

```

```

704     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/italic}{italic*=\#\#1,\}},%
705     bookmark/italic*/.default=true,%  

706     bookmark/color/.code=\ifblank{\#\#1}{%
707         {\pgfkeysset{/itf/titlesec/#1/bookmark/color}\@empty}%
708         {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/color}{color=\#\#1,\}},%
709         bookmark/color/.default=%
710         bookmark/color*/.code=\ifblank{\#\#1}{%
711             {\pgfkeysset{/itf/titlesec/#1/bookmark/color*}\@empty}%
712             {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/color*}{color*=\#\#1,\}},%
713             bookmark/color*/.default=%
714             bookmark/on/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on}{disable=false,\},%
715             bookmark/on/.value forbidden,%  

716             bookmark/off/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on}{disable=true,\},%
717             bookmark/off/.value forbidden,%  

718             bookmark/none/.alias=bookmark/off,%  

719             bookmark/on*/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on*}{disable*=false,\},%
720             bookmark/off*/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on*}{disable*=true,\},%
721             bookmark/none*/.alias=bookmark/off*,%
722             bookmark/disable*/.alias=bookmark/off*,%
723             bookmark/style/.code=\ifblank{\#\#1}{%
724                 {\pgfkeysset{/itf/titlesec/#1/bookmark/style}\@empty}%
725                 {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/style}{style=\#\#1,\}},%
726                 bookmark/style/.default=%
727                 bookmark/.unknown/.code=\pgfkeysEsetvalue{/itf/titlesec/#1/bookmark/other}%
728                     {\pgfkeyscurrentname=\#\#1\},%
729                 bookmark/bold/.initial=%  

730                 bookmark/bold*/.initial=%  

731                 bookmark/italic/.initial=%  

732                 bookmark/italic*/.initial=%  

733                 bookmark/color/.initial=%  

734                 bookmark/color*/.initial=%  

735                 bookmark/on/.initial=%  

736                 bookmark/on*/.initial=%  

737                 bookmark/style/.initial=%
738 }%\ AfterPackage*\{interfaces-bookmark}
739 }% not starred (numberless) \section
740 {}% starred/numberless \section*: do not create the keys
741 }% \itf@ttl@newsection

```

\addcontentsline (patch)

This modification of `\addcontentsline` occurs only if both `interfaces-bookmark` and `interfaces-titlesec` are loaded. Implementation of the `bookmark` key and sub-keys of `\sectionformat` for each sectionning command.

```

742 %% bookmarks styles for sectionning commands:
743 \@makeother\#
744 \itf@AfterPackage{interfaces-bookmark}{%
745   \itf@patch\itf@preapptocmd\addcontentsline % #2=section name in
746   {\begingroup\itf@ttl@BKM{/itf/titlesec/#2/bookmark}}% \addcontentsline
747   {\endgroup}
748   {\@PackageInfoNoLine{interfaces-titlesec}
749    {Command \string\addcontentsline\space successfully patched!}}
750   {\@PackageWarningNoLine{interfaces-titlesec}
751    {Patch of \string\addcontentsline\space failed
752     \MessageBreak bookmark-style specs for sections
753     \MessageBreak will be unavailable}}%
754 }% AfterPackage interfaces-bookmark
755 \catcode`\#=6
756 \def\itf@ttl@BKM#1{%
757   \pgfkeysifdefined{#1/.@cmd}{%
758     {%
759       \edef\itf@tp{%
760         \do nothing if the keys are not set (shortcut)
761         \pgfkeysvalueof{#1:bold}%
762         \pgfkeysvalueof{#1:bold*}%
763         \pgfkeysvalueof{#1:italic}%
764         \pgfkeysvalueof{#1:italic*}%
765       }%
766     }%
767   }%
768 }
```

```

764     \pgfkeysvalueof{#1/color}%
765     \pgfkeysvalueof{#1/color*}%
766     \pgfkeysvalueof{#1/on}%
767     \pgfkeysvalueof{#1/on*}%
768     \pgfkeysvalueof{#1/style}%
769     \ifx\itf@tp\@empty
770     \else
771         \global\pgfkeyslet{#1/bold*}\@empty
772         \global\pgfkeyslet{#1/italic*}\@empty
773         \global\pgfkeyslet{#1/color*}\@empty
774         \global\pgfkeyslet{#1/on*}\@empty
775         \expandafter\bookmarksetup\expandafter{\itf@tp}%
776         \pgfkeysifdefined{/itf/bookmark/*}%
777             {%
778                 \% \bookmarksetup{color*=} overwrites \sectionformat...{bookmark/color*=}
779                 \pgfkeysgetvalue{/itf/bookmark/*}\itf@tp
780                 \expandafter\bookmarksetup\expandafter{\itf@tp}%
781             }%
782     \fi%
783     {}%
784 }% \itf@ttl@BKM

```

21.5.4 `\bottomtitles`

`\bottomtitles` This is a “skip command”: it can be assigned a number or a skip.

```

785 %-----%
786 %% b o t t o m t i t l e s
787 \newrobustcmd*\bottomtitles{\@ifnextchar\bgroup
788     \itf@setbottomtitlespace
789     {\afterassignment\itf@set@bottomtitle@space\itf@sk}%
790 }% \bottomtitles
791 \def\itf@setbottomtitlespace#1{%
792     \@defaultunits\itf@sk#1\textheight\relax\@nnil
793     \itf@set@bottomtitle@space
794 }% \itf@setbottomtitlespace
795 \def\itf@set@bottomtitle@space{\xdef\bottomtitlespace{\the\dimexpr\itf@sk}%
796 </titlesec>

```

21.6 Interface `hyperref`

```

797 (*hyperref)
798 \ProvidesPackage{interfaces-hyperref}
799   [2010/12/27 v2.6 provides raised hyperdef (FC)]
800 \RequirePackage{interfaces-base}
801 \itf@Check@Package{hyperref}

```

`\raisedhyperdef` The anchor is made with `\Hy@raisedlink`. The raised amount is used in `\HyperRaiseLinkHook`:

```

802 \newrobustcmd\raisedhyperdef[4][]{\begingroup
803     \ifblank{#1}{}
804     {}%
805     {}%
806     \expandafter\def\expandafter\HyperRaiseLinkHook\expandafter{%
807         \expandafter\def\expandafter\HyperRaiseLinkHook
808             \expandafter{\HyperRaiseLinkHook}% restore current definition
809             \HyperRaiseLinkHook
810             \advance\noexpand\HyperRaiseLinkLength by\dimexpr#1\relax\relax}%
811     }%
812     \ifx \\#2\\def \Hy@AnchorName {#3}\else
813         \def \Hy@AnchorName {#2.#3}%
814     \fi
815     \ltx@leavevmode
816     \Hy@raisedlink{\hyper@anchorstart{\Hy@AnchorName}\hyper@anchorend}%

```

```

817 \endgroup
818 #4%
819 }% \raisedhyperdef
820{/hyperref}

```

21.7 Interface bookmark

```

821 {*bookmark}
822 \ProvidesPackage{interfaces-bookmark}
823 [2010/12/27 v2.6 more control on bookmarks (FC)]
824 \RequirePackage{interfaces-base}
825 \RequirePackage{interfaces-hyperref}
826 \RequirePackage{refcount}
827 \itf@Check@Package{bookmark}

```

\`bookmark (patch) Implementation of `disable`, and starred keys `*` (previous values must be restored after `\write`):

```

828 \expandafter\itf@patch\expandafter\patchcmd\csname\string\bookmark\endcsname
829 {\immediate\write\mainaux}{\itf@BKM@write}
830 {\@PackageInfoNoLine{interfaces-bookmark}
831 {Command \expandafter\string\csname\string\bookmark\endcsname\space
832 successfully patched!}}
833 {\@PackageWarningNoLine{interfaces-bookmark}
834 {Command \expandafter\string\csname\string\bookmark\endcsname\space
835 could not be patched\MessageBreak
836 Loading of interfaces-bookmark is canceled}%
837 \endinput}

```

21.7.1 star keys* definitions

```

838 \global\let\itf@BKM@enable@ne@0= disable*, 1=enable, \@M = disable
839 \pgfqkeys{/itf/bookmark}{%
840   disable/.true=\global\let\itf@BKM@enable@\M,%
841   disable/.false=\global\let\itf@BKM@enable@\ne,%
842   disable*/.true=\global\let\itf@BKM@enable@z@,%
843   disable*/.false=\global\let\itf@BKM@enable@\ne,%
844   atend/.true=\ifnum\itf@BKM@enable>0\edef\itf@BKM@enable{\number-\itf@BKM@enable}\fi,%
845   atend/.false=\ifnum\itf@BKM@enable<0\edef\itf@BKM@enable{\number-\itf@BKM@enable}\fi,%
846   bold*/.true=\pgfqkeys{/itf/bookmark}{@save={bold=true}},%
847   bold*/.false=\pgfqkeys{/itf/bookmark}{@save={bold=false}},%
848   bold*/@blank/.code=\pgfqkeys{/itf/bookmark/@restore}{bold},%
849   bold*/.blank=@blank,%
850   italic*/.true=\pgfqkeys{/itf/bookmark}{@save={italic=true}},%
851   italic*/.false=\pgfkeys{/itf/bookmark}{@save={italic=false}},%
852   italic*/@blank/.code=\pgfqkeys{/itf/bookmark/@restore}{italic},%
853   italic*/.blank=@blank,%
854   color*/.code={\ifblank{#1}{%
855     \pgfqkeys{/itf/bookmark/@restore}{color}}{%
856     \pgfqkeys{/itf/bookmark}{@save={color={#1}}}}}}%
857 },%
858 @save/.code={%
859   \pgfkeysifdefined{/itf/bookmark/@save/color}{}{%
860     \pgfkeyslet{/itf/bookmark/@save/color}\BKM@color}%
861   \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{}{%
862     \pgfkeyslet{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}%
863   \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}{}{%
864     \pgfkeysEsetvalue{/itf/bookmark/@save/grouplevel}{\the\currentgrouplevel}}%
865   \pgfkeysaddvalue{/itf/bookmark/*}{}{, #1}{%
866     \bookmarksetup{#1}}%
867 },%
868 @restore/color/.code=%
869   \pgfkeysifdefined{/itf/bookmark/@save/color}{%
870     \pgfkeysgetvalue{/itf/bookmark/@save/color}\BKM@color}{}%
871   {}%

```

```

872  @restore/.code=%
873    \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}
874      {\pgfkeysgetvalue{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}
875      {}%
876  @restore/.code=%
877    \pgfkeysifdefined{/itf/bookmark/@save/color}
878      {\pgfkeysgetvalue{/itf/bookmark/@save/color}\BKM@color}
879      {}%
880    \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}
881      {\pgfkeysgetvalue{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}
882      {}%
883    \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}
884      {% shortcut: \bookmarksetup{bold*} called inside main group (level 0)
885       \ifnum\pgfkeysvalueof{/itf/bookmark/@save/grouplevel}=`z@
886         \global\pgfkeyslet{/itf/bookmark/@save/FontStyle}@undefined
887         \global\pgfkeyslet{/itf/bookmark/@save/color}@undefined
888         \global\pgfkeyslet{/itf/bookmark/@save/grouplevel}@undefined
889         \global\pgfkeyslet{/itf/bookmark/*}@undefined
890         \global\let\BKM@color\BKM@color
891         \global\let\BS@BKM@FontStyle\BS@BKM@FontStyle
892       \else % restore after each group as needed
893         \pgfkeyslet{/itf/bookmark/@save/FontStyle}@undefined
894         \pgfkeyslet{/itf/bookmark/@save/color}@undefined
895         \pgfkeyslet{/itf/bookmark/*}@undefined
896         \ifnum\currentgrouplevel>z@
897           \aftergroup\itf@BKM@restore
898         \else
899           \pgfkeyslet{/itf/bookmark/@save/grouplevel}@undefined
900       \fi\fi
901     }
902     {}%
903   }%
904 @restore/bold/.code={% bold is "hidden" in \BS@BKM@FontStyle
905   \begingroup
906     \edef\itf@tp{\bookmarkget{bold}}%
907     \pgfqkeys{/itf/bookmark/@restore}{FontStyle}%
908     \count@=\bookmarkget{bold}%
909     \edef\itf@tp{\endgroup
910       \ifnum\itf@tp=\count@
911         \else \noexpand\bookmarksetup{%
912           bold\ifcase\count@ =false\fi}\fi
913       }\itf@tp}%
914 @restore/italic/.code={% italic is "hidden" in \BS@BKM@FontStyle
915   \begingroup
916     \edef\itf@tp{\bookmarkget{italic}}%
917     \pgfqkeys{/itf/bookmark/@restore}{FontStyle}%
918     \count@=\bookmarkget{italic}%
919     \edef\itf@tp{\endgroup
920       \ifnum\itf@tp=\count@
921         \else \noexpand\bookmarksetup{%
922           italic\ifcase\count@ =false\fi}\fi
923       }\itf@tp}%
924 }
925 \def\itf@BKM@restore{%
926   \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}
927     {\pgfkeysalso{/itf/bookmark/@restore}}
928     {}%
929 }% \itf@BKM@restore
930 %% Plug the BKM keys in the interfaces pgfkeys (not efficient but quickly written...)
931 \define@key{BKM}{disable}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/disable={#1}}}
932 \define@key{BKM}{disable*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/disable*={#1}}}
933 \define@key{BKM}{bold*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/bold*={#1}}}
934 \define@key{BKM}{italic*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/italic*={#1}}}
935 \define@key{BKM}{color*}[]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/color*={#1}}}
936 \define@key{BKM}{atend}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/atend={#1}}}%

```

check if the bookmark is disable

2) restore previous values in case starred* keys have been set.

```

937 \def\itf@BKM@write#1{%
938   \aftergroup\itf@BKM@restore
939   \ifcase\itf@BKM@enable
940     \global\let\itf@BKM@enable\@ne
941   \or
942     \immediate\write\@mainaux{\#1}%
943   \else
944     \ifcase -\itf@BKM@enable % (case -0 = 0 tested before)
945       \or % (case -1 = enable but At End)
946         \let\itf@BKM@enable\@ne
947         \edef\itf@tp{\immediate\write\@mainaux{\#1}}%
948         \expandafter\BookmarkAtEnd\expandafter{\itf@tp}%
949       \fi
950   \fi
951 }% \itf@BKM@write

```

21.7.2 ref and pageref keys

\itf@BKM@pageref

```

952 %% pageref= key for \bookmark (using refcount : \getpagerefnumber)
953 \def\itf@BKM@pageref[#1]{%
954   \expandafter\itf@BKM@@pageref
955   \csname r@\#1\endcsname{#1}%
956 }% \itf@BKM@pageref
957 \def\itf@BKM@@pageref#1#2{%
958   \ifx#1\relax
959     \protect\G@refundefinedtrue
960     \@latex@warning{%
961       Hyper reference '#2' on page \thepage \space undefined
962       \MessageBreak for bookmark destination page
963     }%
964   \fi
965   \edef\BKM@page{\getrefbykeydefault{#2}{page}{1}}%
966 }% \itf@BKM@pageref
967 \define@key{BKM}{pageref}{%
968   \Hy@babelnormalise\itf@BKM@pageref[{"#1"}]

```

\itf@BKM@ref

```

969 %% ref= key for \bookmark
970 \def\itf@BKM@ref[#1]{%
971   \expandafter\itf@BKM@@ref
972   \csname r@\#1\endcsname{#1}%
973 }% \itf@BKM@ref
974 \def\itf@BKM@@ref#1#2{%
975   \ifx#1\relax
976     \protect\G@refundefinedtrue
977     \@latex@warning{%
978       Hyper reference '#2' on page \thepage \space undefined
979       \MessageBreak for bookmark destination
980     }%
981     \def\BKM@dest{??}%
982   \else
983     \edef\BKM@dest{\expandafter\@fourthoffive#1\@empty\@empty}%
984   \fi
985 }% \BKM@ref
986 \define@key{BKM}{ref}{%
987   \Hy@babelnormalise\itf@BKM@ref[{"#1"}]
988 </bookmark>

```

21.8 + Interface `hypbmsec`

21.8.1 `interfaces-hypbmsec`

```

989 /*hypbmsec*/
990 \ProvidesPackage{interfaces-hypbmsec}
991   [2010/12/27 v2.6 lastsection,title,anchor, etc. (FC)]
992 \RequirePackage{interfaces-base}
993 \RequirePackage{gettitlestring}
994 \itf@Check@Package{hypbmsec}

\hbs@do (patch)

995 \def\itf@hbs@do#1{%
996   \xdef\itf@hbs@currentname{\expandafter\itf@strip@at\string#1}%
997   \let\hbs@tocstring\relax % <for starred sections>
998   \let\hbs@bmstring\relax % <for starred sections>
999   \@ifstar
1000     {\itf@hbs@ssect#1}%
1001     {\let\hbs@seccmd=#1\hbs@checkarg}%
1002 }% \itf@hbs@do
1003 \let\hbs@do\itf@hbs@do

```

\hbs@process (patch)

```

1004 \itf@patch\pretocmd\hbs@process{%
1005   \itf@hbs@process{#1}%
1006 }%
1007 { \PackageInfoNoLine{interfaces-hypbmsec}{%
1008   {Command \string\hbs@process\space successfully patched}%
1009 } \PackageWarningNoLine{interfaces-hypbmsec}{%
1010   {Command \string\hbs@process\space could not be patched}%
1011   \MessageBreak loading of ‘interfaces-hypbmsec’ aborted%
1012   \MessageBreak \string\lastsection\space will not work}%
1013 \endinput}%

```

\itf@hbs@ssect For starred sections:

```

1014 \long\def\itf@hbs@ssect#1#2{%
1015   \itf@Not@titlesec#1
1016   {\expandafter\itf@hbs@last@starred\expandafter{\itf@hbs@currentname}{#2}}%
1017   {\let\itf@hbs@last\itf@hbs@last@starred@ttl}% <expanded at \ttl@finmarks>
1018   #1*{#2}%
1019 }% \itf@hbs@ssect

```

\itf@hbs@process For non-starred sections:

```

1020 \long\def\itf@hbs@process#1{%
1021   \itf@Not@titlesec \hbs@seccmd
1022   {%
1023     \ifdef\hyper@anchor
1024       \expandafter\itf@hbs@last@nostar@nottl@Hy\expandafter{\itf@hbs@currentname}{#1}%
1025     \else
1026       \expandafter\itf@hbs@last@nostar@nottl\expandafter{\itf@hbs@currentname}{#1}%
1027     \fi
1028   }%
1029   {\let\itf@hbs@last \itf@hbs@last@nostar@ttl}% <expanded at \ttl@finmarks>
1030 }% \itf@hbs@process

```

\itf@hbs@last@starred Update the values of \lastsectiontitle etc.. for starred sections:

```

1031 \long\def\itf@hbs@last@starred#1#2{%
1032   <expanded immediately>
1033   \GetTitleString{#2}\let\itf@hbs@currenttitle\GetTitleStringResult
1034   \begingroup
1035   \let\@elt \itf@resetlastsections
1036   \globaldefs @ne
1037   \expandafter\let\csname last#1title\endcsname\itf@hbs@currenttitle
1038   \expandafter\let\csname last#1toc\endcsname@\empty

```

```

1038      \expandafter\let\csname last#1bookmark\endcsname\@empty
1039      \expandafter\let\csname last#1anchor\endcsname\relax
1040      \let\lastsectionname \itf@hbs@currentname
1041      \csname cl@#1\endcsname
1042  \endgroup
1043 }% \itf@hbs@last@starred
1044 \def\itf@hbs@last@starred@ttl{<expanded at \ttl@finmarks>
1045   \global\let\itf@hbs@last \empty
1046   \expandafter\itf@hbs@last@starred
1047     \expandafter\itf@hbs@currentname \expandafter{\ttl@savetitle}%
1048 }% \itf@hbs@last@starred@ttl

\itf@hbs@last@nostar          Update de values of \lastsectiontitle etc.. for non-starred sections:
\itf@hbs@last@nostar@ttl
1049 \long\def\itf@hbs@last@nostar#1#2{%
1050   \GetTitleString{#2}\let\itf@hbs@currenttitle\GetTitleStringResult
\itf@hbs@last@nostar@nottl
1051 \begingroup
1052   \let\@elt \itf@resetlastsections
1053   \globaldefs \ne
1054   \expandafter\let\csname last#1title\endcsname\itf@hbs@currenttitle
1055   \expandafter\let\csname last#1toc\expandafter\endcsname
1056     \csname \ifx\hbs@tocstring\relax
1057       \itf@hbs@currenttitle\else \hbs@tocstring\fi \endcsname
1058   \expandafter\let\csname last#1bookmark\expandafter\endcsname
1059     \csname \ifx\hbs@bmstring\relax
1060       \last#1toc\else \hbs@bmstring\fi \endcsname
1061   \expandafter\let\csname last#1anchor\endcsname\HyperLocalCurrentHref
1062   \let\lastsectionname \itf@hbs@currentname
1063   \csname cl@#1\endcsname
1064 \endgroup
1065 }% \itf@hbs@last@starred
1066 \def\itf@hbs@last@nostar@ttl{<expanded at \ttl@finmarks>
1067   \global\let\itf@hbs@last \empty
1068   \expandafter\itf@hbs@last@nostar
1069     \expandafter\itf@hbs@currentname \expandafter{\ttl@savetitle}%
1070 }% \itf@hbs@last@nostar@ttl
1071 \long\def\itf@hbs@last@nostar@nottl#1#2{<expanded at \cl@section>
1072   \expandafter\let\expandafter\itf@hbs@cl@ORI\csname cl@#1\endcsname%
1073   \expandafter\def\csname cl@#1\endcsname{%
1074     \expandafter\let\csname cl@#1\endcsname\itf@hbs@cl@ORI
1075     \itf@hbs@cl@ORI
1076     \itf@hbs@last@nostar {#1}{#2}%
1077 }% \itf@hbs@last@nostar@nottl
1078 \long\def\itf@hbs@last@nostar@nottl@Hy#1#2{<expanded at \hyper@anchorend>
1079   \begingroup
1080   \toks@\expandafter{%
1081     \expandafter\def\expandafter\hyper@anchorend
1082       \expandafter{\expandafter\hyper@anchorend}%
1083     \itf@hbs@last@nostar {#1}{#2}%
1084   }\edef\x{\endgroup
1085   \def\noexpand\hyper@anchorend{\the\toks@}%
1086   }\x
1087 }% \itf@hbs@last@nostar@nottl@Hy

```

`\itf@Not@titlesec` Expands to `\@firstoftwo` if the section is not formatted by a class of `titlesec` package. Otherwise (`titlesec` formats the given section): expands to `\@secondoftwo`.

```

1088 \def\itf@Not@titlesec#1{\@firstoftwo}
1089 \AfterPackage*{titlesec}{%
1090 \def\itf@Not@titlesec#1{\begingroup
1091   \onelevel@sanitize#1%
1092   \edef\@tempa{\noexpand\in@
1093     {\noexpand\nil \string\ttl@}
1094     {\noexpand\nil #1}%
1095   }\@tempa
1096   \expandafter\endgroup

```

```

1097   \ifin@ \expandafter\@secondoftwo
1098   \else \expandafter\@firstoftwo
1099   \fi
1100 }% \itf@Not@titlesec
1101 }% AfterPackage titlesec

```

`\ttl@finmarks` When `titlesec` is used, assignments to `\lastsectiontitle`, `\lastsubsectiontoc` etc.. are done at this point.

```

1102 \AfterPackage*{titlesec}{%
1103 \expandafter\def\expandafter\ttl@finmarks\expandafter{%
1104   \expandafter\itf@hbs@last \ttl@finmarks}%
1105 \let\itf@hbs@last \empty
1106 }% AfterPackage

```

`\itf@resetlastsections` Assignments are global (`\globaldefs= 1`).

```

1107 \def\itf@resetlastsections#1{%
1108   \expandafter\let\csname last#1title\endcsname\relax
1109   \expandafter\let\csname last#1toc\endcsname\relax
1110   \expandafter\let\csname last#1bookmark\endcsname\relax
1111   \expandafter\let\csname last#1anchor\endcsname\relax
1112   \csname cl@#1\endcsname
1113 }% \itf@resetlastsections

1114 \let\lastsectionname\relax
1115 @for\itf@tp:=part, section, subsection, subsubsection, paragraph, subparagraph%
1116   \do{\expandafter\let\csname last\itf@tp title\endcsname\relax
1117     \expandafter\let\csname last\itf@tp toc\endcsname\relax
1118     \expandafter\let\csname last\itf@tp bookmark\endcsname\relax
1119     \expandafter\let\csname last\itf@tp anchor\endcsname\relax
1120   }%
1121 \AtBeginDocument{\ifundef\chapter{}%
1122   {\let\lastchaptertitle\relax
1123   \let\lastchaptertos\relax
1124   \let\lastchapterbookmark\relax
1125   \let\lastchapteranchor\relax}%
1126 }%
1127 /hypbmsec
```

21.9 `fancyhdr` : `\pagesetup`

```

1128 {*fancyhdr}
1129 \ProvidesPackage{interfaces-fancyhdr}
1130   [2010/12/27 v2.6 pgfkeys macros for fancyhdr (FC)]
1131 \RequirePackage{interfaces-base}
1132 \itf@Check@Package{fancyhdr}
1133 \robustify\fancyhf
1134 \robustify\fancyhfoffset

```

21.9.1 `\pagesetup`

`\pagesetup`

```

1135 \newrobustcmd*\pagesetup{%
1136   \let\fancy@gbl\relax % Just in case...
1137   \@ifstar
1138     {\itf@pagesetup{}}
1139     {\itf@pagesetup{reset,}}%
1140 }% \pagesetup
1141 \def\itf@pagesetup#1{@testopt
1142   {\itf@p@gesetup{#1}}{}%
1143 }% \itf@pagesetup
1144 \def\itf@p@gesetup#1[#2]#3{\ifblank{#2}%
1145   {}% modifies the current page style
1146   \itf@pgfkeys\pgfqkeys{/itf/fancyhdr}{#1#3}\relax\ps@@fancy}%

```

```

1147  {%
1148    % defines or modifies a given page style
1149    \ifblank{#1}%
1150      \csappto% star form: keep existing value (if not overwritten)
1151      \csdef% no star form: overwrite (+ reset at \pagestyle time)
1152      {ps@#2}{\itf@pgfkeys\pgfqkeys{/itf/fancyhdr}{#1#3}\relax\ps@@fancy}%
1153  }%
1154 \itf@p@gesetup
1155   @current/.is private=interfaces-fancyhdr,%
1156   @style/.is private=interfaces-fancyhdr,%
1157 }

\pagestyle

1158 \renewrobustcmd*\pagestyle[1]{\ifcsundef{ps@#1}%
1159   {\itf@undefinedpagestyle{#1}}%
1160   {\csname ps@#1\endcsname}%
1161 }% \pagestyle

\thispagestyle

1162 \renewrobustcmd*\thispagestyle[1]{\ifcsundef{ps@#1}%
1163   {\itf@undefinedpagestyle{#1}}%
1164   {\global\@specialpagetrue\gdef\@specialstyle{#1}}%
1165 }% \thispagestyle
1166 \newcommand*\itf@undefinedpagestyle[1]{%
1167   \@latex@error{Undefined page style: #1}%
1168   \MessageBreak your command will be ignored\@eha
1169 }% \itf@undefinedpagestyle

```

21.9.2 Keys for \page setup

```

1170 \pgfkeys{%
1171   /handlers/.fancy@hf/.code=%
1172     \edef\itf@tp{\pgfkeyscurrentpath}%
1173     \pgfkeysalso{\pgfkeyscurrentpath/.store as code}%
1174     \pgfkeysalso{\pgfkeyscurrentpath/.append code/.expanded=%
1175       \csdef{f@ncy#1}{%
1176         \noexpand\pgfkeysvalueof{\itf@tp/font}%
1177         \noexpand\pgfkeysvalueof{\itf@tp}\noexpand\strut}},%
1178 }% handler for fancyhdr interface
1179 \pgfqkeys{/itf/fancyhdr}{% keys for \page setup
1180   code/.code={#1},% execution at expansion of \ps@... => \pgfqkeys{...,code={...},...}
1181   page style/.style={code=\pagestyle{#1}},%
1182   page style/.alias=pagestyle,%
1183   load style/.alias=pagestyle,%
1184   head/rule/.store as dimen in cs=headrulewidth,%
1185   head/rule/.default=.4pt,%
1186   head/rule/.blank=\z@,%
1187     head/rulewidth/.alias*=head/rule,%
1188     headrulewidth/.alias*=head/rule,%
1189     head/rule/width/.alias*=head/rule,%
1190   head/rule/font/.store as code,%
1191   head/rule/color/.style={head/rule/font+=\color{#1}},%
1192   headruler/.code=%
1193     \let\headrule\pgfkeyscurrentvalue
1194     \ifx\headrule\empty\else
1195       \preto\headrule{\bgroup\pgfkeysvalueof{/itf/fancyhdr/head/rule/font}}%
1196       \appto\headrule{\egroup}%
1197     \fi,%
1198   headruler/.default=%
1199     \if@fancyplain\let\headrulewidth\plainheadrulewidth\fi
1200     \hrule@height\headrulewidth\@width\headwidth
1201     \vskip-\headrulewidth\relax,%
1202   headrule/.alias=headruler,%

```

```

1203     headrule,% sets default
1204     foot/rule/.store as dimen in cs=footrulewidth,%  

1205         foot/rule/.default=.4pt,%  

1206         foot/rule/.blank=\z@,%  

1207         foot/rulewidth/.alias*=foot/rule,%  

1208         footrulewidth/.alias*=foot/rule,%  

1209         foot/rule/width/.alias*=foot/rule,%  

1210     foot/rule/font/.store as code,%  

1211     foot/rule/color/.style={foot/rule/font+={color{#1}}},%  

1212     footruler/.code=%  

1213         \let\footrule\pgfkeyscurrentvalue  

1214         \ifx\footrule\@empty\else  

1215             \preto\footrule{\bgroup\pgfkeysvalueof{/itf/fancyhdr/head/rule/font}}%  

1216             \appto\footrule{\egroup}  

1217         \fi,%  

1218     footruler/.default=%  

1219         \if@fancyplain\let\footrulewidth\plainfootrulewidth\fi  

1220         \vskip-2\footrulewidth  

1221         \hrule@height\footrulewidth \@width\headwidth\vskip\footrulewidth,%  

1222     footrule/.alias=footruler,%  

1223     footrule,% sets default
1224     rule/.style={head/rule={#1},foot/rule={#1}},%  

1225     rule+/.style={head/rule+={#1},foot/rule+={#1}},%  

1226     rule-/.style={head/rule-={#1},foot/rule-={#1}},%  

1227     rule*/.style={head/rule*={#1},foot/rule*={#1}},%  

1228         rules/.alias*=rule,rulewidth/.alias*=rule,%  

1229         rule/width/.alias*=rule,rules/width/.alias*=rule,%  

1230     norule/.style={rule=\z@},%  

1231     norule/.value forbidden,%  

1232         norules/.alias=norule,%  

1233     rule/font/.style={head/rule/font={#1},foot/rule/font={#1}},%  

1234     rule/font+/.style={head/rule/font+={#1},foot/rule/font+={#1}},%  

1235         rule/font*/.alias=rule/font+,%  

1236         rules/font/.alias=rule/font,  

1237         rules/font+/.alias=rule/font+,%  

1238         rules/font*/.alias=rule/font+,%  

1239     rule/color/.style={head/rule/color={#1},foot/rule/color={#1}},%  

1240         rules/color/.alias=rule/color,%  

1241     foot/rule/skip/.estore as glue in cs=footruleskip,%  

1242         foot/rule/skip/.default=.3\normalbaselineskip,%  

1243         foot/rule/skip/.blank=\z@,%  

1244         foot/ruleskip/.alias*=foot/rule/skip,%  

1245         footruleskip/.alias*=foot/rule/skip,%  

1246     font/.style={head/font={#1},foot/font={#1}},%  

1247         font+/.style={head/font+={#1},foot/font+={#1}},%  

1248         font*/.alias=font+,%  

1249     color/.style={head/color={#1},foot/color={#1}},%  

1250         head/rule/color={#1},foot/rule/color={#1},%  

1251     head/.style={head/left={#1},head/center={#1},head/right={#1}},%  

1252     head+/.style={head/left+={#1},head/center+={#1},head/right+={#1}},  

1253     head*/.alias=head+,%  

1254         head/font/.style={head/left/font={#1},head/center/font={#1},head/right/font={#1}},%  

1255         head/font+/.style={head/left/font+={#1},head/center/font+={#1},head/right/font+={#1}},%  

1256         head/font*/.alias=head/font+,%  

1257         head/color/.style={head/left/color={#1},head/center/color={#1},head/right/color={#1}},%  

1258         head/rule/color={#1},%  

1259     nohead/.style={head=,head/font=,head/rule=,head/rule/font=},  

1260     foot/.style={foot/left={#1},foot/center={#1},foot/right={#1}},%  

1261     foot+/.style={foot/left+={#1},foot/center+={#1},foot/right+={#1}},  

1262     foot*/.alias=foot+,%  

1263         foot/font/.style={foot/left/font={#1},foot/center/font={#1},foot/right/font={#1}},%  

1264         foot/font+/.style={foot/left/font+={#1},foot/center/font+={#1},foot/right/font+={#1}},%  

1265         foot/font*/.alias=foot/font+,%  

1266         foot/color/.style={foot/left/color={#1},foot/center/color={#1},foot/right/color={#1}},%  

1267             foot/rule/color={#1},%

```

```

1268 nofoot/.style={foot=,foot/font=,foot/rule=,foot/rule/font=},
1269 left/.style={head/left={#1},foot/left={#1}},%
1270 left+/.style={head/left+={#1},foot/left+={#1}},%
1271 left*//.alias=left+,%
1272     left/font/.style={head/left/font={#1},foot/left/font={#1}},%
1273     left/font+/.style={head/left/font+={#1},foot/left/font+={#1}},%
1274     left/font*//.alias=left/font+,%
1275     left/color/.style={head/left/color={#1},foot/left/color={#1}},%
1276 center/.style={head/center={#1},foot/center={#1}},%
1277 center+/.style={head/center+={#1},foot/center+={#1}},%
1278 center*//.alias=center+,%
1279     center/font/.style={head/center/font={#1},foot/center/font={#1}},%
1280     center/font+/.style={head/center/font+={#1},foot/center/font+={#1}},%
1281     center/font*//.alias=center/font+,%
1282     center/color/.style={head/center/color={#1},foot/center/color={#1}},%
1283 right/.style={head/right={#1},foot/right={#1}},%
1284 right+/.style={head/right+={#1},foot/right+={#1}},%
1285 right*//.alias=right+,%
1286     right/font/.style={head/right/font={#1},foot/right/font={#1}},%
1287     right/font+/.style={head/right/font+={#1},foot/right/font+={#1}},%
1288     right/font*//.alias=right/font+,%
1289     right/color/.style={head/right/color={#1},foot/right/color={#1}},%
1290 head/left/.style={head/left/even={#1},head/left/odd={#1}},%
1291 head/left+/.style={head/left/even+={#1},head/left/odd+={#1}},%
1292 head/left*//.alias=head/left+,%
1293     head/left/font/.style={head/left/even/font={#1},head/left/odd/font={#1}},%
1294     head/left/font+/.style={head/left/even/font+={#1},head/left/odd/font+={#1}},%
1295     head/left/font*//.alias=head/left/font+,%
1296     head/left/color/.style={head/left/even/color={#1},head/left/odd/color={#1}},%
1297         left/head/.alias*=head/left,%
1298         left/head/font/.alias*=head/left/font,%
1299         left/head/color/.alias=head/left/color,%
1300 head/center/.style={head/center/even={#1},head/center/odd={#1}},%
1301 head/center+/.style={head/center/even+={#1},head/center/odd+={#1}},%
1302 head/center*//.alias=head/center+,%
1303     head/center/font/.style={head/center/even/font={#1},head/center/odd/font={#1}},%
1304     head/center/font+/.style={head/center/even/font+={#1},head/center/odd/font+={#1}},%
1305     head/center/font*//.alias=head/center/font+,%
1306     head/center/color/.style={head/center/even/color={#1},head/center/odd/color={#1}},%
1307         center/head/.alias*=head/center,%
1308         center/head/font/.alias*=head/center/font,%
1309         center/head/color/.alias=head/center/color,%
1310 head/right/.style={head/right/even={#1},head/right/odd={#1}},%
1311 head/right+/.style={head/right/even+={#1},head/right/odd+={#1}},%
1312 head/right*//.alias=head/right+,%
1313     head/right/font/.style={head/right/even/font={#1},head/right/odd/font={#1}},%
1314     head/right/font+/.style={head/right/even/font+={#1},head/right/odd/font+={#1}},%
1315     head/right/font*//.alias=head/right/font+,%
1316     head/right/color/.style={head/right/even/color={#1},head/right/odd/color={#1}},%
1317         right/head/.alias*=head/right,%
1318         right/head/font/.alias*=head/right/font,%
1319         right/head/color/.alias=head/right/color,%
1320 foot/left/.style={foot/left/even={#1},foot/left/odd={#1}},%
1321 foot/left+/.style={foot/left/even+={#1},foot/left/odd+={#1}},%
1322 foot/left*//.alias=foot/left+,%
1323     foot/left/font/.style={foot/left/even/font={#1},foot/left/odd/font={#1}},%
1324     foot/left/font+/.style={foot/left/even/font+={#1},foot/left/odd/font+={#1}},%
1325     foot/left/font*//.alias=foot/left/font+,%
1326     foot/left/color/.style={foot/left/even/color={#1},foot/left/odd/color={#1}},%
1327         left/foot/.alias*=foot/left,%
1328         left/foot/font/.alias*=foot/left/font,%
1329         left/foot/color/.alias=foot/left/color,%
1330 foot/center/.style={foot/center/even={#1},foot/center/odd={#1}},%
1331 foot/center+/.style={foot/center/even+={#1},foot/center/odd+={#1}},%
1332 foot/center*//.alias=foot/center,%

```

```

1333     foot/center/font/.style={foot/center/even/font={#1},foot/center/odd/font={#1}},%
1334     foot/center/font+/.style={foot/center/even/font+={#1},foot/center/odd/font+={#1}},%
1335     foot/center/font*/.alias=foot/center/font+,%
1336     foot/center/color/.style={foot/center/even/color={#1},foot/center/odd/color={#1}},%
1337         center/foot/.alias*=foot/center,%
1338         center/foot/font/.alias*=foot/center/font,%
1339         center/foot/color/.alias=foot/center/color,%
1340     foot/right/.style={foot/right/even={#1},foot/right/odd={#1}},%
1341     foot/right+/.style={foot/right/even+={#1},foot/right/odd+={#1}},%
1342     foot/right*/.alias=foot/right+,%
1343         foot/right/font/.style={foot/right/even/font={#1},foot/right/odd/font={#1}},%
1344         foot/right/font+/.style={foot/right/even/font+={#1},foot/right/odd/font+={#1}},%
1345         foot/right/font*/.alias=foot/right/font+,%
1346         foot/right/color/.style={foot/right/even/color={#1},foot/right/odd/color={#1}},%
1347             right/foot/.alias*=foot/right,%
1348             right/foot/font/.alias*=foot/right/font,%
1349             right/foot/color/.alias=foot/right/color,%
1350     head/left/odd/.fancy@hf=olh,%
1351     head/left/odd/font/.store as code,%
1352     head/left/odd/color/.style={head/left/odd/font+=\color{#1}},%
1353         head/odd/left/.alias*=head/left/odd,%
1354         odd/head/left/.alias*=head/left/odd,%
1355         odd/left/head/.alias*=head/left/odd,%
1356         left/odd/head/.alias*=head/left/odd,%
1357         left/head/odd/.alias*=head/left/odd,%
1358         head/odd/left/font/.alias*=head/left/odd/font,%
1359         odd/head/left/font/.alias*=head/left/odd/font,%
1360         odd/left/head/font/.alias*=head/left/odd/font,%
1361         left/odd/head/font/.alias*=head/left/odd/font,%
1362         left/head/odd/font/.alias*=head/left/odd/font,%
1363         head/odd/left/color/.alias=head/left/odd/color,%
1364         odd/head/left/color/.alias=head/left/odd/color,%
1365         odd/left/head/color/.alias=head/left/odd/color,%
1366         left/odd/head/color/.alias=head/left/odd/color,%
1367         left/head/odd/color/.alias=head/left/odd/color,%
1368     head/center/odd/.fancy@hf=och,%
1369     head/center/odd/font/.store as code,%
1370     head/center/odd/color/.style={head/center/odd/font+=\color{#1}},%
1371         head/odd/center/.alias*=head/center/odd,%
1372         odd/head/center/.alias*=head/center/odd,%
1373         odd/center/head/.alias*=head/center/odd,%
1374         center/odd/head/.alias*=head/center/odd,%
1375         center/head/odd/.alias*=head/center/odd,%
1376         head/odd/center/font/.alias*=head/center/odd/font,%
1377         odd/head/center/font/.alias*=head/center/odd/font,%
1378         odd/center/head/font/.alias*=head/center/odd/font,%
1379         center/odd/head/font/.alias*=head/center/odd/font,%
1380         center/head/odd/font/.alias*=head/center/odd/font,%
1381         head/odd/center/color/.alias=head/center/odd/color,%
1382         odd/head/center/color/.alias=head/center/odd/color,%
1383         odd/center/head/color/.alias=head/center/odd/color,%
1384         center/odd/head/color/.alias=head/center/odd/color,%
1385         center/head/odd/color/.alias=head/center/odd/color,%
1386     head/right/odd/.fancy@hf=orh,%
1387     head/right/odd/font/.store as code,%
1388     head/right/odd/color/.style={head/right/odd/font+=\color{#1}},%
1389         head/odd/right/.alias*=head/right/odd,%
1390         odd/head/right/.alias*=head/right/odd,%
1391         odd/right/head/.alias*=head/right/odd,%
1392         right/odd/head/.alias*=head/right/odd,%
1393         right/head/odd/.alias*=head/right/odd,%
1394         head/odd/right/font/.alias*=head/right/odd/font,%
1395         odd/head/right/font/.alias*=head/right/odd/font,%
1396         odd/right/head/font/.alias*=head/right/odd/font,%
1397         right/odd/head/font/.alias*=head/right/odd/font,%

```

```

1398     right/head/odd/font/.alias*=head/right/odd/font,%
1399     head/odd/right/color/.alias=head/right/odd/color,%
1400     odd/head/right/color/.alias=head/right/odd/color,%
1401     odd/right/head/color/.alias=head/right/odd/color,%
1402     right/odd/head/color/.alias=head/right/odd/color,%
1403     right/head/odd/color/.alias=head/right/odd/color,%
1404 head/left/even/.fancy@hf=elh,%
1405 head/left/even/font/.store as code,%
1406 head/left/even/color/.style={head/left/even/font+=\color{#1}},%
1407     head/even/left/.alias*=head/left/even,%
1408     even/head/left/.alias*=head/left/even,%
1409     even/left/head/.alias*=head/left/even,%
1410     left/even/head/.alias*=head/left/even,%
1411     left/head/even/.alias*=head/left/even,%
1412     head/even/left/font/.alias*=head/left/even/font,%
1413     even/head/left/font/.alias*=head/left/even/font,%
1414     even/left/head/font/.alias*=head/left/even/font,%
1415     left/even/head/font/.alias*=head/left/even/font,%
1416     left/head/even/font/.alias*=head/left/even/font,%
1417     head/even/left/color/.alias=head/left/even/color,%
1418     even/head/left/color/.alias=head/left/even/color,%
1419     even/left/head/color/.alias=head/left/even/color,%
1420     left/even/head/color/.alias=head/left/even/color,%
1421     left/head/even/color/.alias=head/left/even/color,%
1422 head/center/even/.fancy@hf=ech,%
1423 head/center/even/font/.store as code,%
1424 head/center/even/color/.style={head/center/even/font+=\color{#1}},%
1425     head/even/center/.alias*=head/center/even,%
1426     even/head/center/.alias*=head/center/even,%
1427     even/center/head/.alias*=head/center/even,%
1428     center/even/head/.alias*=head/center/even,%
1429     center/head/even/.alias*=head/center/even,%
1430     head/even/center/font/.alias*=head/center/even/font,%
1431     even/head/center/font/.alias*=head/center/even/font,%
1432     even/center/head/font/.alias*=head/center/even/font,%
1433     center/even/head/font/.alias*=head/center/even/font,%
1434     center/head/even/font/.alias*=head/center/even/font,%
1435     head/even/center/color/.alias=head/center/even/color,%
1436     even/head/center/color/.alias=head/center/even/color,%
1437     even/center/head/color/.alias=head/center/even/color,%
1438     center/even/head/color/.alias=head/center/even/color,%
1439     center/head/even/color/.alias=head/center/even/color,%
1440 head/right/even/.fancy@hf=erh,%
1441 head/right/even/font/.store as code,%
1442 head/right/even/color/.style={head/right/even/font+=\color{#1}},%
1443     head/even/right/.alias*=head/right/even,%
1444     even/head/right/.alias*=head/right/even,%
1445     even/right/head/.alias*=head/right/even,%
1446     right/even/head/.alias*=head/right/even,%
1447     right/head/even/.alias*=head/right/even,%
1448     head/even/right/font/.alias*=head/right/even/font,%
1449     even/head/right/font/.alias*=head/right/even/font,%
1450     even/right/head/font/.alias*=head/right/even/font,%
1451     right/even/head/font/.alias*=head/right/even/font,%
1452     right/head/even/font/.alias*=head/right/even/font,%
1453     head/even/right/color/.alias=head/right/even/color,%
1454     even/head/right/color/.alias=head/right/even/color,%
1455     even/right/head/color/.alias=head/right/even/color,%
1456     right/even/head/color/.alias=head/right/even/color,%
1457     right/head/even/color/.alias=head/right/even/color,%
1458 foot/left/odd/.fancy@hf=olff,%
1459 foot/left/odd/font/.store as code,%
1460 foot/left/odd/color/.style={foot/left/odd/font+=\color{#1}},%
1461     foot/odd/left/.alias*=foot/left/odd,%
1462     odd/foot/left/.alias*=foot/left/odd,%

```

```
1463     odd/left/foot/.alias*=foot/left/odd,%
1464     left/odd/foot/.alias*=foot/left/odd,%
1465     left/foot/odd/.alias*=foot/left/odd,%
1466     foot/odd/left/font/.alias*=foot/left/odd/font,%
1467     odd/foot/left/font/.alias*=foot/left/odd/font,%
1468     odd/left/foot/font/.alias*=foot/left/odd/font,%
1469     left/odd/foot/font/.alias*=foot/left/odd/font,%
1470     left/foot/odd/font/.alias*=foot/left/odd/font,%
1471     foot/odd/left/color/.alias=foot/left/odd/color,%
1472     odd/foot/left/color/.alias=foot/left/odd/color,%
1473     odd/left/foot/color/.alias=foot/left/odd/color,%
1474     left/odd/foot/color/.alias=foot/left/odd/color,%
1475     left/foot/odd/color/.alias=foot/left/odd/color,%
1476     foot/center/odd/.fancy@hf=ocf,%
1477     foot/center/odd/font/.store as code,%
1478     foot/center/odd/color/.style={foot/center/odd/font+=\color{#1}},%
1479         foot/odd/center/.alias*=foot/center/odd,%
1480         odd/foot/center/.alias*=foot/center/odd,%
1481         odd/center/foot/.alias*=foot/center/odd,%
1482         center/odd/foot/.alias*=foot/center/odd,%
1483         center/foot/odd/.alias*=foot/center/odd,%
1484         foot/odd/center/font/.alias*=foot/center/odd/font,%
1485         odd/foot/center/font/.alias*=foot/center/odd/font,%
1486         odd/center/foot/font/.alias*=foot/center/odd/font,%
1487         center/odd/foot/font/.alias*=foot/center/odd/font,%
1488         center/foot/odd/font/.alias*=foot/center/odd/font,%
1489         foot/odd/center/color/.alias=foot/center/odd/color,%
1490         odd/foot/center/color/.alias=foot/center/odd/color,%
1491         odd/center/foot/color/.alias=foot/center/odd/color,%
1492         center/odd/foot/color/.alias=foot/center/odd/color,%
1493         center/foot/odd/color/.alias=foot/center/odd/color,%
1494     foot/right/odd/.fancy@hf=orf,%
1495     foot/right/odd/font/.store as code,%
1496     foot/right/odd/color/.style={foot/right/odd/font+=\color{#1}},%
1497         foot/odd/right/.alias*=foot/right/odd,%
1498         odd/foot/right/.alias*=foot/right/odd,%
1499         odd/right/foot/.alias*=foot/right/odd,%
1500         right/odd/foot/.alias*=foot/right/odd,%
1501         right/foot/odd/.alias*=foot/right/odd,%
1502         foot/odd/right/font/.alias*=foot/right/odd/font,%
1503         odd/foot/right/font/.alias*=foot/right/odd/font,%
1504         odd/right/foot/font/.alias*=foot/right/odd/font,%
1505         right/odd/foot/font/.alias*=foot/right/odd/font,%
1506         right/foot/odd/font/.alias*=foot/right/odd/font,%
1507         foot/odd/right/color/.alias=foot/right/odd/color,%
1508         odd/foot/right/color/.alias=foot/right/odd/color,%
1509         odd/right/foot/color/.alias=foot/right/odd/color,%
1510         right/odd/foot/color/.alias=foot/right/odd/color,%
1511         right/foot/odd/color/.alias=foot/right/odd/color,%
1512     foot/left/even/.fancy@hf=elf,%
1513     foot/left/even/font/.store as code,%
1514     foot/left/even/color/.style={foot/left/even/font+=\color{#1}},%
1515         foot/even/left/.alias*=foot/left/even,%
1516         even/foot/left/.alias*=foot/left/even,%
1517         even/left/foot/.alias*=foot/left/even,%
1518         left/even/foot/.alias*=foot/left/even,%
1519         left/foot/even/.alias*=foot/left/even,%
1520         foot/even/left/font/.alias*=foot/left/even/font,%
1521         even/foot/left/font/.alias*=foot/left/even/font,%
1522         even/left/foot/font/.alias*=foot/left/even/font,%
1523         left/even/foot/font/.alias*=foot/left/even/font,%
1524         left/foot/even/font/.alias*=foot/left/even/font,%
1525         foot/even/left/color/.alias=foot/left/even/color,%
1526         even/foot/left/color/.alias=foot/left/even/color,%
1527         even/left/foot/color/.alias=foot/left/even/color,%
```

```

1528     left/even/foot/color/.alias=foot/left/even/color,%  

1529         left/foot/even/color/.alias=foot/left/even/color,%  

1530         foot/center/even/.fancy@hf=ecf,%  

1531         foot/center/even/font/.store as code,%  

1532         foot/center/even/color/.style={foot/center/even/font+=\color{\#1}},%  

1533             foot/even/center/.alias*=foot/center/even,%  

1534                 even/foot/center/.alias*=foot/center/even,%  

1535                 even/center/foot/.alias*=foot/center/even,%  

1536                 center/even/foot/.alias*=foot/center/even,%  

1537                 center/foot/even/.alias*=foot/center/even,%  

1538                 foot/even/center/font/.alias*=foot/center/even/font,%  

1539                     even/foot/center/font/.alias*=foot/center/even/font,%  

1540                     even/center/foot/font/.alias*=foot/center/even/font,%  

1541                     center/even/foot/font/.alias*=foot/center/even/font,%  

1542                     center/foot/even/font/.alias*=foot/center/even/font,%  

1543                     foot/even/center/color/.alias=foot/center/even/color,%  

1544                         even/foot/center/color/.alias=foot/center/even/color,%  

1545                         even/center/foot/color/.alias=foot/center/even/color,%  

1546                         center/even/foot/color/.alias=foot/center/even/color,%  

1547                         center/foot/even/color/.alias=foot/center/even/color,%  

1548         foot/right/even/.fancy@hf=erf,%  

1549         foot/right/even/font/.store as code,%  

1550         foot/right/even/color/.style={foot/right/even/font+=\color{\#1}},%  

1551             foot/even/right/.alias*=foot/right/even,%  

1552                 even/foot/right/.alias*=foot/right/even,%  

1553                 even/right/foot/.alias*=foot/right/even,%  

1554                 right/even/foot/.alias*=foot/right/even,%  

1555                 right/foot/even/.alias*=foot/right/even,%  

1556                 foot/even/right/font/.alias*=foot/right/even/font,%  

1557                     even/foot/right/font/.alias*=foot/right/even/font,%  

1558                     even/right/foot/font/.alias*=foot/right/even/font,%  

1559                     right/even/foot/font/.alias*=foot/right/even/font,%  

1560                     right/foot/even/font/.alias*=foot/right/even/font,%  

1561                     foot/even/right/color/.alias=foot/right/even/color,%  

1562                         even/foot/right/color/.alias=foot/right/even/color,%  

1563                         even/right/foot/color/.alias=foot/right/even/color,%  

1564                         right/even/foot/color/.alias=foot/right/even/color,%  

1565                         right/foot/even/color/.alias=foot/right/even/color,%  

1566         foot/inner/.style={foot/right/even=\#1,foot/left/odd=\#1},%  

1567         foot/inner+/.style={foot/right/even+=\#1,foot/left/odd+=\#1},%  

1568         foot/inner*/.alias=foot/inner+,%  

1569         foot/inner/font/.style={foot/right/even/font=\#1,foot/left/odd/font=\#1},%  

1570         foot/inner/color/.style={foot/right/even/color=\#1,foot/left/odd/color=\#1},%  

1571         foot/inner/font+/.style={foot/right/even/font+=\#1,foot/left/odd/font+=\#1},%  

1572         foot/inner/font*/.alias=foot/inner/font+,%  

1573             inner/foot/.alias*=foot/inner,%  

1574                 inner/foot/font/.alias*=foot/inner/font,%  

1575                 inner/foot/color/.alias=foot/inner/color,%  

1576         foot/outer/.style={foot/left/even=\#1,foot/right/odd=\#1},%  

1577         foot/outer+/.style={foot/left/even+=\#1,foot/right/odd+=\#1},%  

1578         foot/outer*/.alias=foot/outer+,%  

1579         foot/outer/font/.style={foot/left/even/font=\#1,foot/right/odd/font=\#1},%  

1580         foot/outer/color/.style={foot/left/even/color=\#1,foot/right/odd/color=\#1},%  

1581         foot/outer/font+/.style={foot/left/even/font+=\#1,foot/right/odd/font+=\#1},%  

1582         foot/outer/font*/.alias=foot/outer/font+,%  

1583             outer/foot/.alias*=foot/outer,%  

1584                 outer/foot/font/.alias*=foot/outer/font,%  

1585                 outer/foot/color/.alias=foot/outer/color,%  

1586         head/inner/.style={head/right/even=\#1,head/left/odd=\#1},%  

1587         head/inner+/.style={head/right/even+=\#1,head/left/odd+=\#1},%  

1588         head/inner*/.alias=head/inner+,%  

1589         head/inner/font/.style={head/right/even/font=\#1,head/left/odd/font=\#1},%  

1590         head/inner/color/.style={head/right/even/color=\#1,head/left/odd/color=\#1},%  

1591         head/inner/font+/.style={head/right/even/font+=\#1,head/left/odd/font+=\#1},%  

1592         head/inner/font*/.alias=head/inner/font+,%

```

```

1593     inner/head/.alias*=head/inner,%  

1594     inner/head/font/.alias*=head/inner/font,%  

1595     inner/head/color/.alias=head/inner/color,%  

1596 head/outer/.style={head/left/even={#1},head/right/odd={#1}},%  

1597 head/outer+/.style={head/left/even+={#1},head/right/odd+={#1}},%  

1598 head/outer*/.alias=head/outer+,%  

1599 head/outer/font/.style={head/left/even/font={#1},head/right/odd/font={#1}},%  

1600 head/outer/color/.style={head/left/even/color={#1},head/right/odd/color={#1}},%  

1601 head/outer/font+/.style={head/left/even/font+={#1},head/right/odd/font+={#1}},%  

1602 head/outer/font*/.alias=head/outer/font+,%  

1603     outer/head/.alias*=head/outer,%  

1604     outer/head/font/.alias*=head/outer/font,%  

1605     outer/head/color/.alias=head/outer/color,%  

1606 inner/.style={foot/inner={#1},head/inner={#1}},%  

1607 inner+/.style={foot/inner+={#1},head/inner+={#1}},%  

1608 inner*/.alias=inner+,%  

1609 inner/font/.style={foot/inner/font={#1},head/inner/font={#1}},%  

1610 inner/font+/.style={foot/inner/font+={#1},head/inner/font+={#1}},%  

1611 inner/font*/.alias=inner/font+,%  

1612 inner/color/.style={foot/inner/color={#1},head/inner/color={#1}},%  

1613 outer/.style={foot/outer={#1},head/outer={#1}},%  

1614 outer+/.style={foot/outer+={#1},head/outer+={#1}},%  

1615 outer*/.alias=outer+,%  

1616 outer/font/.style={foot/outer/font={#1},head/outer/font={#1}},%  

1617 outer/font+/.style={foot/outer/font+={#1},head/outer/font+={#1}},%  

1618 outer/font*/.alias=outer/font+,%  

1619 outer/color/.style={foot/outer/color={#1},head/outer/color={#1}},%  

1620 offset/.style={head/offset={#1},foot/offset={#1}},%  

1621     offset+/.style={head/offset+={#1},foot/offset+={#1}},%  

1622     offset-/.style={head/offset-={#1},foot/offset-={#1}},%  

1623     offset*/.style={head/offset*={#1},foot/offset*={#1}},%  

1624 head/offset/.style={head/left/offset={#1},head/right/offset={#1}},%  

1625     head/offset+/.style={head/left/offset+={#1},head/right/offset+={#1}},%  

1626     head/offset-/.style={head/left/offset-={#1},head/right/offset-={#1}},%  

1627     head/offset*/.style={head/left/offset*={#1},head/right/offset*={#1}},%  

1628         offset/head/.alias*=head/offset,%  

1629 foot/offset/.style={foot/left/offset={#1},foot/right/offset={#1}},%  

1630     foot/offset+/.style={foot/left/offset+={#1},foot/right/offset+={#1}},%  

1631     foot/offset-/.style={foot/left/offset-={#1},foot/right/offset-={#1}},%  

1632     foot/offset*/.style={foot/left/offset*={#1},foot/right/offset*={#1}},%  

1633         offset/foot/.alias*=foot/offset,%  

1634 odd/offset/.style={left/odd/offset={#1},right/odd/offset={#1}},%  

1635     odd/offset+/.style={left/odd/offset+={#1},right/odd/offset+={#1}},%  

1636     odd/offset-/.style={left/odd/offset-={#1},right/odd/offset-={#1}},%  

1637         offset/odd/.alias*=odd/offset,%  

1638 even/offset/.style={left/even/offset={#1},right/even/offset={#1}},%  

1639     even/offset+/.style={left/even/offset+={#1},right/even/offset+={#1}},%  

1640     even/offset-/.style={left/even/offset-={#1},right/even/offset-={#1}},%  

1641         offset/even/.alias*=even/offset,%  

1642 left/offset/.style={foot/left/offset={#1},head/left/offset={#1}},%  

1643     left/offset+/.style={foot/left/offset+={#1},head/left/offset+={#1}},%  

1644     left/offset-/.style={foot/left/offset-={#1},head/left/offset-={#1}},%  

1645     left/offset*/.style={foot/left/offset*={#1},head/left/offset*={#1}},%  

1646         offset/left/.alias*=left/offset,%  

1647 left/even/offset/.style={foot/left/even/offset={#1},head/left/even/offset={#1}},%  

1648     left/even/offset+/.style={foot/left/even/offset+={#1},head/left/even/offset+={#1}},%  

1649     left/even/offset-/.style={foot/left/even/offset-={#1},head/left/even/offset-={#1}},%  

1650     left/even/offset*/.style={foot/left/even/offset*={#1},head/left/even/offset*={#1}},%  

1651         left/offset/even/.alias*=left/even/offset,%  

1652         even/left/offset/.alias*=left/even/offset,%  

1653         even/offset/left/.alias*=left/even/offset,%  

1654         offset/even/left/.alias*=left/even/offset,%  

1655         offset/left/even/.alias*=left/even/offset,%  

1656 left/odd/offset/.style={foot/left/odd/offset={#1},head/left/odd/offset={#1}},%  

1657     left/odd/offset+/.style={foot/left/odd/offset+={#1},head/left/odd/offset+={#1}},%

```

```

1658     left/odd/offset/.style={foot/left/odd/offset-={#1},head/left/odd/offset-={#1}},%
1659     left/odd/offset*/.style={foot/left/odd/offset*={#1},head/left/odd/offset*={#1}},%
1660         left/offset/odd/.alias*=left/odd/offset,%
1661         odd/left/offset/.alias*=left/odd/offset,%
1662         odd/offset/left/.alias*=left/odd/offset,%
1663         offset/odd/left/.alias*=left/odd/offset,%
1664         offset/left/odd/.alias*=left/odd/offset,%
1665     right/offset/.style={foot/right/offset={#1},head/right/offset={#1}},%
1666         right/offset+/.style={foot/right/offset+={#1},head/right/offset+={#1}},%
1667         right/offset-/.style={foot/right/offset-={#1},head/right/offset-={#1}},%
1668         right/offset*/.style={foot/right/offset*={#1},head/right/offset*={#1}},%
1669             offset/right/.alias*=right/offset,%
1670     right/even/offset/.style={foot/right/even/offset={#1},head/right/even/offset={#1}},%
1671         right/even/offset+/.style={foot/right/even/offset+={#1},head/right/even/offset+={#1}},%
1672         right/even/offset-/.style={foot/right/even/offset-={#1},head/right/even/offset-={#1}},%
1673         right/even/offset*/.style={foot/right/even/offset*={#1},head/right/even/offset*={#1}},%
1674             right/offset/even/.alias*=right/even/offset,%
1675             even/right/offset/.alias*=right/even/offset,%
1676             even/offset/right/.alias*=right/even/offset,%
1677             offset/even/right/.alias*=right/even/offset,%
1678             offset/right/even/.alias*=right/even/offset,%
1679     right/odd/offset/.style={foot/right/odd/offset={#1},head/right/odd/offset={#1}},%
1680         right/odd/offset+/.style={foot/right/odd/offset+={#1},head/right/odd/offset+={#1}},%
1681         right/odd/offset-/.style={foot/right/odd/offset-={#1},head/right/odd/offset-={#1}},%
1682         right/odd/offset*/.style={foot/right/odd/offset*={#1},head/right/odd/offset*={#1}},%
1683             right/offset/odd/.alias*=right/odd/offset,%
1684             odd/right/offset/.alias*=right/odd/offset,%
1685             odd/offset/right/.alias*=right/odd/offset,%
1686             offset/odd/right/.alias*=right/odd/offset,%
1687             offset/right/odd/.alias*=right/odd/offset,%
1688     head/left/offset/.style={head/left/even/offset={#1},head/left/odd/offset={#1}},%
1689         head/left/offset+/.style={head/left/even/offset+={#1},head/left/odd/offset+={#1}},%
1690         head/left/offset-/.style={head/left/even/offset-={#1},head/left/odd/offset-={#1}},%
1691         head/left/offset*/.style={head/left/even/offset*={#1},head/left/odd/offset*={#1}},%
1692             head/offset/left/.alias*=head/left/offset,%
1693             left/head/offset/.alias*=head/left/offset,%
1694             left/offset/head/.alias*=head/left/offset,%
1695             offset/head/left/.alias*=head/left/offset,%
1696             offset/left/head/.alias*=head/left/offset,%
1697     head/right/offset/.style={head/right/even/offset={#1},head/right/odd/offset={#1}},%
1698         head/right/offset+/.style={head/right/even/offset+={#1},head/right/odd/offset+={#1}},%
1699         head/right/offset-/.style={head/right/even/offset-={#1},head/right/odd/offset-={#1}},%
1700         head/right/offset*/.style={head/right/even/offset*={#1},head/right/odd/offset*={#1}},%
1701             head/offset/right/.alias*=head/right/offset,%
1702             right/head/offset/.alias*=head/right/offset,%
1703             right/offset/head/.alias*=head/right/offset,%
1704             offset/head/right/.alias*=head/right/offset,%
1705             offset/right/head/.alias*=head/right/offset,%
1706     foot/left/offset/.style={foot/left/even/offset={#1},foot/left/odd/offset={#1}},%
1707         foot/left/offset+/.style={foot/left/even/offset+={#1},foot/left/odd/offset+={#1}},%
1708         foot/left/offset-/.style={foot/left/even/offset-={#1},foot/left/odd/offset-={#1}},%
1709         foot/left/offset*/.style={foot/left/even/offset*={#1},foot/left/odd/offset*={#1}},%
1710             foot/offset/left/.alias*=foot/left/offset,%
1711             left/foot/offset/.alias*=foot/left/offset,%
1712             left/offset/foot/.alias*=foot/left/offset,%
1713             offset/foot/left/.alias*=foot/left/offset,%
1714             offset/left/foot/.alias*=foot/left/offset,%
1715     foot/right/offset/.style={foot/right/even/offset={#1},foot/right/odd/offset={#1}},%
1716         foot/right/offset+/.style={foot/right/even/offset+={#1},foot/right/odd/offset+={#1}},%
1717         foot/right/offset-/.style={foot/right/even/offset-={#1},foot/right/odd/offset-={#1}},%
1718         foot/right/offset*/.style={foot/right/even/offset*={#1},foot/right/odd/offset*={#1}},%
1719             foot/offset/right/.alias*=foot/right/offset,%
1720             right/foot/offset/.alias*=foot/right/offset,%
1721             right/offset/foot/.alias*=foot/right/offset,%
1722             offset/foot/right/.alias*=foot/right/offset,%

```

```

1723     offset/right/foot/.alias*=foot/right/offset,%
1724 head/left/even/offset/.store in skip cs=f@ncy0@elh,%
1725     head/left/even/offset/.append code=\fancy@setoffs,%
1726         head/even/offset/left/.alias*=head/left/even/offset,%
1727         head/even/left/offset/.alias*=head/left/even/offset,%
1728         head/offset/even/left/.alias*=head/left/even/offset,%
1729         head/offset/left/even/.alias*=head/left/even/offset,%
1730         head/left/offset/even/.alias*=head/left/even/offset,%
1731         left/head/offset/even/.alias*=head/left/even/offset,%
1732         left/head/even/offset/.alias*=head/left/even/offset,%
1733         left/offset/head/even/.alias*=head/left/even/offset,%
1734         left/offset/even/head/.alias*=head/left/even/offset,%
1735         left/even/offset/head/.alias*=head/left/even/offset,%
1736         left/even/head/offset/.alias*=head/left/even/offset,%
1737         even/head/offset/left/.alias*=head/left/even/offset,%
1738         even/head/left/offset/.alias*=head/left/even/offset,%
1739         even/offset/head/left/.alias*=head/left/even/offset,%
1740         even/offset/left/head/.alias*=head/left/even/offset,%
1741         even/left/offset/head/.alias*=head/left/even/offset,%
1742         even/left/head/offset/.alias*=head/left/even/offset,%
1743         offset/head/left/even/.alias*=head/left/even/offset,%
1744         offset/head/even/left/.alias*=head/left/even/offset,%
1745         offset/left/head/even/.alias*=head/left/even/offset,%
1746         offset/left/even/head/.alias*=head/left/even/offset,%
1747         offset/even/left/head/.alias*=head/left/even/offset,%
1748         offset/even/head/left/.alias*=head/left/even/offset,%
1749 head/left/odd/offset/.store in skip cs=f@ncy0@olh,%
1750     head/left/odd/offset/.append code=\fancy@setoffs,%
1751         head/odd/offset/left/.alias*=head/left/odd/offset,%
1752         head/odd/left/offset/.alias*=head/left/odd/offset,%
1753         head/offset/odd/left/.alias*=head/left/odd/offset,%
1754         head/offset/left/odd/.alias*=head/left/odd/offset,%
1755         head/left/offset/odd/.alias*=head/left/odd/offset,%
1756         left/head/offset/odd/.alias*=head/left/odd/offset,%
1757         left/head/odd/offset/.alias*=head/left/odd/offset,%
1758         left/offset/head/odd/.alias*=head/left/odd/offset,%
1759         left/offset/odd/head/.alias*=head/left/odd/offset,%
1760         left/odd/offset/head/.alias*=head/left/odd/offset,%
1761         left/odd/head/offset/.alias*=head/left/odd/offset,%
1762         odd/head/offset/left/.alias*=head/left/odd/offset,%
1763         odd/head/left/offset/.alias*=head/left/odd/offset,%
1764         odd/offset/head/left/.alias*=head/left/odd/offset,%
1765         odd/offset/left/head/.alias*=head/left/odd/offset,%
1766         odd/left/offset/head/.alias*=head/left/odd/offset,%
1767         odd/left/head/offset/.alias*=head/left/odd/offset,%
1768         offset/head/left/odd/.alias*=head/left/odd/offset,%
1769         offset/head/odd/left/.alias*=head/left/odd/offset,%
1770         offset/left/head/odd/.alias*=head/left/odd/offset,%
1771         offset/left/odd/head/.alias*=head/left/odd/offset,%
1772         offset/odd/left/head/.alias*=head/left/odd/offset,%
1773         offset/odd/head/left/.alias*=head/left/odd/offset,%
1774 head/right/even/offset/.store in skip cs=f@ncy0@erh,%
1775     head/right/even/offset/.append code=\fancy@setoffs,%
1776         head/even/offset/right/.alias*=head/right/even/offset,%
1777         head/even/right/offset/.alias*=head/right/even/offset,%
1778         head/offset/even/right/.alias*=head/right/even/offset,%
1779         head/offset/right/even/.alias*=head/right/even/offset,%
1780         head/right/offset/even/.alias*=head/right/even/offset,%
1781         right/head/offset/even/.alias*=head/right/even/offset,%
1782         right/head/even/offset/.alias*=head/right/even/offset,%
1783         right/offset/head/even/.alias*=head/right/even/offset,%
1784         right/offset/even/head/.alias*=head/right/even/offset,%
1785         right/even/offset/head/.alias*=head/right/even/offset,%
1786         right/even/head/offset/.alias*=head/right/even/offset,%
1787         even/head/offset/right/.alias*=head/right/even/offset,%

```

```

1788 even/head/right/offset/.alias*=head/right/even/offset,%
1789 even/offset/head/right/.alias*=head/right/even/offset,%
1790 even/offset/right/head/.alias*=head/right/even/offset,%
1791 even/right/offset/head/.alias*=head/right/even/offset,%
1792 even/right/head/offset/.alias*=head/right/even/offset,%
1793 offset/head/right/even/.alias*=head/right/even/offset,%
1794 offset/head/even/right/.alias*=head/right/even/offset,%
1795 offset/right/head/even/.alias*=head/right/even/offset,%
1796 offset/right/even/head/.alias*=head/right/even/offset,%
1797 offset/even/right/head/.alias*=head/right/even/offset,%
1798 offset/even/head/right/.alias*=head/right/even/offset,%
1799 head/right/odd/offset/.store in skip cs=f@ncy0@orh,%
1800     head/right/odd/offset/.append code=\fancy@setoffs,%
1801     head/odd/offset/right/.alias*=head/right/odd/offset,%
1802     head/odd/right/offset/.alias*=head/right/odd/offset,%
1803     head/offset/odd/right/.alias*=head/right/odd/offset,%
1804     head/offset/right/odd/.alias*=head/right/odd/offset,%
1805     head/right/offset/odd/.alias*=head/right/odd/offset,%
1806     right/head/offset/odd/.alias*=head/right/odd/offset,%
1807     right/head/odd/offset/.alias*=head/right/odd/offset,%
1808     right/offset/head/odd/.alias*=head/right/odd/offset,%
1809     right/offset/odd/head/.alias*=head/right/odd/offset,%
1810     right/odd/offset/head/.alias*=head/right/odd/offset,%
1811     right/odd/head/offset/.alias*=head/right/odd/offset,%
1812     odd/head/offset/right/.alias*=head/right/odd/offset,%
1813     odd/head/right/offset/.alias*=head/right/odd/offset,%
1814     odd/offset/head/right/.alias*=head/right/odd/offset,%
1815     odd/offset/right/head/.alias*=head/right/odd/offset,%
1816     odd/right/offset/head/.alias*=head/right/odd/offset,%
1817     odd/right/head/offset/.alias*=head/right/odd/offset,%
1818     offset/head/right/odd/.alias*=head/right/odd/offset,%
1819     offset/head/odd/right/.alias*=head/right/odd/offset,%
1820     offset/right/head/odd/.alias*=head/right/odd/offset,%
1821     offset/right/odd/head/.alias*=head/right/odd/offset,%
1822     offset/odd/right/head/.alias*=head/right/odd/offset,%
1823     offset/odd/head/right/.alias*=head/right/odd/offset,%
1824 foot/left/even/offset/.store in skip cs=f@ncy0@elf,%
1825     foot/left/even/offset/.append code=\fancy@setoffs,%
1826     foot/even/offset/left/.alias*=foot/left/even/offset,%
1827     foot/even/left/offset/.alias*=foot/left/even/offset,%
1828     foot/offset/even/left/.alias*=foot/left/even/offset,%
1829     foot/offset/left/even/.alias*=foot/left/even/offset,%
1830     foot/left/offset/even/.alias*=foot/left/even/offset,%
1831     left/foot/offset/even/.alias*=foot/left/even/offset,%
1832     left/foot/even/offset/.alias*=foot/left/even/offset,%
1833     left/offset/foot/even/.alias*=foot/left/even/offset,%
1834     left/offset/even/foot/.alias*=foot/left/even/offset,%
1835     left/even/offset/foot/.alias*=foot/left/even/offset,%
1836     left/even/foot/offset/.alias*=foot/left/even/offset,%
1837     even/foot/offset/left/.alias*=foot/left/even/offset,%
1838     even/foot/left/offset/.alias*=foot/left/even/offset,%
1839     even/offset/foot/left/.alias*=foot/left/even/offset,%
1840     even/offset/left/foot/.alias*=foot/left/even/offset,%
1841     even/left/offset/foot/.alias*=foot/left/even/offset,%
1842     even/left/foot/offset/.alias*=foot/left/even/offset,%
1843     offset/foot/left/even/.alias*=foot/left/even/offset,%
1844     offset/foot/even/left/.alias*=foot/left/even/offset,%
1845     offset/left/foot/even/.alias*=foot/left/even/offset,%
1846     offset/left/even/foot/.alias*=foot/left/even/offset,%
1847     offset/even/left/foot/.alias*=foot/left/even/offset,%
1848     offset/even/foot/left/.alias*=foot/left/even/offset,%
1849 foot/left/odd/offset/.store in skip cs=f@ncy0@olf,%
1850     foot/left/odd/offset/.append code=\fancy@setoffs,%
1851     foot/odd/offset/left/.alias*=foot/left/odd/offset,%
1852     foot/odd/left/offset/.alias*=foot/left/odd/offset,%

```

```
1853     foot/offset/odd/left/.alias*=foot/left/odd/offset,%
1854     foot/offset/left/odd/.alias*=foot/left/odd/offset,%
1855     foot/left/offset/odd/.alias*=foot/left/odd/offset,%
1856     left/foot/offset/odd/.alias*=foot/left/odd/offset,%
1857     left/foot/odd/offset/.alias*=foot/left/odd/offset,%
1858     left/offset/foot/odd/.alias*=foot/left/odd/offset,%
1859     left/offset/odd/foot/.alias*=foot/left/odd/offset,%
1860     left/odd/offset/foot/.alias*=foot/left/odd/offset,%
1861     left/odd/foot/offset/.alias*=foot/left/odd/offset,%
1862     odd/foot/offset/left/.alias*=foot/left/odd/offset,%
1863     odd/foot/left/offset/.alias*=foot/left/odd/offset,%
1864     odd/offset/foot/left/.alias*=foot/left/odd/offset,%
1865     odd/offset/left/foot/.alias*=foot/left/odd/offset,%
1866     odd/left/offset/foot/.alias*=foot/left/odd/offset,%
1867     odd/left/foot/offset/.alias*=foot/left/odd/offset,%
1868     offset/foot/left/odd/.alias*=foot/left/odd/offset,%
1869     offset/foot/odd/left/.alias*=foot/left/odd/offset,%
1870     offset/left/foot/odd/.alias*=foot/left/odd/offset,%
1871     offset/left/odd/foot/.alias*=foot/left/odd/offset,%
1872     offset/odd/left/foot/.alias*=foot/left/odd/offset,%
1873     offset/odd/foot/left/.alias*=foot/left/odd/offset,%
1874     foot/right/even/offset/.store in skip cs=f@ncy0@erf,%
1875         foot/right/even/offset/.append code=\fancy@setoffs,%
1876             foot/even/offset/right/.alias*=foot/right/even/offset,%
1877             foot/even/right/offset/.alias*=foot/right/even/offset,%
1878             foot/offset/even/right/.alias*=foot/right/even/offset,%
1879             foot/offset/right/even/.alias*=foot/right/even/offset,%
1880             foot/right/offset/even/.alias*=foot/right/even/offset,%
1881             right/foot/offset/even/.alias*=foot/right/even/offset,%
1882             right/foot/even/offset/.alias*=foot/right/even/offset,%
1883             right/offset/foot/even/.alias*=foot/right/even/offset,%
1884             right/offset/even/foot/.alias*=foot/right/even/offset,%
1885             right/even/offset/foot/.alias*=foot/right/even/offset,%
1886             right/even/foot/offset/.alias*=foot/right/even/offset,%
1887             even/foot/offset/right/.alias*=foot/right/even/offset,%
1888             even/foot/right/offset/.alias*=foot/right/even/offset,%
1889             even/offset/foot/right/.alias*=foot/right/even/offset,%
1890             even/offset/right/foot/.alias*=foot/right/even/offset,%
1891             even/right/offset/foot/.alias*=foot/right/even/offset,%
1892             even/right/foot/offset/.alias*=foot/right/even/offset,%
1893             offset/foot/right/even/.alias*=foot/right/even/offset,%
1894             offset/foot/even/right/.alias*=foot/right/even/offset,%
1895             offset/right/foot/even/.alias*=foot/right/even/offset,%
1896             offset/right/even/foot/.alias*=foot/right/even/offset,%
1897             offset/even/right/foot/.alias*=foot/right/even/offset,%
1898             offset/even/foot/right/.alias*=foot/right/even/offset,%
1899     foot/right/odd/offset/.store in skip cs=f@ncy0@orf,%
1900         foot/right/odd/offset/.append code=\fancy@setoffs,%
1901             foot/odd/offset/right/.alias*=foot/right/odd/offset,%
1902             foot/odd/right/offset/.alias*=foot/right/odd/offset,%
1903             foot/offset/odd/right/.alias*=foot/right/odd/offset,%
1904             foot/offset/right/odd/.alias*=foot/right/odd/offset,%
1905             foot/right/offset/odd/.alias*=foot/right/odd/offset,%
1906             right/foot/offset/odd/.alias*=foot/right/odd/offset,%
1907             right/foot/odd/offset/.alias*=foot/right/odd/offset,%
1908             right/offset/foot/odd/.alias*=foot/right/odd/offset,%
1909             right/offset/odd/foot/.alias*=foot/right/odd/offset,%
1910             right/odd/offset/foot/.alias*=foot/right/odd/offset,%
1911             right/odd/foot/offset/.alias*=foot/right/odd/offset,%
1912             odd/foot/offset/right/.alias*=foot/right/odd/offset,%
1913             odd/foot/right/offset/.alias*=foot/right/odd/offset,%
1914             odd/offset/foot/right/.alias*=foot/right/odd/offset,%
1915             odd/offset/right/foot/.alias*=foot/right/odd/offset,%
1916             odd/right/offset/foot/.alias*=foot/right/odd/offset,%
1917             odd/right/foot/offset/.alias*=foot/right/odd/offset,%
```

```

1918     offset/foot/right/odd/.alias*=foot/right/odd/offset,%
1919     offset/foot/odd/right/.alias*=foot/right/odd/offset,%
1920     offset/right/foot/odd/.alias*=foot/right/odd/offset,%
1921     offset/right/odd/foot/.alias*=foot/right/odd/offset,%
1922     offset/odd/right/foot/.alias*=foot/right/odd/offset,%
1923     offset/odd/foot/right/.alias*=foot/right/odd/offset,%
1924 head/inner/offset/.style={head/right/even/offset={#1},head/left/odd/offset={#1}},%
1925 head/inner/offset+/.style={head/right/even/offset+={#1},head/left/odd/offset+={#1}},%
1926 head/inner/offset-/.style={head/right/even/offset-={#1},head/left/odd/offset-={#1}},%
1927 head/inner/offset*/.style={head/right/even/offset*={#1},head/left/odd/offset*={#1}},%
1928     head/offset/inner/.alias*=head/inner/offset,%
1929     offset/head/inner/.alias*=head/inner/offset,%
1930     offset/inner/head/.alias*=head/inner/offset,%
1931     inner/head/offset/.alias*=head/inner/offset,%
1932     inner/offset/head/.alias*=head/inner/offset,%
1933 head/outer/offset/.style={head/left/even/offset={#1},head/right/odd/offset={#1}},%
1934 head/outer/offset+/.style={head/left/even/offset+={#1},head/right/odd/offset+={#1}},%
1935 head/outer/offset-/.style={head/left/even/offset-={#1},head/right/odd/offset-={#1}},%
1936 head/outer/offset*/.style={head/left/even/offset*={#1},head/right/odd/offset*={#1}},%
1937     head/offset/outer/.alias*=head/outer/offset,%
1938     offset/head/outer/.alias*=head/outer/offset,%
1939     offset/outer/head/.alias*=head/outer/offset,%
1940     outer/head/offset/.alias*=head/outer/offset,%
1941     outer/offset/head/.alias*=head/outer/offset,%
1942 foot/inner/offset/.style={foot/right/even/offset={#1},foot/left/odd/offset={#1}},%
1943 foot/inner/offset+/.style={foot/right/even/offset+={#1},foot/left/odd/offset+={#1}},%
1944 foot/inner/offset-/.style={foot/right/even/offset-={#1},foot/left/odd/offset-={#1}},%
1945 foot/inner/offset*/.style={foot/right/even/offset*={#1},foot/left/odd/offset*={#1}},%
1946     foot/offset/inner/.alias*=foot/inner/offset,%
1947     offset/foot/inner/.alias*=foot/inner/offset,%
1948     offset/inner/foot/.alias*=foot/inner/offset,%
1949     inner/foot/offset/.alias*=foot/inner/offset,%
1950     inner/offset/foot/.alias*=foot/inner/offset,%
1951 foot/outer/offset/.style={foot/left/even/offset={#1},foot/right/odd/offset={#1}},%
1952 foot/outer/offset+/.style={foot/left/even/offset+={#1},foot/right/odd/offset+={#1}},%
1953 foot/outer/offset-/.style={foot/left/even/offset-={#1},foot/right/odd/offset-={#1}},%
1954 foot/outer/offset*/.style={foot/left/even/offset*={#1},foot/right/odd/offset*={#1}},%
1955     foot/offset/outer/.alias*=foot/outer/offset,%
1956     offset/foot/outer/.alias*=foot/outer/offset,%
1957     offset/outer/foot/.alias*=foot/outer/offset,%
1958     outer/foot/offset/.alias*=foot/outer/offset,%
1959     outer/offset/foot/.alias*=foot/outer/offset,%
1960 inner/offset/.style={foot/inner/offset={#1},head/inner/offset={#1}},%
1961 inner/offset+/.style={foot/inner/offset+={#1},head/inner/offset+={#1}},%
1962 inner/offset-/.style={foot/inner/offset-={#1},head/inner/offset-={#1}},%
1963 inner/offset*/.style={foot/inner/offset*={#1},head/inner/offset*={#1}},%
1964     offset/inner/.alias*=inner/offset,%
1965 outer/offset/.style={foot/outer/offset={#1},head/outer/offset={#1}},%
1966 outer/offset+/.style={foot/outer/offset+={#1},head/outer/offset+={#1}},%
1967 outer/offset-/.style={foot/outer/offset-={#1},head/outer/offset-={#1}},%
1968 outer/offset*/.style={foot/outer/offset*={#1},head/outer/offset*={#1}},%
1969     offset/outer/.alias*=outer/offset,%
1970 reset/.style={norule,head=,foot=,offset=\z@,foot/rule/skip,rule/font=,font=},%
1971     reset/.value forbidden,%
1972 }% keys for \pageSetup (stored into token \itf@tk)
1973 \AtBeginDocument{%
1974 \ifdim\headwidth<\z@
1975 %% This catches the case that \headwidth hasn't been initialized and the
1976 %% case that the user added something to \headwidth in the expectation that
1977 %% it was initialized to \textwidth. We compensate this now. This loses if
1978 %% the user intended to multiply it by a factor. But that case is more
1979 %% likely done by saying something like \headwidth=1.2\textwidth.
1980 %% The doc says you have to change \headwidth after the first call to
1981 %% \pagestyle{fancy}. This code is just to catch the most common cases where
1982 %% that requirement is violated.

```

```
1983     \global\advance\headwidth123456789sp\global\advance\headwidth\textwidth
1984 \fi}
1985 
```

21.10 marks

```
1986 {*marks}
1987 \ProvidesPackage{interfaces-marks}
1988 [2010/12/27 v2.6 easier use of e-TeX marks (FC)]
1989 \RequirePackage{interfaces-base}
```

```
\marksthe
```

```
1990 \newrobustcmd\marksthe[2]{%
1991   \ifcsname marks@\#1\endcsname \else \itf@marks@new{\#1}\fi
1992   {\toks@\#2\expandafter\marks\csname marks@\#1\endcsname{\the\toks@}}%
1993 }% \marksthe
```

```
\marksthecs
```

```
1994 \newrobustcmd*\marksthecs[2]{%
1995   \ifcsname marks@\#1\endcsname \else \itf@marks@new{\#1}\fi
1996   \ifcsundef{\#2}
1997   {}
1998   {{\toks@\expandafter\expandafter\expandafter{%
1999     \csname #2\endcsname}%
2000     \expandafter\marks\csname marks@\#1\endcsname{\the\toks@}}}}%
2001 }% \marksthecs
```

```
\itf@marks@new
```

```
2002 \def\itf@marks@new#1{%
2003   \expandafter\newmarks\csname marks@\#1\endcsname
2004   \PackageInfo{interfaces-marks}{marks '#1' defined}%
2005 }% \itf@marks@new
```

`\thefirstmarks` Purely expandable in exactly one step of expansion.

```
\thebotmarks 2006 \newcommand*\thefirstmarks[1]{\romannumeral
\thetopmarks 2007   \ifcsname marks@\#1\endcsname \expandafter\z@
2008     \firstmarks\csname marks@\#1\expandafter\endcsname
2009   \else \expandafter\z@
2010   \fi
2011 }% \thefirstmarks
2012 \newcommand*\thebotmarks[1]{\romannumeral
2013   \ifcsname marks@\#1\endcsname \expandafter\z@
2014     \botmarks\csname marks@\#1\expandafter\endcsname
2015   \else \expandafter\z@
2016   \fi
2017 }% \thebotmarks
2018 \newcommand*\thetopmarks[1]{\romannumeral
2019   \ifcsname marks@\#1\endcsname \expandafter\z@
2020     \topmarks\csname marks@\#1\expandafter\endcsname
2021   \else \expandafter\z@
2022   \fi
2023 }% \thetopmarks
```

```
\getthemarks \getthemarks\firstmarks{\<named-mark>}\result
```

```
2024 %% \getthemarks\firstmarks{section}\macro
2025 \newrobustcmd*\getthemarks[3]{%
2026   \ifcsname marks@\#2\endcsname
2027     \edef#3{\unexpanded\expandafter{\#1\csname marks@\#2\endcsname}}%
2028   \else \let#3\undefined
2029   \fi
2030 }% \getthemarks
```

```
\getthefirstmarks \getthefirstmarks{\<named-mark>}\result: shortcut for \getthemarks\firstmarks
```

```
\getthebotmarks
\gettethopmarks
2031 \newrobustcmd*\getthefirstmarks{\gettethmarks\firstmarks}
2032 \newrobustcmd*\getthebotmarks{\gettethmarks\botmarks}
2033 \newrobustcmd*\gettethopmarks{\gettethmarks\topmarks}

\ifmarksequal
2034 %% \ifmarksequal\firstmarks\botmarks{subsection}{ true }{ false }
2035 \newrobustcmd*\ifmarksequal[3]{%
2036   \begin{group}
2037     \gettethmarks{#1}{#3}\x
2038     \gettethmarks{#2}{#3}\y
2039   \expandafter\endgroup
2040   \csname ltx@\ifdefined\x\ifdefined\y
2041     \ifx\x\y first\else
2042     second\fi\else second\fi\else second\fi
2043   \oftwo\endcsname
2044 }% \ifmarksequal
2045 
```

21.11 † Interface ⇔ `tocloft` : \tocsetup, \lofsetup, \lotsetup and \listofsetup

```
2046 {*tocloft}
2047 \ProvidesPackage{interfaces-tocloft}
2048   [2010/12/27 v2.6 pgfkeys macros for tocloft (FC)]
2049 \RequirePackage{interfaces-base}
2050 \itf@Check@Package{tocloft}
```

The user commands.

```
\tocsetup
\lofsetup
\lotsetup
\listofsetup
2051 \newrobustcmd\tocsetup[1]{%
2052   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/toc}{#1}}%
2053 }% \tocsetup
2054 \newrobustcmd\lofsetup[1]{%
2055   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/lof}{#1}}%
2056 }% \lofsetup
2057 \newrobustcmd\lotsetup[1]{%
2058   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/lot}{#1}}%
2059 }% \lotsetup
2060 \newrobustcmd\listofsetup[2]{%
2061   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/#1}{#2}}%
2062 }% \listofsetup
```

`\itf@cft@newlistof` This macro creates the keys for `toc`, `lof` and `lot`, and is able to create the keys for each new list created with `tocloft`'s `\newlistof`.

```
2063 \def\itf@cft@newlistof#1#2#3{%
2064   #1=toc/lof (ext.), #2=entry (\newlistof only), #3=title (%
2065   \pgfqkeys{/itf/tocloft/#1}{%
2066     depth/.store in counter cs=c@#1depth,%
2067     depth/.append code=\expandafter\global\csname c@#1depth\endcsname
2068     \csname c@#1depth\endcsname
2069     \addtocontents{#1}{\csname c@#1depth\endcsname
2070     \the\csname c@#1depth\endcsname\relax},%
2071     #1depth/.alias*=depth,%
2072     pagestyle/.code=\tocloftpagestyle{##1},%
2073     parskip/.store in skip cs=cftparskip,%
2074     parsep/.alias=parskip,%
2075     mark/.store in cs=cftmark#1,%
2076     marks/.alias=mark,%
2077     mark=,%
2078     label/.store as code,%
2079     ref/.store as code,%
2080     ref/.initial=\csname cft#1title\endcsname,% overwritten (toc,lof,lot)
2081     ref/.default=\csname cft#1title\endcsname,% overwritten (toc,lof,lot)
2082     ref/.blank=\pgfkeysnovalue,%
```

```

2082     title/.store as code in cs=cft#1title,% overwritten below for toc, lof and lot
2083     title/top/.store in skip cs=cftbefore#1titleskip,%
2084         beforetitleskip/.alias*=title/top,%
2085     title/bottom/.store in skip cs=cftafter#1titleskip,%
2086         aftertitleskip/.alias*=title/bottom,%
2087     title/font/.store as code in cs=cft#1titlefont,%
2088         title/format/.alias*=title/font,%
2089     title/color/.store as color,%
2090         title/color/.append code=\pgfkeysalso{/itf/tocloft/#1/title/font+={%
2091             \color{\pgfkeysvalueof{/itf/tocloft/#1/title/color}}}},%
2092         title/font/color/.alias=title/color,%
2093         title/format/color/.alias=title/color,%
2094     title/after/.store as code in cs=cftafter#1title,%
2095     title/command/.code=\csdef{@cftmake#1title}####1{##1},%
2096     dotsep/.code=\itf@units@mu\itf@musk{##1}% converts to mu
2097         \edef\cftdotsep{\itf@strip@mu\itf@musk},%
2098         dotsep/.default=4.5mu,%
2099         dotsep/.blank=\cftnodots,%
2100     dot/.store in=\cftdot,%
2101     before/.store as code,%
2102     after/.store as code,%
2103     columns/before@hook/.is private=interfaces-tocloft,%
2104     columns/after@hook/.is private=interfaces-tocloft,%
2105     columns/before@hook/.initial=,%
2106     columns/after@hook/.initial=,%
2107     columns/.code={\ltx@ifpackageloaded{multicol}{%
2108         \pgfkeysEsetvalue\pgfkeyscurrentkey{\number##1}%
2109         \ifnum\number##1>\@ne
2110             \pgfkeysgetvalue{/itf/tocloft/#1/columns/before@hook}\itf@tp
2111             \ifx\itf@tp\@empty
2112                 \pgfkeyssetvalue{/itf/tocloft/#1/columns/before@hook}{%
2113                     \begingroup
2114                         \multicols\pgfkeysvalueof{/itf/tocloft/#1/multicolsep}\relax
2115                         \columnsep\pgfkeysvalueof{/itf/tocloft/#1/columns/sep}\relax
2116                         \columnseprule\pgfkeysvalueof{/itf/tocloft/#1/columns/rule}\relax
2117                         \begin{multicols}{\pgfkeysvalueof{/itf/tocloft/#1/columns}}%
2118                             \pgfkeysgetvalue{/itf/tocloft/#1/columns/rule/color}\columnseprulecolor
2119                             \pgfkeysvalueof{/itf/tocloft/#1/multicols/begin}%
2120                         }%
2121                         \pgfkeyssetvalue{/itf/tocloft/#1/columns/after@hook}{%
2122                             \pgfkeysvalueof{/itf/tocloft/#1/multicols/end}%
2123                             \end{multicols}
2124                             \endgroup
2125                         }%
2126                     \fi
2127                 \else
2128                     \pgfkeyslet\pgfkeyscurrentkey\@ne
2129                     \pgfkeyslet{/itf/tocloft/#1/columns/before@hook}\@empty
2130                     \pgfkeyslet{/itf/tocloft/#1/columns/after@hook}\@empty
2131                 \fi}
2132                 {\PackageError{interfaces-tocloft}%
2133                     {Key ‘columns’: package multicol is required}\@eha}%
2134 },%
2135     columns/.default=2,columns/.blank=1,%
2136         multicols/.alias=columns,%
2137         multicols/.alias=columns,%
2138     twocolumns/.true=\pgfkeysalso{columns=2},%
2139     twocolumns/.false=\pgfkeysalso{columns=1},%
2140     columns/rule/.estore as dimen,%
2141     columns/rule/.default=\columnseprule,%
2142     columns/rule,% assign default
2143         columnseprule/.alias*=columns/rule,%
2144         columns/rule/width/.alias*=columns/rule,%
2145     columns/sep/.estore as dimen,%
2146     columns/sep/.default=\columnsep,%

```

```

2147     columns/sep/.blank=10\p@,%
2148     columns/sep,% assign default
2149         columnsep/.alias*=columns/sep,%
2150         columns/rule/color/.store as color command,%
2151         columns/rule/color,% assign default (\normalcolor)
2152             columnseprulecolor/.alias*=columns/rule/color,%
2153             columns/rulecolor/.alias*=columns/rule/color,%
2154             columns/color/.alias*=columns/rule/color,%
2155             multicolsep/.estore as glue,%
2156                 multicolsep/.default=0pt plus4pt minus3pt,%
2157                 multicolsep,% sets default
2158             multicols/begin/.initial=,multicols/begin/.default=,% 
2159                 multicols/before/.alias=multicols/begin,% 
2160             multicols/end/.initial=,multicols/end/.default=,% 
2161                 multicols/after/.alias=multicols/end,% 
2162 }% \pgfkeys
2163 \itf@AfterPackage{interfaces-bookmark}{%
2164 \pgfqkeys{/itf/tocloft/#1}{%
2165     bookmark/.is multiple choice,%
2166     bookmark:bold/.true=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{,bold=true},%
2167     bookmark:bold/.false=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{,bold=false},%
2168     bookmark/italic/.true=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{,italic=true},%
2169     bookmark/italic/.false=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{,italic=false},%
2170     bookmark/color/.code=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{,color={##1}},%
2171     bookmark/level/.code=\pgfkeysEaddvalue{/itf/tocloft/#1/bookmark}{}{,level=%
2172         \ifcsname toclevel##1\endcsname
2173             \csname toclevel##1\endcsname
2174         \else##1\fi}}},%
2175     bookmark/text/.store as code,%
2176         bookmark/text/.default=\contentsname,%
2177     bookmark/.unknown/.code=\pgfkeysEaddvalue{/itf/tocloft/#1/bookmark}{}{\pgfkeyscurrentname={##1}},%
2178     bookmark/.initial=,% 
2179 }% \pgfkeys
2180 }% AfterPackage interfaces-bookmark
2181 %% while making a format \jobname is expanded to the name of the format...
2182 %% Therefore, the following commands must be expanded \AtEndPreamble
2183 \AfterPreamble{%
2184     \BeforeFile{\jobname.#1}{%
2185         \pgfkeysvalueof{/itf/tocloft/#1/before}%
2186         \pgfkeysvalueof{/itf/tocloft/#1/columns/before@hook}%
2187     }%
2188     \AfterFile{\jobname.#1}{%
2189         \pgfkeysvalueof{/itf/tocloft/#1/columns/after@hook}%
2190         \pgfkeysvalueof{/itf/tocloft/#1/after}%
2191     }%
2192 }%
2193 }%
2194 \ifcsname #1setup\endcsname\else
2195     \pgfqkeys{/itf/tocloft/#1}{%
2196         title={#3},%
2197         pagenumbers/.true=@cftpnumon{#2},%
2198         pagenumbers/.false=@cftpnumoff{#2},%
2199             pagenumbers/on/.alias=pagenumbers/true,
2200             pagenumbers/off/.alias=pagenumbers/false,
2201             pagenumber/.alias=pagenumbers,% 
2202     }%
2203     \expandafter\pretocmd\csname@cftmake#1title\expandafter\endcsname
2204         \expandafter{\expandafter\itf@tocloft\csname cft#1title\endcsname{#1}{listof#2}}%
2205     {\@PackageInfoNoLine{interfaces-tocloft}
2206         {Command \expandafter\string\csname@cftmake#1title\endcsname\space
2207             successfully augmented}}
2208     {\@PackageWarningNoLine{interfaces-tocloft}
2209         {Command \expandafter\string\csname@cftmake#1title\endcsname\space
2210             \space could not be patched
2211             \MessageBreak Keys label and bookmark won't work for \string\listofsetup}}%

```

```
2212   \fi
2213 }% \itf@cft@newlistof
```

Then create the keys for `toc`, `lof`, and `lot`:

```
2214 \itf@cft@newlistof{toc}{}{}
2215 \itf@cft@newlistof{lof}{}{}
2216 \itf@cft@newlistof{lot}{}{}

2217 %% Specific to toc only -----
2218 \pgfqkeys{/itf/tocloft/toc}{%
2219   title/.store as code in cs=contentsname,%
2220   contentsname/.alias=title,%
2221   ref/.initial=\contentsname,%
2222   ref/.default=\contentsname,%
2223   pagenumbers/.style={% boolean key
2224     part/pagenumbers={#1}, chapter/pagenumbers={#1}, section/pagenumbers={#1}, %
2225     subsubsection/pagenumbers={#1}, subsection/pagenumbers={#1}, %
2226     paragraph/pagenumbers={#1}, subparagraph/pagenumbers={#1}, %
2227     pagenumber/.alias=pagenumbers,%
2228   }%
2229 %% Specific to lof and lot -----
2230 \pgfqkeys{/itf/tocloft/lof}{%
2231   title/.store as code in cs=listfigurename,%
2232   listfigurename/.alias=title,%
2233   ref/.initial=\listfigurename,%
2234   ref/.default=\listfigurename,%
2235 }%
2236 \pgfqkeys{/itf/tocloft/lot}{%
2237   title/.store as code in cs=listtablename,%
2238   listtablename/.alias=title,%
2239   ref/.initial=\listtablename,%
2240   ref/.default=\listtablename,%
2241 }%
2242 \if@cftsubfigopt
2243 \pgfqkeys{/itf/tocloft}{%
2244   lof/pagenumbers/.style={figure/pagenumbers={#1}, subfigure/pagenumbers={#1}}, %
2245   lof/pagenumber/.alias=lof/pagenumbers,%
2246   lot/pagenumbers/.style={figure/pagenumbers={#1}, subfigure/pagenumbers={#1}}, %
2247   lot/pagenumber/.alias=lot/pagenumbers,%
2248 }%
2249 \else
2250 \pgfqkeys{/itf/tocloft}{%
2251   lof/pagenumbers/.style={figure/pagenumbers={#1}}, %
2252   lof/pagenumber/.alias=lof/pagenumbers,%
2253   lot/pagenumbers/.style={figure/pagenumbers={#1}}, %
2254   lot/pagenumber/.alias=lot/pagenumbers,%
2255 }%
2256 \fi
```

\itf@cft@sectionname This macro converts `subsection` into `subsec` etc.. because of `tocloft` command names:
`\cftbeforesecskip`, `\cftsubsecindent` etc..

And the keys are either `subsec` or `subsection`.

```
2257 \def\itf@cft@name#1.#2.#3@nil{\cft#1\itf@cft@secname{#2}#3}
2258 \def\itf@cft@secname#1{%
2259   \expandafter\itf@cft@sec@name\csname#1\endcsname{#1}%
2260 }% \itf@cft@secname
2261 \def\itf@cft@sec@name#1#2{%
2262   \ifx#1\relax #2\else
2263   \ifx#1\section sec\else
2264     \ifx#1\subsection subsec\else
2265       \ifx#1\subsubsection subsubsec\else
2266         \ifx#1\paragraph para\else
2267           \ifx#1\subparagraph subpara\else
```

```

2268         \ifx#1\figure fig\else
2269             \ifx#1\subsubfigure subfig\else
2270                 \ifx#1\table tab\else
2271                     \ifx#1\subtable subtab\else
2272                         \ifx#1\chapter chap\else
2273                             #2\fi
2274             \fi\fi\fi\fi\fi\fi\fi
2275 }% \itf@cft@section@name

2276 % tocloft setup for tables sections :
2277 \pgfkeys{%
2278 /handlers/.cft@def/.style={%
2279     \pgfkeyscurrentpath/.store in cs=\itf@cft@name #1@nil},%
2280 /handlers/.cft@code/.style={%
2281     \pgfkeyscurrentpath/.store as code in cs=\itf@cft@name #1@nil},%
2282 /handlers/.cft@length/.style={%
2283     \pgfkeyscurrentpath/.store in skip cs=\itf@cft@name #1@nil},%
2284 }

```

`\itf@cft@newsection` This macro creates the keys for each sectionning command.

```

2285 \def\itf@cft@newsection{\itf@cft@new@section{/itf/tocloft}}
2286 \def\itf@cft@new@section#1#2#3{%
2287     \pgfkeys{#1/#2/#3}{%
2288         skip/.cft@length=before.#3.skip,%  

2289         beforeskip/.alias*=skip,%  

2290         top/.alias*=skip,%  

2291         indent/.cft@length=.#3.indent,%  

2292         font/.cft@code=.#3.font,%  

2293         font/.append code=\pgfkeysalso{/itf/tocloft/#2/#3/pagenumber/font={##1}},%  

2294         format/.alias*=font,%  

2295         color/.store as color,%  

2296         color/.append code=\pgfkeysalso{/itf/tocloft/#2/#3/font+=%  

2297             \color{\pgfkeysvalueof{/itf/tocloft/#2/#3/color}}}%  

2298         \ifdefined\hypersetup \hypersetup{%
2299             linkcolor=\pgfkeysvalueof{/itf/tocloft/#2/#3/color}}%  

2300         \fi},%  

2301         font/color/.alias=color,%  

2302         format/color/.alias=color,%  

2303         number/width/.cft@length=.#3.numwidth,%  

2304         numwidth/.alias*=number/width,%  

2305         presnum/.cft@code=.#3.presnum,%  

2306         number/before/.alias*=presnum,%  

2307         aftersnum/.cft@code=.#3.aftersnum,%  

2308         number/after/.alias*=aftersnum,%  

2309         aftersnumb/.cft@code=.#3.aftersnumb,%  

2310         title/before/.alias*=aftersnumb,%  

2311         leader/.cft@def=.#3.leader,%  

2312         leader/.default=\cftdotfill{\csname\itf@cft@name.#3.dotsep@nil\endcsname},%  

2313         leaders/.alias=leader,%  

2314         dotsep/.code={\ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text  

2315             \csdef{\itf@cft@name.#3.dotsep@nil}{\cftdotsep}%  

2316             \else  

2317                 \itf@units@mu\itf@musk{##1}%  

2318                 \csedef{\itf@cft@name.#3.dotsep@nil}{\itf@strip@mu\itf@musk}%  

2319             \fi},%  

2320             dotsep/.blank=\cftnodots,%  

2321             nodots/.code=\csdef{\itf@cft@name.#3.dotsep@nil}{\cftnodots},%  

2322             nodots/.value forbidden,%  

2323             pagefont/.cft@code=.#3.pagefont,%  

2324             page/font/.alias*=pagefont,%  

2325             pagenumbers/font/.alias*=pagefont,%  

2326             pagenumbers/font/.alias*=pagefont,%  

2327             page/color/.store as color,%  

2328             page/color/.append code=\pgfkeysalso{/itf/tocloft/#2/#3/pagefont+=%  

2329                 \color{\pgfkeysvalueof{/itf/tocloft/#2/#3/page/color}}}}%

```

```

2330          \ifdefined\hypersetup \hypersetup{%
2331              linkcolor=\pgfkeysvalueof{/itf/tocloft/#2/#3/page/color}}%
2332          \fi},%
2333          pagenumber/color/.alias=page/color,%
2334          pagenumbers/color/.alias=page/color,%
2335          page/font/color/.alias=page/color,%
2336          pagenumber/font/color/.alias=page/color,%
2337          pagenumbers/font/color/.alias=page/color,%
2338          afterpnum/.cft@code=.#3.afterpnum,%
2339          page/after/.alias*=afterpnum,%
2340          pagenumber/after/.alias*=afterpnum,%
2341          pagenums/.on=@cftpnumon{\itf@cft@secname{#3}},%
2342          pagenums/.off=@cftpnumoff{\itf@cft@secname{#3}},%
2343          pagenumber/.alias=pagenums,%
2344      }% \pgfqkeys
2345 }% \itf@cft@new@section

2346 \@for\itf@tp:=part, chap, chapter, sec, section, subsec, subsection,%
2347             subsubsec, subsubsection, para, paragraph, subparagraph,%
2348 \do{\edef\itf@tp{\noexpand\itf@cft@newsection{toc}{\itf@tp}}\itf@tp}%
2349 \if@cftsubfigopt
2350     \@for\itf@tp:=fig, figure, subfig, subfigure%
2351         \do{\edef\itf@tp{\noexpand\itf@cft@newsection{lof}{\itf@tp}}\itf@tp}%
2352     \@for\itf@tp:=tab, table, subtab, subtable%
2353         \do{\edef\itf@tp{\noexpand\itf@cft@newsection{lot}{\itf@tp}}\itf@tp}%
2354 \else
2355     \@for\itf@tp:=fig, figure%
2356         \do{\edef\itf@tp{\noexpand\itf@cft@newsection{lof}{\itf@tp}}\itf@tp}%
2357     \@for\itf@tp:=tab, table%
2358         \do{\edef\itf@tp{\noexpand\itf@cft@newsection{lot}{\itf@tp}}\itf@tp}%
2359 \fi

%@cftmaketoctitle (patch)
\@cftmakeloftitle (patch)
2360 \itf@patch\pretocmd@cftmaketoctitle
\@cftmakelottitle (patch)
2361 {\itf@tocloft\contentsname{toc}{tableofcontents}}
\\newlistof (patch)
2362 {\@PackageInfoNoLine{interfaces-tocloft}
2363     {Command \string@\cftmaketoctitle\space successfully augmented}}
2364 {\@PackageWarningNoLine{interfaces-tocloft}
2365     {Command \string@\cftmaketoctitle\space could not be patched
2366      \MessageBreak label and bookmark keys for \string\tocsetup\space won't work}}%
2367 \itf@patch\pretocmd@cftmakeloftitle
2368 {\itf@tocloft\listfigurename{lof}{listoffigures}}
2369 {\@PackageInfoNoLine{interfaces-tocloft}
2370     {Command \string@\cftmakeloftitle\space successfully augmented}}
2371 {\@PackageWarningNoLine{interfaces-tocloft}
2372     {Command \string@\cftmakeloftitle\space could not be patched
2373      \MessageBreak label and bookmark keys for \string\lofsetup\space won't work}}%
2374 \itf@patch\pretocmd@cftmakelottitle
2375 {\itf@tocloft\listtablename{lot}{listoftables}}
2376 {\@PackageInfoNoLine{interfaces-tocloft}
2377     {Command \string@\cftmakelottitle\space successfully augmented}}
2378 {\@PackageWarningNoLine{interfaces-tocloft}
2379     {Command \string@\cftmakelottitle\space could not be patched
2380      \MessageBreak label and bookmark keys for \string\lotsetup\space won't work}}%
2381 \expandafter\itf@patch\expandafter\patchcmd\csname\string\newlistof\endcsname
2382 {#4}\csname cft#3title\endcsname
2383 {\@PackageInfoNoLine{interfaces-tocloft}
2384     {Command \string\newlistof\space successfully patched}}
2385 {\@PackageWarningNoLine{interfaces-tocloft}
2386     {Command \string\newlistof\space could not be patched
2387      \MessageBreak \string\listofsetup\space will not work}}%
2388 \expandafter\itf@patch\expandafter\apptocmd\csname\string\newlistof\endcsname
2389 {\itf@cft@newlistof{#3}{#2}{#4}}
2390 {\@PackageInfoNoLine{interfaces-tocloft}
2391     {Command \string\newlistof\space successfully augmented}}

```

```

2392   { \@PackageWarningNoLine{interfaces-tocloft}
2393     {Command \string\newlistof\space could not be patched
2394      \MessageBreak \string\listofsetup\space will not work} }%
2395 \expandafter\itf@patch\expandafter\apptocmd\csname\string\newlistentry\endcsname
2396   {\itf@cft@newsection{\#3}{\#2}}
2397   { \@PackageInfoNoLine{interfaces-tocloft}
2398     {Command \string\newlistentry\space successfully augmented} }
2399   { \@PackageWarningNoLine{interfaces-tocloft}
2400     {Command \string\newlistentry\space could not be patched
2401      \MessageBreak \string\listofsetup\space will not work} }

\itf@cft@label    \itf@cft@label insert the anchor for the table of contents and makes the label (if the key label of
                   \tocsetup, \lofsetup etc. has been set to a non empty value.

2402 \def\itf@tocloft#1#2#3{%
2403   \expandafter\def\expandafter#1\expandafter{%
2404     \expandafter\itf@cft@label
2405       \expandafter#1\expandafter{#1}{#2}{#3}}%
2406 }% \itf@tocloft
2407 \long\def\itf@cft@label#1#2#3#4{%
2408   \def#1{#2}% restore original
2409   \begingroup
2410   \pgfkeysifdefined{/itf/tocloft/#3/bookmark}
2411   {%
2412     \raisedhyperdef{interfaces}{#4}{#2}%
2413     \pgfkeysgetvalue{/itf/tocloft/#3/bookmark/text}\itf@tp
2414     \ifx\itf@tp\@empty\else
2415       \pgfkeysgetvalue{/itf/tocloft/#3/bookmark}\@tempa
2416       \expandafter\bookmark\expandafter[%
2417         \expandafter{\@tempa,dest=interfaces.#4}]%
2418         \pgfkeysvalueof{/itf/tocloft/#3/bookmark/text}}%
2419       \global\BKM@startatroottrue
2420   \fi
2421   {#2}%
2422   \pgfkeysgetvalue{/itf/tocloft/#3/label}\itf@tp
2423   \ifx\itf@tp\@empty\else
2424     \def@\currentHref{interfaces.#4}%
2425     \pgfkeysgetvalue{/itf/tocloft/#3/ref}\@currentlabel
2426     \def@\currentlabelname{#2}%
2427     \label{\pgfkeysvalueof{/itf/tocloft/#3/label}}%
2428   \fi
2429   \endgroup
2430   \gdef\itf@tp{\def#1{#2}\let\itf@tp\@undefined}% restore original
2431   \aftergroup\itf@tp % simple group introduced by tocloft to print \contentsname
2432 }% \itf@cft@label

2433 </tocloft>

```

21.12 Interface `makecell`: `\repeatcell`

```

2434 (*makecell)
2435 \ProvidesPackage{interfaces-makecell}
2436   [2010/12/27 v2.6 repeatcell to fill in tabulars (FC)]
2437 \RequirePackage{interfaces-base}
2438 \itf@Check@Package{makecell}
2439 \edef\itf@AtEnd{\catcode126 \the\catcode126}% 126 = ~
2440 \AtEndOfPackage{\itf@AtEnd}
2441 \catcode126 \active % 126 = ~

```

21.12.1 `countvalues`

```

\countvalues

2442 \def\countvalues#1{\number\numexpr
2443   \expandafter\itf@countvalues\expandafter\relax\#1,\countvalues\@nnil
2444 }% \countvalues

```

```

2445 \def\itf@countvalues#1#2,#3{%
2446   \ifx\countvalues#3%
2447     +#1\expandafter\remove@to@nnil
2448   \else
2449     +1\expandafter\itf@countvalues
2450   \fi 1#3%
2451 }% \itf@countvalues

\itf@getvalue

2452 \def\itf@getvalue #1{\romannumeral -`\\q
2453   \expandafter\itf@getv@lue \expandafter#1,\itf@getvalue}
2454 \def\itf@getv@lue#1#2\itf@getvalue #3{\itf@getv@lu@{#1}{#3-1}#2\itf@getvalue@nnil}
2455 \def\itf@getv@lu@#1#2#3,#4{%
2456   \ifnum \numexpr#1=\numexpr#2\relax
2457     #3\expandafter\remove@to@nnil
2458   \else\ifx\itf@getvalue#4\@empty
2459     \expandafter\expandafter
2460     \expandafter\remove@to@nnil
2461   \else \expandafter\expandafter
2462     \expandafter\itf@getv@lu@
2463   \fi\fi{#1+1}{#2}#4%
2464 }% \itf@getv@lu@

```

21.12.2 \repeatcell

```

\repeatcell

2465 \newif\ifitf@celltranspose
2466 \newrobustcmd*\repeatcell{\begingroup \iffalse{\fi \ifnum0='}\fi \itf@repeatcell}
2467 \newrobustcmd*\itf@repeatcell[3][]{%
2468   \ifblank{#1}{ alignment option ?
2469     {\let\itf@makecell@\firstofone}
2470     {\def\itf@makecell##1{\noexpand\makecell[{#1}] {##1}}}}%
2471   \countdef\itf@col=100% local
2472   \countdef\itf@row=101% local
2473   \countdef\itf@cnt=102% local
2474   \toksdef\itf@cell=1% local
2475   \itf@pgfkeys
2476   \pgfqkeys{/itf/makecell/repeatcell}{@reset={#2},#3}%
2477   \toks@{\ifnum0='{}'\fi \endgroup}% local
2478   \ifnum\itf@row>\z@ % number
2479     \edef\itf@rowloop{\noexpand\@whilenum\itf@row<\the\itf@row}%
2480     \def\itf@cell@init{\def\r{\row}\let\R\@empty}%
2481   \else % list
2482     \def\itf@rowloop{@for\r:=\itf@rows}%
2483     \expandafter\def\expandafter\itf@cell@init\expandafter{%
2484       \expandafter\def\expandafter\R\expandafter{%
2485         \expandafter\itf@getvalue\expandafter{\expandafter\@empty\itf@rows}}}%
2486   \fi
2487   \ifnum\itf@col>\z@ % number
2488     \edef\itf@colloop{\noexpand\@whilenum\itf@col<\the\itf@col}%
2489     \expandafter\def\expandafter\itf@cell@init\expandafter{%
2490       \itf@cell@init\def\c{\column}\let\C\@empty}%
2491   \else % list
2492     \def\itf@colloop{@for\c:=\itf@cols}%
2493     \expandafter\expandafter\expandafter\def
2494       \expandafter\expandafter\expandafter\itf@cell@init
2495       \expandafter\expandafter\expandafter{%
2496         \expandafter\itf@cell@init
2497         \expandafter\def\expandafter\R\expandafter{%
2498           \expandafter\itf@getvalue\expandafter{\expandafter\@empty\itf@cols}}}%
2499   \fi
2500   \itf@cell@init % in case they are used in formulas
2501   \ifitf@celltranspose

```

```

2502     \itf@col\z@
2503     \itf@colloop\do{%
2504         \ifnum\itf@col>\z@
2505             \pgfkeysifdefined{/itf/makecell/repeatcell/cr/row\the\itf@col}{%
2506                 \pgfkeysgetvalue{/itf/makecell/repeatcell/cr/row\the\itf@col}\itf@thecr}%
2507                 \pgfkeysgetvalue{/itf/makecell/repeatcell/cr}\itf@thecr}%
2508             \toks@\expandafter{\the\expandafter\toks@ \itf@thecr}%
2509         \fi
2510         \advance\itf@col\@ne
2511         \itf@row\z@
2512         \itf@rowloop\do{%
2513             \advance\itf@row\@ne
2514             \ifnum\itf@row>\@ne
2515                 \toks@\expandafter{\the\toks@&}%
2516             \fi
2517             \toks@\expandafter{\the\expandafter\toks@ \itf@cell@init}%
2518             \itf@cell@insert}%
2519         }% cols loop
2520     \else
2521     \itf@row\z@
2522     \itf@rowloop\do{%
2523         \ifnum\itf@row>\z@
2524             \pgfkeysifdefined{/itf/makecell/repeatcell/cr/row\the\itf@row}{%
2525                 \pgfkeysgetvalue{/itf/makecell/repeatcell/cr/row\the\itf@row}\itf@thecr}%
2526                 \pgfkeysgetvalue{/itf/makecell/repeatcell/cr}\itf@thecr}%
2527             \toks@\expandafter{\the\expandafter\toks@ \itf@thecr}%
2528         \fi
2529         \advance\itf@row\@ne
2530         \itf@col\z@
2531         \itf@colloop\do{%
2532             \advance\itf@col\@ne
2533             \ifnum\itf@col>\@ne
2534                 \toks@\expandafter{\the\toks@&}%
2535             \fi
2536             \toks@\expandafter{\the\expandafter\toks@ \itf@cell@init}%
2537             \itf@cell@insert}%
2538         }% rows loop
2539     \fi
2540     \itf@message{Code introduced by repeatcell in the \@currenvir: ^[]%
2541                 \the\toks@}%
2542     \the\toks@ \ignorespaces
2543 }% \itf@repeatcell

```

\itf@message

```
2544 \let\itf@message@gobble
```

Now the keys for \repeatcell:

```

2545 \def\itf@cell@rowcolkey{%
2546     \removefromstart\pgfkeyscurrentname{row}%
2547     \ifin@\else
2548         \removefromstart\pgfkeyscurrentname{col}%
2549     \fi
2550     \ifin@
2551         \itf@number\itf@cnt\pgfkeyscurrentname\relax\@nnil
2552             {\pgfkeyslet\pgfkeyscurrentkey\pgfkeyscurrentvalue}%
2553             \in@false
2554     \fi
2555     \ifin@\else
2556         \pgfkeysvalueof{/errors/unknown key/.@cmd}\pgfkeyscurrentkey
2557                         \pgfkeyscurrentvalue
2558                         \pgfeov
2559     \fi
2560 }% \itf@cell@rowcolkey
2561 \pgfqkeys{/itf/makecell/repeatcell}{%

```

```

2562   font/.store in=\itf@cellfont,%
2563   font/.unknown/.code=\itf@cell@rowcolkey,%
2564   text/.store in=\itf@celltext,%
2565     text/.default=\ifdefined\c\c\column\fi,%
2566   text/.unknown/.code=\itf@cell@rowcolkey,%
2567   cr/.initial=\,,%
2568     cr/.default=\crcr,cr/.blank=\,,end/.alias=cr,%
2569   cr/.unknown/.code=\itf@cell@rowcolkey,%
2570   rows/.code=\itf@number\itf@row#1\relax\@nnil
2571     \ifnum\itf@row<\z@\itf@row@ne\fi
2572       {\itf@row\z@
2573         \def\itf@rows{\#1}\itf@expandonceifcs\itf@rows},%
2574   rows/.default=1,rows/.blank=1,%
2575 @cols/.is private=interfaces-makecell,%
2576 column/.is private=interfaces-makecell,%
2577 row/.is private=interfaces-makecell,%
2578 transpose/.is if=itf@celltranspose,%
2579 .unknown/.code=%
2580   \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text \itf@cell@formulaerror{}\z@ \fi
2581   \ifx\pgfkeyscurrentvalue\empty \itf@cell@formulaerror{}\@ne \fi
2582   \pgfkeysifdefined{\pgfkeyscurrentkey@calc}{%
2583     {\itf@cell@formulaerror{}\tw@}
2584     {\pgfkeyslet{\pgfkeyscurrentkey@calc}\relax}%
2585   \itf@cell@getformula{\#1}%
2586   \pgfkeysEaddvalue{/itf/makecell/repeatcell/@calc}{}
2587     {\the\toks@\expandafter\noexpand\csname\pgfkeyscurrentname\endcsname}%
2588   \expandafter\let\csname\pgfkeyscurrentname\endcsname\undefined,%
2589 @reset/.code=% #1 are the columns
2590   \itf@celltransposefalse
2591   \let\itf@celltext\empty
2592   \let\itf@cellfont\empty
2593   \pgfkeyssetvalue{/itf/makecell/repeatcell/cr}{\,}%
2594   \itf@row@ne
2595   \pgfkeyssetvalue{/itf/makecell/repeatcell/@calc}{%
2596     \itf@cell@numcalc\itf@col\column
2597     \itf@cell@numcalc\itf@row\row}%
2598   \itf@number\itf@col#1\relax\@nnil
2599     \ifnum\itf@col<\z@\itf@col@ne\fi
2600     {\itf@col\z@
2601       \def\itf@cols{\#1}\itf@expandonceifcs\itf@cols},%
2602 }% keys for \repeatcell

```

`\itf@cell@getformula` Determines whether the formula should be evaluated with `\numexpr`, with `\pgfmath` or `\FPeval`:

```

2603 \def\itf@cell@getformula#1{%
2604   \itf@cell@getf@rmula #1$$\itf@cell@getformula$\relax\@nnil
2605   \expandafter\ifx\expandafter\toks@\the\toks@
2606     \itf@cell@formulaerror{\#1}\thr@@
2607   \fi
2608 }% \itf@cell@getformula
2609 \def\itf@cell@getf@rmula #1##2##3##4##5##6{%
2610   \toks@{\toks@}%
2611   \ifx\itf@cell@getformula #4%
2612     \toks@{\itf@cell@numcalc{\#1}}%
2613   \else\ifx $#4%
2614     \ifx\itf@cell@getformula #6%
2615       \ifdefined\pgfmathparse\else\itf@cell@formulaerror{\#2}4\fi
2616       \toks@{\itf@cell@pgfcalc{\#2}}%
2617     \else\ifx $#6%
2618       \ifdefined\FPeval\else\itf@cell@formulaerror{\#3}5\fi
2619       \toks@{\itf@cell@FPcalc{\#3}}%
2620     \fi\fi
2621   \fi\fi\remove@to@nnil
2622 }% \itf@cell@getf@rmula
2623 \def\itf@cell@formulaerror#1#2{%

```

```
2624   \def\itf@tp{\#1}@onel level@sanitize\itf@tp
2625   \ifcase#2% 0
2626     \PackageError{interfaces-makecell}
2627       {Missing formula for key \pgfkeyscurrentname!}@\ehd
2628   \or %1
2629     \PackageError{interfaces-makecell}
2630       {Computed key \pgfkeyscurrentname\space was left blank!}@\ehd
2631   \or %2
2632     \PackageError{interfaces-makecell}
2633       {Duplicate definition for computed key \pgfkeyscurrentname
2634         \MessageBreak\string\repeatcell\space will redefine the key}@\eha
2635   \or %3
2636     \PackageError{interfaces-makecell}
2637       {Incorrect formula in \string\repeatcell
2638         \MessageBreak Check: \itf@tp
2639           \MessageBreak Take care of commas and equal signs...}@\ehd
2640   \or %4
2641     \PackageError{interfaces-makecell}
2642       {Formula: \itf@tp
2643         \MessageBreak requires \string\pgfmathparse:\space\string\usetikz{pgfmath}}@\ehd
2644   \or %5
2645     \PackageError{interfaces-makecell}
2646       {Formula: \itf@tp
2647         \MessageBreak requires \string\FPeval:\space\string\usepackage{fp}}@\ehd
2648   \else
2649     \PackageError{interfaces-makecell}
2650       {Impossible value!}@\ehd
2651   \fi
2652 }% \itf@cell@formulaerror
```

```

\itf@cell@calc
\itf@cell@numcalc 2653 \protected\def\itf@cell@calc#1#2{%
\itf@cell@pgfcalc 2654   \edef#1{\def\noexpand#1{#2}}%
\itf@cell@FPCalc 2655   \toks@\expandafter{\the\expandafter\toks@ #1}%
2656   #1%
2657 }% \itf@cell@calc
2658 \protected\def\itf@cell@numcalc#1#2{%
2659   \itf@cell@calc#2{\number\numexpr#1}%
2660 }% \itf@cell@numcalc
2661 \protected\def\itf@cell@pgfcalc#1#2{%
2662   \pgfmathparse{#1}%
2663   \itf@cell@calc#2\pgfmathresult
2664 }% \itf@cell@pgfcalc
2665 \protected\def\itf@cell@FPCalc#1#2{%
2666   \begingroup
2667   \setbox\z@\vbox{\FPeval introduces some weird code, hence the calculation in a \vbox
2668     \FPeval\FPresult{#1}%
2669     \global\let\FPresult\FPresult}%
2670   \endgroup
2671   \itf@cell@calc#2\FPresult
2672 }% \itf@cell@FPCalc

```

`\itf@cell@insert` Insert a cell (into the temporary `\toks@` token register):

```

2673 \def\itf@cell@getifdefined#1#2{%
2674   \pgfkeysifdefined{/itf/makecell/repeatcell/#1}%
2675     {\pgfkeysgetvalue{/itf/makecell/repeatcell/#1}#2}%
2676     {\let#2\empty}%
2677 }%
2678 \def\itf@cell@insert{%
2679   \pgfkeysvalueof{/itf/makecell/repeatcell/@calc}%
2680   \ifx\R\empty\else
2681     \toks@\expandafter{\the\expandafter\toks@
2682       \expandafter\def\expandafter\r\expandafter{\r}}%
2683   \fi
2684   \ifx\C\empty\else
2685     \toks@\expandafter{\the\expandafter\toks@
2686       \expandafter\def\expandafter\c\expandafter{\c}}%
2687   \fi
2688   \itf@cell@getifdefined{font/row\the\itf@row}\itf@cellfontrow
2689   \itf@cell@getifdefined{font/col\the\itf@col}\itf@cellfontcol
2690   \itf@cell@getifdefined{text/row\the\itf@row}\itf@cellrow
2691   \itf@cell@getifdefined{text/col\the\itf@col}\itf@cellcol
2692   \toks@\expandafter{\the\expandafter\toks@ \itf@cellfont}%
2693   \ifx\itf@cellrow\empty
2694     \ifx\itf@cellcol\empty
2695       \itf@cell\expandafter\expandafter\expandafter{%
2696         \expandafter\itf@cellfontcol \itf@cellfontrow}%
2697       \let\itf@tp\itf@celltext
2698     \else
2699       \itf@cell\expandafter{\itf@cellfontcol}%
2700       \let\itf@tp\itf@cellcol
2701     \fi
2702   \else
2703     \ifx\itf@cellcol\empty
2704       \itf@cell\expandafter{\itf@cellfontrow}%
2705       \let\itf@tp\itf@cellrow
2706     \else
2707       \itf@cell{}%
2708       \ifitf@celltranspose
2709         \edef\itf@tp{\noexpand\itf@diaghead(-2,1)
2710           {\expandonce\itf@cellfontrow\expandonce\itf@cellrow}
2711           {\expandonce\itf@cellfontcol\expandonce\itf@cellcol}}%
2712       \else

```

```

2713     \edef\itf@tp{\noexpand\itf@diaghead(-2,1)
2714         {\expandonce\itf@cellfontcol\expandonce\itf@cellcol}
2715         {\expandonce\itf@cellfontrow\expandonce\itf@cellrow}}%
2716     \fi
2717     \fi
2718     \fi
2719     \itf@cell\expandafter{\the\expandafter\itf@cell \itf@tp}%
2720     \edef\itf@tp{\itf@makecell{\the\itf@cell}}%
2721     \toks@\expandafter{\the\expandafter\toks@ \itf@tp}%
2722 }% \itf@cell@insert

```

`\itf@diaghead` A helper macro to `makecell`'s `\diaghead`

```

2723 \def\itf@diaghead(#1)#2#3{%
2724   \begingroup
2725   \edef\itf@tp{%
2726     \noexpand\diaghead({#1}){\unexpanded{#2 #3}}%
2727         {\unexpanded{#2}}%
2728         {\unexpanded{#3}}}%
2729   \expandafter\endgroup\itf@tp
2730 }% \itf@diaghead

```

21.12.3 `colortbl`: patch for `\CT@cellc`

`\CT@cellc`

```

2731 \AfterPackage*{colortbl}{%
2732 \def\CT@cellc#1[#2]#3{%
2733   #1 is either \@firstofone or \@gobble, [#2]#3 = color spec
2734   \xdef\CT@cell@color{\noexpand\CT@color#1[{{#2}}]}{#3}%
2735   \global\let\noexpand\CT@cell@color\relax}%
2735 }%
2736 </makecell>

```

21.13 ☈ Interface ☈ enumitem : `\setlist`

```

2737 (*enumitem)
2738 \ProvidesPackage{interfaces-enumitem}
2739   [2010/12/27 v2.6 more control on lists (FC)]
2740 \RequirePackage{interfaces-base}
2741 \itf@Check@Package{enumitem}

```

`\interitem` Like `amstext \intertext`, `\interitem` allows to add some non indented text inside a list environment:

```

2742 \providerobustcmd*\interitem{%
2743   \item[]\hspace{-\leftmargin}
2744 }% \interitem

```

`\setlist` (extension, star form)

```

2745 \renewrobustcmd*\setlist{\begingroup \@ifstar
2746   {\let\do\csgappto\itf@setlist}
2747   {\let\do\csgdef\itf@setlist}%
2748 }% \setlist
2749 \newrobustcmd*\itf@setlist[2][0]{%
2750   \let\enit@a@\empty
2751   \let\enit@b@\empty
2752   \@for\x:=#1\do{%
2753     \ifx\x@\empty\else
2754       \itf@number\count@\x\relax\@nnil
2755       {\edef\enit@b{\enit@b,\the\count@}}
2756       {\edef\enit@a{\enit@a,\x}}%
2757     \fi}%
2758   \ifx\enit@a@\empty\def\enit@a{list}\fi
2759   \ifx\enit@b@\empty\def\enit@b{0}\fi
2760   \@for\aa:=\enit@a\do{%
2761     \ifx\aa@\empty\else

```

```

2762     \@for\b:=\enit@b\do{%
2763         \ifx\b\empty\else
2764             \ifcsundef{end\a}{%
2765                 \expandafter\itf@setlist@error\expandafter{\a}%
2766                 {\do{\enit@@\a\romannumeral\b}{,\#2}}%
2767             \fi}%
2768         \fi}
2769     \endgroup
2770 }% \itf@setlist
2771 \def\itf@setlist@error#1{%
2772     \PackageError{interfaces-enumitem}{%
2773         Environment '#1' is undefined
2774         MessageBreak Use \string\newlist\space before \string\setlist
2775         MessageBreak Your command will be ignored}@eha
2776 }% \itf@setlist@error
2777 </enumitem>

```

21.14 Interface `truncate`

```

2778 <*truncate>
2779 \ProvidesPackage{interfaces-truncate}

\trunc{text} \trunc{text} is like \truncate but leaves the text to its natural width if possible.

2780 \newrobustcmd\trunc{text}[3][\TruncateMarker]{% do nothing if width of #3 is less than #2
2781     \begingroup
2782         \hbadness\maxdimen\hfuzz\vsize
2783         \setbox\z@=\hbox to\dimexpr#2\relax\bgrou\color@begingroup
2784             \ignorespaces#3\unskip
2785             \color@endgroup\egroup
2786         \ifnum\badness>@M \truncate[{\#1}]{\#2}{\#3}%
2787     }% \def\next{\truncate[{\#1}]{\#2}{\#3}}%
2788     \else #3\def\next{\#3}%
2789     \fi
2790     \endgroup
2791 }% \trunc{text}

\htrunc{adjust}
2792 \newrobustcmd\htrunc{adjust}[1]{\@testopt
2793     {\itf@htrunc{adjust}{#1}}{}%
2794 }% \htrunc{adjust}
2795 \long\def\itf@htrunc{adjust}{\#2}{% #1=total width, #2=[x_1..x_n], #3=text_1..text_n
2796     \begingroup
2797     \let\itf@trunc@width\empty
2798     \let\itf@istruncated@gobble % to remove the first comma
2799     \let\the@dimen\relax
2800     \count@\z@ \dimen@\z@
2801     \ifblank{\#2}{% no optional [x_1,...,x_n] parameters
2802         %
2803         \@for\x:=\#3\do{%
2804             \advance\count@\@ne
2805             \setbox\z@\hbox\bgrou\color@begingroup
2806                 \ignorespaces\x\unskip
2807                 \color@endgroup\egroup
2808             \dimen\count@ = \wd\z@
2809             \toks\count@\expandafter{\the\dimen\count@}%
2810             \advance\dimen@\dimen\count@}%
2811         }
2812         %
2813         \@for\x:=\#2\do{%
2814             \advance\count@\@ne
2815             \toks\count@\expandafter{\x\p@}%
2816             \advance\dimen@\x\p@
2817             \dimen\count@ -\@ne\p@}%
2818     }%

```

```

2819  \advance\count@\@ne \dimendef\itf@freespace=\count@
2820  \advance\count@\@ne \dimendef\itf@tempdim=\count@
2821  \advance\count@\@ne \dimendef\itf@sum=\count@
2822  \count@\z@
2823  \itf@freespace\z@ % free space left when w_i < x_i/sum(x_i) * total
2824  \@for\x:=#3\do{%
2825      \advance \count@\ @ne
2826      \ifdim\dimen@=\z@ \else
2827          \ifdim \dimen\count@<\z@ % not measured before
2828              \setbox\z@\hbox\bgroup\color@begingroup
2829                  \ignorespaces\x\unskip
2830                  \color@endgroup\egroup
2831          \dimen\count@ = \wd\z@
2832      \fi
2833      \itf@tempdim = \itf@scaledimen{\#1}*\the\toks\count@/\dimen@\relax
2834      \ifdim \dimen\count@ > \itf@tempdim
2835          \dimen\count@ = \itf@tempdim
2836          \edef\itf@istruncated{\itf@istruncated,\the\count@}%
2837          \advance\itf@sum by\the\toks\count@
2838      \else
2839          \advance\itf@freespace \itf@tempdim
2840          \advance\itf@freespace -\dimen\count@
2841      \fi
2842  \fi
2843  \edef\itf@trunc@width{\itf@trunc@width\the\count@{\the@dimen\the\count@}}%
2844 }%
2845 %% Free space=\the\itf@freespace\par
2846 \ifdim\itf@freespace > \z@ % reinsert free space proportionally
2847     \ifx\itf@istruncated\gobble\else
2848         \@for\x:=\itf@istruncated\do{%
2849             \advance\dimen\x by\itf@scaledimen\itf@freespace*\the\toks\x/\itf@sum\relax
2850         }%
2851     \fi\fi
2852     \def\the@dimen{\the\dimen}%
2853     \edef\itf@trunc@width{\itf@trunc@width}%
2854     \expandafter\endgroup
2855     \expandafter\def
2856         \expandafter\itf@trunc@width
2857             \expandafter{\itf@trunc@width}%
2858 %% total=\the\dimexpr#1\relax
2859 }% \itf@htruncadjust
2860 \def\itf@scalefactor#1#2{\strip@pt\dimexpr#1*65536/#2\relax}
2861 \def\itf@scaledimen#1#2#3\relax{%
2862     \dimexpr\itf@scalefactor{#2}{#3}\dimexpr#1\relax\relax
2863 }% \itf@scaledimen

\htruncwidth

2864 \newcommand*\htruncwidth[1]{% purely expandable
2865     \expandafter\itf@htruncwidth\itf@trunc@width\relax\relax\@nnil{#1}%
2866 }% \htruncwidth
2867 \def\itf@htruncwidth#1#2#3\@nnil#4{%
2868     \ifx\relax#1\itf@trunc@nosuchwidth{#4}%
2869     \else\ifnum#1=#4
2870         #2%
2871     \else
2872         \itf@htruncwidth#3\@nnil{#4}%
2873     \fi\fi
2874 }% \itf@htruncwidth
2875 \def\itf@trunc@nosuchwidth#1{%
2876     \PackageError{interfaces-truncate}{%
2877         You requested the width for text nr#1
2878         \MessageBreak but there is no such text
2879         \MessageBreak Therefore no such width!}\@ehd
2880 }% \itf@trunc@nosuchwidth

```

2881 `</truncate>`

21.15 † Interface ↗ appendix and the appendices environment

```

2882 <*appendix>
2883 \ProvidesPackage{interfaces-appendix}
2884   [2010/12/27 v2.6 easy control of appendices environment (FC)]
2885 \RequirePackage{interfaces-base}
2886 \itf@Check@Package{appendix}

\appendices
2887 \let\itf@appendices@pp\appendices
2888 \let\itf@endappendices@pp\endappendices
2889 \let\itf@appendices@pp\appendices
2890 \let\itf@endappendices@pp\endappendices
2891 \renewenvironment{appendices}[1][]{%
2892   \pgfqkeys{/itf/appendix/appendices}{#1}%
2893   \itf@appendices@pp
2894 }
2895 {\itf@endappendices@pp}%

2896 \pgfqkeys{/itf/appendix/appendices}{%
2897   title/.store as code in cs=appendixpagename,%
2898     title/.append code=\ifx\appendixpagename\empty \@dopage@ppfalse
2899       \else \@dopage@pptrue\fi,%
2900     title/.value required,%
2901   intoc/.is if=@dotoc@pp,%
2902   intoc/title/.store as code in cs=appendixtocname,%
2903     intoc/title/.append code=\ifx\appendixtocname\empty \@dotoc@ppfalse
2904       \else \@dotoc@pptrue\fi,%
2905     intoc/title/.default=\pgfkeysvalueof{/itf/appendix/appendices/title},%
2906   section/.is enum={section,chapter},%
2907   section/.append code={\@chapter@ppfalse
2908     \ifnum\pgfkeysvalueof{/itf/appendix/appendices/section/.number}=\z@
2909       \else \ifundefined{chapter}
2910         {\PackageWarning{interfaces-appendix}
2911           {You asked chapter-style appendices but your
2912             \MessageBreak current class does not provide chapters!}}
2913         \@chapter@pptrue
2914       \fi},%
2915   chapter/.style={section=chapter},%
2916   chapter/.value forbidden,%
2917 }% keys for appendices environment

2918 </appendix>

```

21.16 † Interface ↗ embedfile : \embedthisfile

```

2919 <*embedfile>
2920 \ProvidesPackage{interfaces-embedfile}
2921   [2010/12/27 v2.6 easy macros to embed files into pdf (FC)]
2922 \RequirePackage{interfaces-base}
2923 \itf@Check@Package{embedfile}
2924 \newrobustcmd*\embedthisfile{\@ifstar
2925   \embedthisfile@main
2926   \embedthisfile@all
2927 }% \embedthisfile
2928 \@onlypreamble\embedthisfile
2929 \def\embedthisfile@main{\embedfile{\jobname.tex}}
2930 \def\embedthisfile@all{%
2931   \embedthisfile@main
2932   \AfterPreamble{%
2933     \ltx@ifundefined{@listfiles}
2934       {\let{@listfiles}\empty}%
2935       to avoid: \global\let@\filelist\relax in \document
2936     {}%
2937   }%
2938 }

```

```

2936    }%
2937    \gpreto{@enddocumenthook}{%
2938        \let\do\itf@EmbedOne
2939        \expandafter\docsVlist\expandafter{\@filelist}%
2940    }%
2941 }% \embedthisfile@all
2942 \def\itf@EmbedOne#1{\itf@Embed@ne#1.tex\@nil}
2943 \def\itf@Embed@ne#1.tex#2\@nil{%
2944     \ltx@ifundefined{cp@#1}%
2945     {}%
2946     {\ifblank{#2}%
2947         {}%
2948         {\embedfile{#1.tex}}}}%
2949 }% \itf@Embed@ne
2950 </embedfile>

```

21.17 † Interface environ

```

2951 <*environ>
2952 \ProvidesPackage{interfaces-environ}
2953   [2010/12/27 v2.6 ReNewEnviron = RenewEnviron (FC)]
2954 \RequirePackage{interfaces-base}
2955 \AfterPackage{environ}{%
2956   \ifundef\ReNewEnviron
2957     {\let\ReNewEnviron\RenewEnviron}%
2958   }%
2959 }%
2960 </environ>

```

21.18 † Interface umrand

```

2961 <*umrand>
2962 \ProvidesPackage{interfaces-umrand}
2963   [2010/12/27 v2.6 patch for umrand.sty ifnotempty (FC)]
2964 \RequirePackage{interfaces-base}
2965 \BeforePackage{umrand}{%
2966   \let\itf@newif\newif
2967   \def\itf@tp{\ifnotempty}%
2968   \def\newif#1{\def\x{#1}\ifx\x\itf@tp \expandafter\@gobble
2969     \else\expandafter\ltx@newif\fi{#1}}%
2970 }
2971 \AfterPackage{umrand}{%
2972   \let\newif\itf@newif
2973   \let\itf@newif\@undefined
2974 }
2975 </umrand>

```

21.19 † Interface scrlfile

```

2976 <*scrlfile>
2977 \ProvidesPackage{interfaces-scrlfile}
2978   [2010/12/27 v2.6 scrlfile hooks improved (FC)]
2979 \RequirePackage{interfaces-base}

```

\@pkgextension In L^AT_EX they are defined \onlypreamble. So we use the ltxcmds equivalents:

```

\@clsextension
2980 \AfterEndPreamble{%
2981   \let@\pkgextension\ltx@pkgextension
2982   \let@\clsextension\ltx@clsextension
2983   \let@\ifpackageloaded\ltx@ifpackageloaded
2984   \let@\ifclassloaded\ltx@ifclassloaded
2985 }

\scr@load@hook
2986 \renewrobustcmd*\scr@load@hook[2]{%
2987   \csuse{#2-@#1}%

```

```

2988 \ifcsname#2-@#1-\string\keep\endcsname\else\global\csundef{#2-@#1}\fi
2989 }% \scr@load@hook

\scr@set@load@hook (optimisation)

2990 \renewrobustcmd*\scr@set@load@hook[2]{\csgappto{#2-@#1}{}}

\BeforeFile Addition of star forms *: the hooks are not canceled after input (in case a file is input more than once...)
\AfterFile
2991 \renewrobustcmd*\BeforeFile{\@ifstar
2992   {\itf@s@LoadFile{before}}
2993   {\itf@LoadFile{before}}%
2994 }% \BeforeFile
2995 \renewrobustcmd*\AfterFile{\@ifstar
2996   {\itf@s@LoadFile{after}}
2997   {\itf@LoadFile{after}}%
2998 }% \AfterFile

2999 \def\itf@s@LoadFile#1#2{%
3000   \global\cslet{#2-@#1-\string\keep}{\empty}
3001   \scr@set@load@hook{#1}{#2}%
3002 }% \itf@s@LoadFile
3003 \def\itf@LoadFile#1#2{%
3004   \global\csundef{#2-@#1-\string\keep}%
3005   \scr@set@load@hook{#1}{#2}%
3006 }% \itf@LoadFile

\@input (patch) This modification allows \@input to behaves like \InputIfFileExists as far as hooks are concerned.
3007 \renewcommand*\@input[1]{\IfFileExists{#1}%
3008   {\scr@load@hook{before}{#1}%
3009     @@\input\@filef@und
3010     \scr@load@hook{after}{#1}%
3011     {\typeout{No file #1.}}%
3012 }% \@input
3013 </scrfile>

```

21.20 Interface pgfkeys ([pgfmanual](#))

```

3014 {*pgfkeys}
3015 \ProvidesPackage{interfaces-pgfkeys}
3016   [2010/12/27 v2.6 pgfkeys handlers definitions (FC)]
3017 \RequirePackage{interfaces-base}
3018 \ifdefined\pgfkeys@ifexecutehandler % pgf version 2.1
3019 \def\itf@pgfkeys{%
3020   \global\let\pgfkeys@unpack \itf@pgfkeys@unpack
3021   \global\let\pgfkeys@case@three \itf@pgfkeys@case@three
3022   \global\let\pgfkeys@case@three@handleall\itf@pgfkeys@case@three@handleall
3023 }% \itf@pgfkeys
3024 \else
3025 \def\itf@pgfkeys{%
3026   \global\let\pgfkeys@unpack \itf@pgfkeys@unpack
3027   \global\let\pgfkeys@case@three \itf@pgfkeys@case@three
3028   \global\let\pgfkeys@addvalue \itf@pgfkeys@addvalue
3029   \global\let\pgfkeys@exp@call \itf@pgfkeys@exp@call
3030 }% \itf@pgfkeys
3031 \fi
3032 \AtBeginDocument{\itf@pgfkeys \let\itf@pgfkeys \relax}

```

Some new pgfkeys macros

```

\pgfkeysmeaning

3033 \providecommand*\pgfkeysmeaning[1]{\pgfkeysifdefined{#1}%
3034   {\expandafter\meaning\csname pgfk@#1\endcsname}%
3035   {\meaning\@undefined}%
3036 }% \pgfkeysmeaning

```

\pgfkeysvalueof@unexpanded

```
3037 \def\pgfkeysvalueof@unexpanded#1{%
3038   \unexpanded\expandafter\expandafter\expandafter{%
3039     \csname pgfk@#1\endcsname}%
3040 }% \pgfkeysvalueof@unexpanded
```

Some pgfkeys macros redefinition/corrections

\pgfkeysaddvalue (bug correction) Bug in \pgfkeysaddvalue for pgf v2.0, fixed in v2.1

```
3041 \ifdefined\pgfkeys@ifexecutehandler \else % pgf v2.1: bug fixed
3042 \def\itf@pgfkeysaddvalue#1#2#3{%
3043   \edef\pgfkeys@global@temp{%
3044     \unexpanded{#2}%
3045     \ifcsname pgfk@#1\endcsname
3046       \unexpanded\expandafter\expandafter\expandafter{%
3047         \csname pgfk@#1\endcsname}%
3048     \fi
3049     \unexpanded{#3}}%
3050   \pgfkeyslet{#1}\pgfkeys@global@temp
3051 }% \itf@pgfkeysaddvalue
3052 \fi
```

\pgfkeysEsetvalue

\pgfkeysEaddvalue

```
3053 \ifdefined\pgfkeys@ifexecutehandler \long \fi % pgf version 2.1
3054 \def\pgfkeysEsetvalue#1#2{%
3055   \expandafter\edef\csname pgfk@#1\endcsname{#2}%
3056 }% \pgfkeysEsetvalue
3057 \ifdefined\pgfkeys@ifexecutehandler \long \fi % pgf v2.1
3058 \def\pgfkeysEaddvalue#1#2#3{%
3059   \edef\pgfkeys@global@temp{%
3060     #2%
3061     \ifcsname pgfk@#1\endcsname
3062       \unexpanded\expandafter\expandafter\expandafter{%
3063         \csname pgfk@#1\endcsname}%
3064     \fi
3065     #3}}%
3066   \pgfkeyslet{#1}\pgfkeys@global@temp
3067 }% \pgfkeysEaddvalue
```

\pgfkeys@exp@call Missed the braces around the argument (in case the argument contains a comma or an = sign...):

```
3068 %% missed a pair of braces around the argument in pgfkeys.code.tex (pgf v2.0)
3069 %% ( the code could break if it contains = or , )
3070 \ifdefined\pgfkeys@ifexecutehandler\else % (bug corrected in pgf v2.1)
3071   \def\itf@pgfkeys@exp@call#1{\pgfkeysalso{\pgfkeyscurrentpath={#1}}}{%
3072 \fi
```

Handlers `/blank`

```
3073 %% The value is now checked by \itf@pgfkeys@default@blank@required:
3074 \ifdefined\pgfkeys@ifexecutehandler \long \fi % pgf v2.1
3075 \def\itf@pgfkeys@unpack#1=#2#3\pgfkeys@stop{%
3076   \pgfkeys@spdef\pgfkeyscurrentkey{#1}%
3077   \edef\pgfkeyscurrentkey{\pgfkeyscurrentkey}%
3078   \ifx\pgfkeyscurrentkey\pgfkeys@empty
3079     % Skip
3080   \else
3081     \pgfkeys@add@path@as@needed
3082     \pgfkeys@spdef\pgfkeyscurrentvalue{#2}%
3083     \itf@pgfkeys@default@blank@required % <FC>
3084           \pgfkeyscurrentkey % <FC>
3085           \pgfkeys@case@one % <FC>
3086   \fi
```

```

3087 }% \itf@pgfkeys@unpack

\itf@pgfkeys@default@blank@required                                Handle the case: ./@def → ./@blank → ./@def ...

3088 \def\itf@pgfkeys@default@blank@required#1#2{%
3089   \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text% Hmm... no value
3090     \pgfkeysifdefined{#1/.@def}{%
3091       {\pgfkeysgetvalue{#1/.@def}\pgfkeyscurrentvalue}
3092       {}% no default, so leave it
3093     \fi
3094   \ifx\pgfkeyscurrentvalue\empty
3095     \pgfkeysifdefined{#1/.@blank}{%
3096       {\pgfkeysgetvalue{#1/.@blank}\pgfkeyscurrentvalue}
3097       {}% no /.blank specified
3098     \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text% in case /.blank -> \pgfkeysnovalue
3099       \pgfkeysifdefined{#1/.@def}{%
3100         {\pgfkeysgetvalue{#1/.@def}\pgfkeyscurrentvalue}
3101         {}% no default, so leave it
3102       \fi\fi
3103     \pgfkeyscurrentvalueof{/errors/value required/.@cmd}\pgfkeyscurrentkey
3104                                         \pgfkeyscurrentvalue
3105                                         \pgfeov
3106   \else
3107     #2%
3108   \fi
3109 \fi
3110 }% \itf@pgfkeys@default@blank@required

```

Allowing `/default` values (and `/blank` values) for handlers:

```

3111 %% To handle the cases of: /handler/.default= /handler/.blank= /handler/.value required,
3112 %% it is necessary to redefine \pgfkeys@case@three
3113 \def\itf@pgfkeys@case@three{%
3114   \pgfkeys@split@path
3115   \pgfkeysifdefined{/handlers/\pgfkeyscurrentname/.@cmd}{%
3116     {%
3117       \itf@pgfkeys@default@blank@required{/handlers/\pgfkeyscurrentname}%
3118       {\pgfkeysgetvalue{/handlers/\pgfkeyscurrentname/.@cmd}\pgfkeys@code
3119        \expandafter\pgfkeys@code\pgfkeyscurrentvalue\pgfeov}%
3120     }
3121     \pgfkeys@unknown % no handler, thus the key is unknown
3122   }% \itf@pgfkeys@case@three
3123 \ifdefined\pgfkeys@ifexecutehandler
3124   \let\itf@pgfkeys@case@three@handleall \itf@pgfkeys@case@three
3125 \fi

```

Loading interfaces modifications for pgfkeys

```
3126 \itf@pgfkeys
```

Handler definition: `/alias`, `/blank`

```

3127 \pgfkeys{%
3128   /handlers/.alias/.code=\itf@pgfkeys@ifaliasexists{#1}{%
3129     \pgfkeysedef\pgfkeyscurrentpath{%
3130       \noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}},%
3131     /handlers/.alias/.value required,%
3132   /handlers/.blank/.code=\pgfkeyssetvalue{\pgfkeyscurrentpath/.@blank}{#1},%
3133   /handlers/.blank/.default=\pgfkeysnovalue,%
3134 }% /.alias and /.blank
3135 \def\itf@pgfkeys@ifaliasexists#1#2{%
3136   \pgfkeysifdefined{\pgfkeysdefaultpath#1/.@cmd}{#2}
3137   {\pgfkeysifdefined{\pgfkeysdefaultpath#1}

```

```

3138      {#2}
3139      {\itf@pgfkeys@aliaserror{#1}}%
3140 }% \itf@pgfkeys@ifaliasexists
3141 \def\itf@pgfkeys@aliaserror#1{%
3142   \PackageError{interfaces-pgfkeys}{%
3143     {You want the key \pgfkeyscurrentpath\space
3144     \MessageBreak to be an alias for \pgfkeysdefaultpath#1
3145     \MessageBreak but the key \pgfkeysdefaultpath#1
3146     \MessageBreak is not defined!}@\eha@\gobble
3147 }% \itf@pgfkeys@aliaserror

```

Handler definition: `/.alias*` For use with handlers `/.store as glue` or `/.store as glue in cs etc..` when a handler defines key with `*` version and/or `+` version and/or `-` versions.

Then, the `/.alias*` handler is a facility to create aliases for starred `*`, `+`, `-` keys in a batch (as long as the alias exists, of course).

```

3148 \pgfkeys{%
3149   /handlers/.alias*/.code=\itf@pgfkeys@ifaliasexists{#1}{%
3150     \pgfkeysedef\pgfkeyscurrentpath{\noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}}{%
3151       @tfor\x:=+-*\do{%
3152         \pgfkeysifdefined{\pgfkeysdefaultpath#1\x/.@cmd}{%
3153           \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3154             \noexpand\pgfkeysalso{\pgfkeysdefaultpath#1\x={##1}}}}}}}}%
3155     }%
3156   },%
3157   /handlers/.alias*/.value required,%
3158 }% /.alias*

```

Handler definition: `/.is private`

```

3159 \pgfkeys{%
3160   /handlers/.is private/.code=\pgfkeysdef\pgfkeyscurrentpath{%
3161     \PackageError{#1}{Key \pgfkeyscurrentkey\space is private
3162     \MessageBreak and should not be modified}@\ehd},%
3163   /handlers/.is private to class/.code=\pgfkeysdef\pgfkeyscurrentpath{%
3164     \ClassError{#1}{Key \pgfkeyscurrentkey\space is private
3165     \MessageBreak and should not be modified}@\ehd}%
3166 }% /.is private

```

Handlers definition: `/.true /.false /.on /.off`

```

3167 \def\itf@pgfkeys@choice[#1]#2#3{%
3168   \pgfkeys{\pgfkeyscurrentpath/.cd,%
3169     .is choice,%
3170     #2/.code={#3},%
3171     .default/.expanded=@firstoftwo#1}%
3172 }% \itf@pgfkeys@choice
3173 \pgfkeys{%
3174   /handlers/.true/.code=\itf@pgfkeys@choice[{\true}{\false}]{\true}{#1},%
3175   /handlers/.false/.code=\itf@pgfkeys@choice[{\true}{\false}]{\false}{#1},%
3176   /handlers/.on/.code=\itf@pgfkeys@choice[{\on}{\off}]{\on}{#1},%
3177   /handlers/.off/.code=\itf@pgfkeys@choice[{\on}{\off}]{\off}{#1},%
3178 }% /.true /.false

```

Handlers definition: `/.store in cs` `/.estore in cs` etc.

```

3179 \pgfkeys{%
3180   /handlers/.store in cs/.code=\pgfkeysdef\pgfkeyscurrentpath{\csdef{\#1}{##1}},%
3181   /handlers/.estore in cs/.code=\pgfkeysdef\pgfkeyscurrentpath{\csedef{\#1}{##1}},%
3182 }% /.store in cs

```

Handler definition: `/.store as code`

```

3183 %% /.store as code handlers
3184 \pgfkeys{%
3185   /handlers/.store as code/.code={%
3186     \pgfkeyscurrentpath{%
3187       \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
3188       \pgfkeyslet{\pgfkeyscurrentpath}@empty
3189       \pgfkeyslet{\pgfkeyscurrentpath/.@def}@empty
3190       @tfor\x:=#1\do{%
3191         \pgfkeyscurrentpath\x}{%
3192           \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
3193             \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
3194             \noexpand\unexpanded{##1}}}}%
3195         \pgfkeyslet{\pgfkeyscurrentpath\x/.@def}@empty
3196       }},%
3197   /handlers/.store as code/.default=*,%
3198 }% /.store as code

```

Handler definition: `/.estore as code`

```

3199 \pgfkeys{%
3200   /handlers/.estore as code/.code={%
3201     \pgfkeyscurrentpath{%
3202       \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{##1}}%
3203       \pgfkeyslet{\pgfkeyscurrentpath}@empty
3204       @tfor\x:=#1\do{%
3205         \pgfkeyscurrentpath\x}{%
3206           \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
3207             \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
3208             \noexpand\unexpanded{##1}}}}%
3209       }},%
3210   /handlers/.estore as code/.default=*,%
3211 }% /.estore as code

```

Handler definition: `/.store as code in cs=<cs-name>`

```

3212 \pgfkeys{%
3213   /handlers/.store as code in cs/.code={%
3214     \pgfkeyscurrentpath{%
3215       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3216       \else\csdef{\#1}{##1}\fi}%
3217       @tfor\x:=*+\do{%
3218         \pgfkeyscurrentpath\x}{%
3219           \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
3220             \noexpand\ifcsname#1\endcsname
3221               \noexpand\expandafter\noexpand\expandonce
3222                 \noexpand\csname#1\endcsname
3223               \noexpand\fi
3224               \noexpand\unexpanded{##1}}}}%
3225         \pgfkeyslet{\pgfkeyscurrentpath\x/.@def}@empty
3226       }},%
3227   /handlers/.store as code in cs/.value required,%
3228 }% /.store as code in cs
3229 \pgfkeys{%
3230   /handlers/.estore as code in cs/.code={%

```

```

3231     \pgfkeysdef{\pgfkeyscurrentpath}{%
3232         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3233         \else\csedef{#1}{##1}\fi}%
3234     \atfor\x:=*+\do{%
3235         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3236             \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
3237                 \noexpand\ifcsname#1\endcsname
3238                     \noexpand\expandafter\noexpand\expandonce
3239                         \noexpand\csname#1\endcsname
3240                     \noexpand\fi
3241                     \noexpand\unexpanded{##1}}}}%
3242     },% default if \pgfkeysnovalue which expands to \@empty
3243     /handlers/.estore as code in cs/.value required,%
3244 }% /.estore as code in cs

```

Handler definition: `./store as color`

```

3245 \protected\def\itf@pgfk@color#1#2@nil{%
3246     \ifx@let@token\color
3247         \def\pgfkeyscurrentvalue{#2}%
3248     \else\ifx@let@token\normalcolor
3249         \def\pgfkeyscurrentvalue{default}%
3250     \else\ifx@let@token\pgfkeysnovalue
3251         \def\pgfkeyscurrentvalue{default}%
3252     \fi\fi\fi
3253 }% \itf@pgfk@color
3254 \pgfkeys{%
3255     /handlers/.store as color/.code=%
3256         \pgfkeysedef{\pgfkeyscurrentpath}{%
3257             \futurelet\noexpand\@let@token\itf@pgfk@color
3258                 ##1\noexpand\pgfkeysnovalue\noexpand\@nil
3259             \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
3260 }% /.store as color

```

Handler definition: `./store as color in cs=<cs-name>`

```

3261 \pgfkeys{%
3262     /handlers/.store as color in cs/.code=%
3263         \pgfkeysedef{\pgfkeyscurrentpath}{%
3264             \futurelet\noexpand\@let@token\itf@pgfk@color
3265                 ##1\noexpand\pgfkeysnovalue\noexpand\@nil
3266             \cslet{#1}\noexpand\pgfkeyscurrentvalue}%
3267 }% /.store as color in cs

```

Handler definition: `./store as color command`

```

3268 \protected\def\itf@pgfk@colorcmd#1@nil{%
3269     \ifcat @\noexpand\@let@token
3270         \def\pgfkeyscurrentvalue{\color{#1}}%
3271     \else\ifcat @\noexpand\@let@token
3272         \def\pgfkeyscurrentvalue{\color{#1}}%
3273     \else\ifx@let@token@nil
3274         \let\pgfkeyscurrentvalue\normalcolor
3275     \fi\fi\fi
3276 }% \itf@pgfk@colorcmd
3277 \pgfkeys{%
3278     /handlers/.store as color command/.code=%
3279         \pgfkeysedef{\pgfkeyscurrentpath}{%
3280             \futurelet\noexpand\@let@token\itf@pgfk@colorcmd
3281                 ##1\noexpand\@nil
3282             \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
3283             \pgfkeyssetvalue{\pgfkeyscurrentpath/.@def}\normalcolor
3284 }% /.store as color command

```

Handler definition: `./store as color command in cs=<cs-name>`

```

3285 \pgfkeys{%
3286   /handlers/.store as color command in cs/.code=%
3287     \pgfkeysedef\pgfkeyscurrentpath{%
3288       \futurelet\noexpand\@let@token\itf@pgfk@colorcmd
3289       ##1\noexpand\@nil
3290       \cslet{#1}\noexpand\pgfkeyscurrentvalue}%
3291     \pgfkeyssetvalue{\pgfkeyscurrentpath/.@def}\normalcolor
3292 }% /.store as color command in cs

```

Handler definition `./store as number`

```

3293 \pgfkeys{%
3294   /handlers/.store as number/.code=%
3295     \pgfkeysedef\pgfkeyscurrentpath{%
3296       \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\numexpr##1\relax}%
3297     \pgfkeyslet\pgfkeyscurrentpath\z@
3298     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
3299     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3300     @tfor\x:=#1\do{%
3301       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3302         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
3303           \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
3304           \x
3305           \numexpr\unexpanded{##1}\relax}%
3306         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@\fi}%
3307         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@\fi}%
3308       }%,%
3309     /handlers/.store as number/.default=+-*,%
3310   }% /.store as number

```

`./store as number in cs` Handler definition: `./store as number in cs`

```

3311 \pgfkeys{%
3312   /handlers/.store as number in cs/.code=%
3313     \pgfkeysdef\pgfkeyscurrentpath{%
3314       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3315       \else\csdef{#1}{\numexpr##1\relax}\fi}%
3316     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3317     @tfor\x:=+-*\do{%
3318       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3319         \let\cs\noexpand\pgfkeys@temp{%
3320           \noexpand\ifcsname#1\endcsname
3321             #1\noexpand\else\if*\x@\ne\else\z@\fi\noexpand\fi}%
3322         \edef\cs\noexpand\pgfkeys@temp{%
3323           \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3324             \noexpand\expandonce\noexpand\pgfkeys@temp
3325             \x
3326             \numexpr\noexpand\unexpanded{##1}\relax}%
3327           }\noexpand\pgfkeys@temp}%
3328         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@\fi}%
3329         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@\fi}%
3330       }%,%
3331     /handlers/.store as number in cs/.value required,%
3332   }% /.store as number in cs

```

Handler definition: `./estore as number`

```

3333 \pgfkeys{%
3334   /handlers/.estore as number/.code={%
3335     \pgfkeysedef\pgfkeyscurrentpath{%
3336       \noexpand\pgfkeysetvalue{\pgfkeyscurrentpath}{%
3337         \noexpand\number\numexpr##1\relax}%
3338       \pgfkeyslet\pgfkeyscurrentpath\z@%
3339       \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@%
3340       \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@%
3341     @tfor\x:=#1\do{%
3342       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3343         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3344           \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}%
3345           \x
3346           \numexpr##1\relax}%
3347     },%
3348   /handlers/.estore as number/.default=+-*,%
3349 }% /.estore as number

```

Handler definition: `./estore as number in cs`

```

3350 \pgfkeys{%
3351   /handlers/.estore as number in cs/.code={%
3352     \pgfkeysedef\pgfkeyscurrentpath{%
3353       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3354         \else\csedef{\#1}{\number\numexpr##1\relax}\fi}%
3355     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3356     @tfor\x:=-*\do{%
3357       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3358         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3359           \noexpand\csname\noexpand\ifcsname#1\endcsname
3360             #1\noexpand\else\if*\x @ne\else z@\fi\noexpand\fi\endcsname
3361             \x
3362             \numexpr##1\relax}%
3363         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else z@\fi}%
3364         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else z@\fi}%
3365     },%
3366   /handlers/.estore as number in cs/.value required,%
3367 }% /.estore as number in cs

```

Handler definition: `./store in counter cs`

```

3368 \pgfkeys{%
3369   /handlers/.store in counter cs/.code={%
3370     \pgfkeysedef\pgfkeyscurrentpath{%
3371       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3372         \else\csname#1\endcsname\numexpr##1\relax\fi}%
3373     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3374     @tfor\x:=-*\do{%
3375       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3376         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3377           \noexpand\csname\noexpand\ifcsname#1\endcsname
3378             #1\noexpand\else\if*\x @ne\else z@\fi\noexpand\fi\endcsname
3379             \x
3380             \numexpr##1\relax}%
3381         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else z@\fi}%
3382         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else z@\fi}%
3383     },%
3384   /handlers/.store in counter cs/.value required,%
3385 }% /.store in counter cs

```

Handlers definition: `/store as glue`

```

3386 \pgfkeys{%
3387   /handlers/.store as glue/.code={%
3388     \pgfkeysedef{\pgfkeyscurrentpath}{%
3389       \noexpand\pgfkeysetvalue{\pgfkeyscurrentpath}{\glueexpr##1\relax}}%
3390     \pgfkeyslet{\pgfkeyscurrentpath}{z@skip}
3391     \pgfkeyslet{\pgfkeyscurrentpath/.@def}{z@skip}
3392     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}{z@skip}
3393     @tfor\x:=#1\do{%
3394       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3395         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
3396           \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
3397           \x
3398           \if*\x\noexpand\itf@units@scaleNoCalc##1!
3399           \else\glueexpr\noexpand\unexpanded{##1}\relax\fi}}%
3400       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
3401       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
3402     }},%
3403   /handlers/.store as glue/.default=+-*,%
3404 }% /.store as glue

```

Handlers definition: `/store as glue in cs=<cs-name>`

```

3405 \pgfkeys{%
3406   /handlers/.store as glue in cs/.code={%
3407     \pgfkeysedef{\pgfkeyscurrentpath}{%
3408       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3409       \else\csdef{#1}{\glueexpr##1\relax}\fi}%
3410     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}{z@skip}
3411     @tfor\x:=-*\do{%
3412       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3413         \letcs\noexpand\pgfkeys@temp{%
3414           \noexpand\ifcsname#1\endcsname
3415             #1\noexpand\else z@skip\noexpand\fi}%
3416           \edef\noexpand\pgfkeys@temp{%
3417             \noexpand\noexpand\noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3418               \noexpand\expandonce\noexpand\pgfkeys@temp
3419               \x
3420               \if*\x\noexpand\itf@units@scaleNoCalc##1!
3421               \else\glueexpr\noexpand\unexpanded{##1}\relax\fi}}%
3422             }\noexpand\pgfkeys@temp}%
3423       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
3424       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
3425     }},%
3426   /handlers/.store as glue in cs/.value required,%
3427 }% /.store as glue in cs

```

Handlers definition: `/estore as glue`

```

3428 \pgfkeys{%
3429   /handlers/.estore as glue/.code={%
3430     \pgfkeysedef{\pgfkeyscurrentpath}{%
3431       \itf@setlength\itf@sk{##1}%
3432       \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{%
3433         \noexpand\the\itf@sk}}%
3434     \pgfkeyslet{\pgfkeyscurrentpath}{z@skip}
3435     \pgfkeyslet{\pgfkeyscurrentpath/.@def}{z@skip}
3436     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}{z@skip}
3437     @tfor\x:=#1\do{%
3438       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3439         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3440           \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\x
3441           \if*\x\noexpand\itf@units@scale##1!
```

```

3442          \else\glueexpr##1\relax\fi}}%
3443      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
3444      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
3445  },%
3446 /handlers/.estore as glue/.default=+-*,%
3447 }% /.estore as glue

```

Handlers definition: .estore as glue in cs=<cs-name>

```

3448 \pgfkeys{%
3449   /handlers/.estore as glue in cs/.code={%
3450     \pgfkeysdef\pgfkeyscurrentpath{%
3451       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3452       \else\itf@setlength\itf@sk{##1}%
3453       \csedef{#1}{\the\itf@sk}\fi}%
3454     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
3455     @tfor\x:=+-*\do{%
3456       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3457         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3458           \noexpand\csname\noexpand\ifcsname#1\endcsname
3459           #1\noexpand\else z@skip\noexpand\fi\endcsname
3460           \x
3461           \if*\x\noexpand\itf@units@scale##1!
3462           \else\glueexpr##1\relax\fi}}%
3463     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
3464     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
3465   },%
3466   /handlers/.etore as glue in cs/.value required,%
3467 }% /.estore as glue in cs

```

Handlers definition: .store in skip cs This is a variant of .estore as glue in cs when the control sequence in a `\skip` (*i.e.* a length). The skip is given by its name, not as a control sequence.

```

3468 \pgfkeys{%
3469   /handlers/.store in skip cs/.code={%
3470     \pgfkeysdef\pgfkeyscurrentpath{%
3471       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3472       \else\expandafter\itf@setlength\csname#1\endcsname{##1}\fi}%
3473     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
3474     @tfor\x:=+-*\do{%
3475       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3476         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3477           \csname#1\endcsname % should have been defined as a skip (not expandable)
3478           \x
3479           \if*\x\noexpand\itf@units@scale##1!
3480           \else\glueexpr##1\relax\fi}}%
3481     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
3482     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
3483   },%
3484   /handlers/.store in skip cs/.value required,%
3485 }% /.store in skip cs

```

Handlers definition: .store as dimen

```

3486 \pgfkeys{%
3487   /handlers/.store as dimen/.code={%
3488     \pgfkeysedef\pgfkeyscurrentpath{%
3489       \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\dimexpr##1\relax}}%
3490     \pgfkeyslet\pgfkeyscurrentpath\z@
3491     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
3492     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3493     @tfor\x:=#1\do{%

```

```

3494     \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3495         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
3496             \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
3497             \x
3498             \if*\x \noexpand\itf@units@scaleNoCalc##1!
3499             \else\dimexpr\noexpand\unexpanded{##1}\relax\fi}}}}%
3500     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
3501     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
3502     },%
3503     /handlers/.store as dimen/.default=+-*,%
3504 }% /.store as dimen

```

Handlers definition: `./store as dimen in cs=<cs-name>`

```

3505 \pgfkeys{%
3506     /handlers/.store as dimen in cs/.code={%
3507         \pgfkeysedef\pgfkeyscurrentpath{%
3508             \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3509             \else\csdef{#1}{\dimexpr##1\relax}\fi}%
3510             \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3511             \@tfor\x:=+-*\do{%
3512                 \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3513                     \letcs\noexpand\pgfkeys@temp{%
3514                         \noexpand\ifcsname#1\endcsname
3515                         #1\noexpand\else\z@\noexpand\fi}%
3516                         \edef\noexpand\pgfkeys@temp{%
3517                             \noexpand\noexpand\noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3518                                 \noexpand\expandonce\noexpand\pgfkeys@temp
3519                                 \x
3520                                 \if*\x\noexpand\itf@units@scaleNoCalc##1!
3521                                 \else\dimexpr\noexpand\unexpanded{##1}\relax\fi}}%
3522                             }\noexpand\pgfkeys@temp}%
3523                             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
3524                             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
3525             },%
3526             /handlers/.store as dimen in cs/.value required,%
3527 }% /.store as dimen in cs {csname}{+-*}

```

Handlers definition: `./estore as dimen`

```

3528 \pgfkeys{%
3529     /handlers/.estore as dimen/.code={%
3530         \pgfkeysedef\pgfkeyscurrentpath{%
3531             \itf@setlength\itf@sk##1}%
3532             \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{%
3533                 \noexpand\the\dimexpr\itf@sk}%
3534             \pgfkeyslet\pgfkeyscurrentpath\z@
3535             \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
3536             \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3537             \@tfor\x:=#1\do{%
3538                 \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3539                     \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3540                         \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\x
3541                         \if*\x \noexpand\itf@units@scale##1!
3542                         \else \dimexpr##1\relax\fi}}%
3543                     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
3544                     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
3545             },%
3546             /handlers/.estore as dimen/.default=+-*,%
3547 }% /.estore as dimen

```

Handlers definition: `./estore as dimen in cs=<cs-name>`

```

3548 \pgfkeys{%
3549   /handlers/.estore as dimen in cs/.code={%
3550     \pgfkeysdef\pgfkeyscurrentpath{%
3551       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3552       \else \itf@setlength\itf@sk{\#1}%
3553         \csedef{\#1}{\the\dimexpr\itf@sk}\fi}%
3554     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3555     @tfor\x:=+-*\do{%
3556       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3557         \noexpand\ifcsname#1\endcsname
3558           \noexpand@\defaultunits \itf@sk
3559             \noexpand\csname#1\endcsname pt\relax\noexpand@nnil
3560           \noexpand\else \itf@sk \z@ \noexpand\fi
3561           \if*\x
3562             \itf@setlength\itf@sk {\itf@sk*\noexpand\itf@units@scale##1!}%
3563           \else
3564             \noexpand@\defaultunits@tempdima ##1pt \relax\noexpand@nnil
3565             \advance\itf@sk\x \atempdima
3566           \fi
3567           \noexpand\pgfkeysalso{\pgfkeyscurrentpath=\itf@sk}%
3568         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
3569         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
3570     }},%
3571   /handlers/.etore as dimen in cs/.value required,%
3572 }% /.estore as dimen in cs

```

Handlers definition: `./store in dimen cs` This is a variant of `./store as dimen in cs` when the control sequence in a `\dimen`. The `\dimen` is given by its name, not as a control sequence.

```

3573 \pgfkeys{%
3574   /handlers/.store in dimen cs/.code={%
3575     \pgfkeysdef\pgfkeyscurrentpath{%
3576       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3577       \else\expandafter\itf@setlength\csname#1\endcsname{\#1}\fi}%
3578     \pgfkeyslet{/.\pgfkeyscurrentpath/.@blank}\z@
3579     @tfor\x:=+-*\do{%
3580       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3581         \if*\x \else
3582           \noexpand@\defaultunits\itf@sk ##1pt \relax\noexpand@nnil
3583         \fi
3584           \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3585             \csname#1\endcsname % has been defined as a dimen (not expandable)
3586             \x
3587             \if*\x \noexpand\itf@units@scale##1!
3588             \else\itf@sk\fi}%
3589           \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
3590           \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
3591     }},%
3592   /handlers/.store in dimen cs/.value required,%
3593 }% /.store in dimen cs

```

Handlers definition: `./is multiple choice`

```

3594 \pgfkeys{%
3595   /handlers/.is multiple choice/.code=%
3596     \pgfkeys{\pgfkeyscurrentpath/.cd,%
3597       .unknown/.style={%
3598         /errors/unknown choice value=\pgfkeyscurrentkey\pgfkeyscurrentvalue},%
3599         .code=\expandafter\pgfkeys\expandafter{\pgfkeyscurrentkey/.cd,###1}%
3600       },%
3601     /handlers/.is multiple choice/.value forbidden,%
3602 }% /.is multiple choice

```

Handlers definition: `/is enum`

```

3603 \pgfkeys{%
3604   /handlers/.is enum/.code={%
3605     \itf@cnt\z@
3606     \@for\x:=\pgfkeyscurrentvalue\do{%
3607       \pgfkeysedef{\pgfkeyscurrentpath/\x}{%
3608         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\x}%
3609         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath/.number}{\the\itf@cnt}%
3610       \ifnum\itf@cnt=\z@
3611         \pgfkeyslet{\pgfkeyscurrentpath/.def}\x
3612         \pgfkeyslet{\pgfkeyscurrentpath/.blank}\x
3613       \fi
3614       \advance\itf@cnt\@ne
3615     }%
3616     \pgfkeysalso{\pgfkeyscurrentpath/.is choice},%
3617   /handlers/.is enum/.value required,%
3618 }% /.is enum

3619 </pgfkeys>

```

21.21 Interface etoolbox - etoolbox-like commands

```

3620 <*etoolbox>
3621 \ProvidesPackage{interfaces-etoolbox}
3622   [2010/12/27 v2.6 more etoolbox-like commands (FC)]
3623 \RequirePackage{interfaces-base}
3624 \begingroup\catcode`\&=3
3625   \long\gdef\etb@doitem#1&{\do{#1}}
3626 \endgroup

\preappto
\epreappto
\gpreappto 3627 \providerobustcmd\preappto[3]{%
3628   \edef#1{\unexpanded{#2}\ifdefined#1\expandonce#1\fi\unexpanded{#3}}%
3629 }% \preappto
3630 \providerobustcmd\epreappto[3]{%
3631   \edef#1{#2\ifdefined#1\expandonce#1\fi#3}%
3632 }% \epreappto
3633 \providerobustcmd*\gpreappto{\global\preappto}
3634 \providerobustcmd*\xpreappto{\global\epreappto}

\cspreappto
\csepreatto
\csgpreappto 3635 \providerobustcmd*\cspreappto[1]{%
3636   \csname itf@\ifcsname#1\endcsname
3637     preappto\else defpreapp\fi\expandafter\endcsname
3638           \csname#1\endcsname
3639 }% \cspreappto
3640 \long\def\itf@preappto#1#2#3{\edef#1{#2\expandonce#1#3}}
3641 \long\def\itf@defpreapp#1#2#3{\edef#1{#2#3}}
3642 \providerobustcmd*\csepreatto[1]{%
3643   \csname itf@\ifcsname#1\endcsname
3644     preappto\else defpreapp\fi\expandafter\endcsname
3645           \csname#1\endcsname
3646 }% \csepreatto
3647 \providerobustcmd*\csgpreappto{\global\cspreappto}
3648 \providerobustcmd*\csxpreappto{\global\csepreatto}

\removefrom
\eremovefrom
\gremovefrom
\xremovefrom 3649 \providerobustcmd\removefrom[2]{\in@false
3650   \long\def\itf@tp##1\itf@tp{%
3651     \in@true
3652     \edef#1{\unexpanded{##1}}%
3653   }\expandafter\itf@removefrom\expandafter{#1}{#2}%
3654 }% \removefrom

```

```

3655 \providerobustcmd\eremovefrom[2]{\begingroup
3656   \edef\x{\endgroup\removefrom\noexpand#1{#2}%
3657   }\x
3658 }% \eremovefrom
3659 \providerobustcmd\gremovefrom[2]{%
3660   \removefrom{#1}{#2}%
3661   \global\let#1=#1%
3662 }% \gremovefrom
3663 \providerobustcmd\xremovefrom[2]{%
3664   \eremovefrom{#1}{#2}%
3665   \global\let#1=#1%
3666 }% \xremovefrom

\itf@removefrom

3667 \long\def\itf@removefrom#1#2{%
  removes #2 (code) from #1 (hook)
  \def\in@##1#2##2\in@@{\itf@tp##1##2\in@@@}%
  \def\in@@@##1\in@@##2\in@@@{%
    \ifx\in@@##2\else##1\fi
  }\expandafter\in@@@\\in@@#1\itf@tp\in@@@#2\in@\in@@
}%
3672 }% \itf@removefrom

\removefromstart
\eremovefromstart
\gremovefromstart
\xremovefromstart
3673 \newrobustcmd\removefromstart[2]{\in@false
3674   \long\def\itf@tp##1\itf@tp{%
3675     \in@true
3676     \edef#1{\unexpanded{##1}}%
3677   }\expandafter\itf@removefromstart\expandafter{#1}{#2}%
}%
3678 }% \removefromstart
3679 \providerobustcmd\eremovefromstart[2]{\begingroup
3680   \edef\x{\endgroup\removefromstart\noexpand#1{#2}%
3681   }\x
}%
3682 }% \eremovefromstart
3683 \providerobustcmd\gremovefromstart[2]{%
3684   \removefromstart{#1}{#2}%
3685   \global\let#1=#1%
3686 }% \gremovefromstart
3687 \providerobustcmd\xremovefromstart[2]{%
3688   \eremovefromstart{#1}{#2}%
3689   \global\let#1=#1%
}%
3690 }% \xremovefromstart

\itf@removefromstart

3691 \long\def\itf@removefromstart#1#2{%
  removes #2 (code) from #1 (hook)
  \long\def\in@##1#2##2\in@@{##1\in@\itf@tp##2\in@@@}%
  \long\def\in@@@##1\in@@##2\in@@@{%
    \ifx\in@@##2\else##1\fi
  }\expandafter\in@@@\\in@@#1\itf@tp\in@@@#2\in@@
}%
3696 }% \itf@removefromstart

3697 </etoolbox>

```

22 History

[2010/12/27 v2.6]

- `\tocsetup`:
A few new aliases for keys.
Key `columns/sep` didn't work. Fixed.
- Generated file `interfaces-hypbmsec.code.tex` is empty since version [v2.5](#).
- Fixed a bug in `\usetikz{all}`: `pgfmath` should not be loaded twice because of `\pgfmathdeclarefunction...` + `pgfmodulesorting.code.tex` removed: bug in the code...
- `interfaces-TikZ` : keys `shorten>`, `shorten<`, and `shorten<>`.
- Optimisation.

[2010/12/07 v2.5]

- `\sectionformat` now works with the `explicit` option of package `titlesec`. (*in `interfaces-titlesec`*)
- Compatibility with `calc` package as been implemented for `pgfkeys` handlers:

| | | |
|-------------------------------|-------------------------------------|---------------------------------|
| <code>\estore as dimen</code> | <code>\estore as dimen in cs</code> | <code>\store in dimen cs</code> |
| <code>\estore as glue</code> | <code>\estore as glue in cs</code> | <code>\store in skip cs</code> |

As a result, it is now possible to write:

```
\sectionformat\section{labelsep=\widthof{ some text }}
\tocsetup{ section/pagenumbers/width=\widthof{ 00.00\enspace } }
```

For scaling operations, `interfaces` keeps the shrink and stretch components of glues, while `calc` discard them.

- `\listofsetup` now works properly ([interfaces-tocloft](#)).
- Some `+` and `*` keys added to `interfaces-fancyhdr`.
- Fix a bug of `titlesec` when inside `.dtx` files: `\input{ttlkeys.def}` shall be done with `%` as a comment character (`\BeforeFile...\\AfterFile` does the trick).

[2010/11/26 v2.4]

- `interfaces-fancyhdr` : complete key set for header, footer and offset settings
Now keys like `foot/inner` or `head/outer/offset` are available (ouf!).
- `interfaces-hypbmsec` has been completely rewritten and is now independant of `titlesec` and `interfaces-titlesec`. `\lastsectiontitle` etc. are defined by the mean of `\GetTitleString` (from H.O. eponym package).
`interfaces-hypbmsec` now provides `\lastsectionanchor` when used with `hyperref`.
- `interfaces-marks` revisited: expandable macros `\thefirstmarks`, `\thebotmarks` and `\thetopmarks` are now expandable in exactly one step of expansion.
`\marksthe` and `\marksthecs` does not use `\unexpanded` (better for the `.log` file to see the marks).
Optimization for `\getthemarks` with incidence on `\ifmarksequal`.
- `\repeatcell`: modification of the code. & alignment tab character is not activated and the “dirty trick” (according to D. Knuth) `\iffalse{\fi \ifnum0='}\fi` is used instead.
- `\htruncadjust` produced an error when used with empty text arguments – fixed.
- Modification in `\bottomtitles`: `\bottomtitlespace` should be stored as a dimen, not a skip.
- `\removefromstart` is now a `\long` macro.

[2010/11/21 v2.3]

- `interfaces-fancyhdr` : new keys and new aliases for header, footer and offset settings.

[2010/11/16 v2.2]

- Fix a bug in `interfaces-tocloft` (key `section/dotsep`).

[2010/11/15 v2.1]

- Improved default and blank values for bookmark levels, based on `\tt11@<section-name>`
- Correction of a bug in `\setlist`: missed a comma
- Correction of a bug in `interfaces-fancyhdr` (fixed in `./fancy@hf` handler).

[2010/10/31 v2.0]

- Comptability of `interfaces-pgfkeys` with pgf version 2.1
- Modification of the key depth for `\tocsetup` to be able to change the t.o.c. depth anywhere inside the document (just like bookmarks depths).
- handler `./defcounter` has been replaced by more powerful `./store in counter cs`

[2010/10/27 v1.2]

- `interfaces-hypbmsec` needed some corrections.
- The `offset` keys for fancyhdr didn't work. Fixed.
- Correction of converton `pt` to `mu` (catcodes...)
- Added microtype to the documentation...

[2010/10/20 v1.0]

- First version.

Appendix: README (for CTAN)

The `interfaces` package provides a small number of macros to modify the settings of a large number of parameters provided by Most Frequently Used Package (MFUP) for typesetting beautiful document with L^AT_EX.

`interfaces` is organised as different modules which are loaded after the master package is loaded. Each module can be loaded separately.

`\sectionformat`: interface for package `titlesec` to modifies the titles produced by the sectionning commands (and their bookmarks).

`\pagesetup`: interface for package `fancyhdr` to modify the current page style or create new page styles.

`\tocsetup`: interface for package `tocloft` to fine tune the aspect of the table of contents, list of figures / of tables. `\tocsetup` also applies to `\shorttableofcontents` provided by the `shorttoc` package.

`\repeatcell`: to create ‘dynamic’ tabulars, possibly with formula computation (`\numexpr`, `\pgfmathparse` or `\FPeval` formulas can be plugged in the tabular or array).

`\rowfont`: to modify the font and/or the alignment of each cell in a tabular/array row.

+ some macros to make easier the use of ε-T_EX `\marks` registers (useful with `\sectionformat` and `\pagesetup`).

`interfaces` is mainly based on `scrlfile.sty` (provided in the Koma-Script bundle, but Koma-Script classes are not required), on `pgfkeys` and `etoolbox`.

`interfaces` defines some `pgfkeys` handlers designed specially to make key-value interface for other packages. Using those handlers allows to make an interface in a few minutes...

23 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

| Symbols | |
|-------------------------------------|--|
| \# | 81, 89, 98, 743, <u>755</u> |
| \& | 3624 |
| \@input | 3009 |
| \@PackageInfoNoLine | 748,
830, 1007, 2205, 2362, 2369, 2376, 2383, 2390, 2397 |
| \@PackageWarningNoLine | 69, 750,
833, 1009, 2208, 2364, 2371, 2378, 2385, 2392, 2399 |
| \@addtoreset | 587, 595, 633, 635 |
| \@break@tfor | 210 |
| \@cftmakeloftitle_(patch) | 2360 |
| \@cftmakelottitle_(patch) | 2360 |
| \@cftmaketoctitle_(patch) | 2360 |
| \@cftpnumoff | 2198, 2342 |
| \@cftpnumon | 2197, 2341 |
| \@clsextension | 2980 |
| \@defaultunits | 133,
134, 216, 231, 278, 280, 282, 792, 3558, 3564, 3582 |
| \@elt | 1034, 1052 |
| \@expandtwoargs | 210, 225, 226 |
| \@filef@und | 3009 |
| \@filelist | 2934, 2939 |
| \@firstofone | 2469, 2732 |
| \@fourthoffive | 983 |
| \@ifclassloaded | 2984 |
| \@ifpackageloaded | 2983 |
| \@input_(patch) | 3007 |
| \@mainaux | 829, 942, 947 |
| \@onelevel@sanitize | 1091, 2624 |
| \@onlypreamble | 51, 2928 |
| \@pkgextension | 2980 |
| \@tfor | 208, 3151, 3190, 3204, 3217,
3234, 3300, 3317, 3341, 3356, 3374, 3393, 3411,
3437, 3455, 3474, 3493, 3511, 3537, 3555, 3579 |
| \@whilenum | 2479, 2488 |
| \bookmark_(patch) | 828 |
| \newlistof_(patch) | 2360 |
| A | |
| \active | 2441 |
| \addcontentsline_(patch) | 742 |
| \AddLineBeginMainAux | 449 |
| \addtocontents | 2068 |
| \AfterFile | 461, 2189, 2991 |
| \aftergroup | 897, 938, 2431 |
| \AfterPackage | 54,
67, 176, 254, 302, 738, 1089, 1102, 2731, 2955, 2971 |
| \allpgfmodules | <u>315</u> , 404, 432 |
| \alltikzlibraries | 325, 405 |
| \alltikzpackages | 308, 400, 403, 429 |
| \appendices | 2887 |
| \apptocmd | 76, 96, 2388, 2395 |
| B | |
| \BeforeFile | 452, 2185, 2991 |
| \BeforePackage | 37, 2965 |
| \BKM@ref | 985 |
| \BKM@color | 860, 870, 878, 890 |
| \BKM@dest | 981, 983 |
| \BKM@page | 965 |
| C | |
| \bookmark | 6, 828, 831, 834, 952, 969, 2416 |
| \BookmarkAtEnd | 948 |
| \bookmarkget | 906, 908, 916, 918 |
| \bookmarksetup | 6, <u>775</u> , 777, 779, 866, 884, 911, 921 |
| \botmarks | 2014, 2032, 2034 |
| \bottomtitles | 5, <u>785</u> |
| \bottomtitlespace | 795 |
| \BS@BKM@FontStyle | 862, 874, 881, 891, 904, 914 |
| D | |
| \C | 204, 2490, 2497, 2565, 2684 |
| \c | 2490, 2492, 2686 |
| \calc@Adimen | 163 |
| \calc@B | 173, 174 |
| \calc@Bdimen | 163, 173, 174 |
| \calc@multiply@by@real | 162 |
| \cftdot | 2100 |
| \cftdotfill | 2312 |
| \cftdotsep | 2097, 2315 |
| \cftnodots | 2099, 2320, 2321 |
| \changefont | 3, <u>251</u> , 286 |
| \chapter | 1121, 2272, 2909 |
| \color | 240, 241, 289, 293, 1191, 1211, 1352, 1370,
1388, 1406, 1424, 1442, 1460, 1478, 1496, 1514,
1532, 1550, 2091, 2297, 2329, 3246, 3270, 3272 |
| \color@begingroup | 2783, 2805, 2828 |
| \color@endgroup | 2785, 2807, 2830 |
| \colorlet | 237 |
| \column | 2490, 2565, 2596 |
| \columnsep | 2115, 2146 |
| \columnseprule | 2116, 2141 |
| \columnseprulecolor | 2118 |
| \contentsname | 2176, 2221, 2222, 2361, 2431 |
| \count | 133 |
| \countdef | 2471, 2472, 2473 |
| \countvalues | 12, <u>2442</u> |
| \cspreappto | 3058, 3635 |
| \csgappto | 2746, 2990 |
| \csgpreappto | 3635 |
| \cspreappto | 3635 |
| \csxpreappto | 3635 |
| \CT@cell@color | 2733, 2734 |
| \CT@cellc | 2731 |
| \CT@color | 2733 |
| \currentgrouplevel | 864, 896 |
| E | |
| \default@color | 241 |
| \diaghead | 2726 |
| \dimen | 2808, 2809, 2810,
2817, 2827, 2831, 2834, 2835, 2840, 2849, 2852 |
| \dimendef | 2819, 2820, 2821 |
| \dump | 93 |
| F | |
| \embedfile | 2929, 2948 |
| \embedthisfile | 18, 2924, 2927, 2928 |
| \endlinechar | 456, 459 |
| \epreappto | 3627 |
| \eremovefrom | 3649 |

| | |
|---|--|
| \eremovefromstart | 3673 |
| F | |
| \f@baselineskip | 266, 269, 276 |
| \f@linespread | 266, 269, 275, 276 |
| \f@size | 266, 269, 276 |
| \fancy@setoffs | 1725, 1750, 1775, 1800, 1825, 1850, 1875, 1900 |
| \fancyhf | 1133 |
| \fancyhoffset | 1134 |
| \figure | 2268 |
| \firstmarks | 2008, 2024, 2031, 2034 |
| \fontdimen | 192, 198 |
| \fontencoding | 257 |
| \fontfamily | 259 |
| \fontseries | 261 |
| \fontshape | 263 |
| \footrule | 1213, 1214, 1215, 1216 |
| \footrulewidth | 1219, 1220, 1221 |
| \FPeval | 2618, 2647, 2667, 2668 |
| \FPresult | 2668, 2669, 2671 |
| \futurelet | 81, 144, 3257, 3264, 3280, 3288 |
| G | |
| \getrefbykeydefault | 965 |
| \getthebotmarks | 2031 |
| \getthefirstmarks | 2031 |
| \getthemarks | 11, 2024, 2031, 2032, 2033, 2037, 2038 |
| \getthetopmarks | 2031 |
| \GetTitleString | 1032, 1050 |
| \GetTitleStringResult | 1032, 1050 |
| \globaldefs | 1035, 1053 |
| \gluetomu | 189 |
| \gpreappto | 3627 |
| \gremovefrom | 3649 |
| \gremovefromstart | 3673 |
| H | |
| \hbs@checkarg | 1001 |
| \hbs@do_(patch) | 995 |
| \hbs@process_(patch) | 1004 |
| \headrule | 1193, 1194, 1195, 1196 |
| \headrulewidth | 1199, 1200, 1201 |
| \headwidth | 1200, 1221, 1974, 1975, 1976, 1979, 1980, 1983 |
| \htruncadjust | 17, 2792 |
| \htruncwidth | 2864 |
| \hyper@anchor | 1023 |
| \HyperLocalCurrentHref | 1061 |
| \HyperRaiseLinkHook | 806, 807, 808, 809 |
| \HyperRaiseLinkLength | 810 |
| \hypersetup | 2298, 2330 |
| I | |
| \ifcsundef | 490, 493, 534, 593, 600, 628, 1158, 1162, 1996, 2764 |
| \ifcsvoid | 686, 687, 688 |
| \ifitf@celltranspose | 2465, 2501, 2708 |
| \ifmarksequal | 12, 2034 |
| \ifnotempty | 2967 |
| \ifttl@explicit | 494, 543 |
| \ifundef | 1121, 2909, 2956 |
| \ignorespaces | 2542, 2784, 2806, 2829 |
| \in@ | 210, 225, 226, 429, 432, 1092, 3670, 3671, 3692, 3694 |
| \interitem | 16, 2742 |
| \itf@AfterPackage | 64 |
| \itf@BKM@enable | 838, 840, 841, 842, 843, 844, 845, 939, 940, 944, 946 |
| \itf@BKM@pageref | 952 |
| \itf@BKM@ref | 969 |
| \itf@BKM@restore | 897, 925, 929, 938 |
| \itf@BKM@write | 829, 937 |
| \itf@calc@Askip | 163, 177 |
| \itf@calc@Bskip | 163, 177 |
| \itf@calc@multiply@by@real | 162, 172, 175 |
| \itf@cell@calc | 2653 |
| \itf@cell@formulaerror | 2580, 2581, 2583, 2606, 2615, 2618, 2623, 2652 |
| \itf@cell@FPcalc | 2619, 2653 |
| \itf@cell@getformula | 2585, 2603 |
| \itf@cell@init | 2480, 2483, 2489, 2490, 2494, 2496, 2500, 2517, 2536 |
| \itf@cell@insert | 2518, 2537, 2673 |
| \itf@cell@numcalc | 2596, 2597, 2612, 2653 |
| \itf@cell@pgfcalc | 2616, 2653 |
| \itf@celltransposefalse | 2590 |
| \itf@cft@label | 2402 |
| \itf@cft@new@section | 2285, 2286, 2345 |
| \itf@cft@newlistof | 2063, 2214, 2215, 2216, 2389 |
| \itf@cft@newsection | 617, 2285, 2348, 2351, 2353, 2356, 2358, 2396 |
| \itf@cft@secname | 2257, 2258, 2260, 2341, 2342 |
| \itf@cft@sectionname | 2257 |
| \itf@countvalues | 2443, 2445, 2449, 2451 |
| \itf@diaghead | 2709, 2713, 2723 |
| \itf@expandonceifcs | 123, 2573, 2601 |
| \itf@getv@lu@ | 2454, 2455, 2462, 2464 |
| \itf@getvalue | 2452, 2485, 2498 |
| \itf@hbs@cl@ORI | 1072, 1074, 1075 |
| \itf@hbs@currentname | 996, 1016, 1024, 1026, 1040, 1047, 1062, 1069 |
| \itf@hbs@currenttitle | 1032, 1036, 1050, 1054 |
| \itf@hbs@do | 995, 1002, 1003 |
| \itf@hbs@last | 1017, 1029, 1045, 1067, 1104, 1105 |
| \itf@hbs@last@nostar | 1049 |
| \itf@hbs@last@nostar@nottl | 1026, 1049 |
| \itf@hbs@last@nostar@nottl@Hy | 1024, 1049 |
| \itf@hbs@last@nostar@ttl | 1029, 1049 |
| \itf@hbs@last@starred | 1016, 1031, 1065 |
| \itf@hbs@last@starred@ttl | 1017, 1044, 1048 |
| \itf@hbs@process | 1005, 1020 |
| \itf@hbs@ssect | 1000, 1014 |
| \itf@htruncadjust | 2793, 2795, 2859 |
| \itf@htruncwidth | 2865, 2867, 2872, 2874 |
| \itf@makecell | 2469, 2470, 2720 |
| \itf@marks@new | 1991, 1995, 2002 |
| \itf@message | 2540, 2544 |
| \itf@musk | 115, 2096, 2097, 2317, 2318 |
| \itf@Not@titlesec | 1015, 1021, 1088 |
| \itf@number | 132, 2551, 2570, 2598, 2754 |
| \itf@pagesetup | 1138, 1139, 1141, 1143 |
| \itf@patch | 81, 745, 828, 1004, 2360, 2367, 2374, 2381, 2388, 2395 |
| \itf@patch@inside@ | 101, 107, 113 |
| \itf@pgfk@color | 3245, 3253, 3257, 3264 |
| \itf@pgfk@colorcmd | 3268, 3276, 3280, 3288 |
| \itf@pgfkeys@aliaserror | 3139, 3141, 3147 |
| \itf@pgfkeys@case@three | 3021, 3027, 3113, 3122, 3124 |
| \itf@pgfkeys@case@three@handleall | 3022, 3124 |

| | | | |
|---|--|---|--|
| \itf@pgfkeys@default@blank@required | 3073, 3083, 3088, 3117 | \LoadInterface | 6, 7, 9, 39 |
| \itf@pgfkeys@ifaliasexists | 3128, 3135, 3140, 3149 | \lofsetup | 2051, 2373 |
| \itf@preappto | 3640 | \loop | 108 |
| \itf@preapptocmd | 74, 84, 745 | \lotsetup | 2051, 2380 |
| \itf@rem@mu | 182, 186 | \ltx@clsextension | 2982 |
| \itf@removefrom | 3653, 3667 | \ltx@ifclassloaded | 2984 |
| \itf@removefromstart | 3677, 3691 | \ltx@ifpackageloaded | 52, 56, 65, 68, 235, 239, 2107, 2983 |
| \itf@repeatcell | 2466, 2467, 2543 | \ltx@pkgextension | 2981 |
| \itf@resetlastsections | 1034, 1052, 1107 | | |
| \itf@scaledimen | 2833, 2849, 2861, 2863 | M | |
| \itf@scalefactor | 2860, 2862 | \makecell | 2470 |
| \itf@sectionformat | 468, 469, 471 | \marks | 42, 1992, 2000 |
| \itf@setbottomtitlespace | 788, 791, 794 | \marksthe | 11, 680, 1990 |
| \itf@setlength | 150, 178, 3431, 3452, 3472, 3531, 3552, 3562, 3577 | \marksthecs | 1994 |
| \itf@setlength@calc | 161 | \maxdimen | 2782 |
| \itf@setlist | 2746, 2747, 2749, 2770 | \muexpr | 196 |
| \itf@setlist@error | 2765, 2771, 2776 | \multicolssep | 2114 |
| \itf@sk | 114, 278, 279, 280, 281, 282, 283, 789, 792, 795, 3431, 3433, 3452, 3453, 3531, 3533, 3552, 3553, 3558, 3560, 3562, 3565, 3567, 3582, 3588 | \muskip | 188, 201, 220 |
| \itf@strip@at | 118, 521, 996 | \mutoglu | 196 |
| \itf@strip@dot | 118, 191, 197 | | |
| \itf@strip@mu | 181, 2097, 2318 | N | |
| \itf@tk | 116, 1972 | \newcounter | 594 |
| \itf@tocloft | 2204, 2361, 2368, 2375, 2402, 2406 | \newlist | 2774 |
| \itf@ttl@BKM | 746, 756, 784 | \newlistof | 2063, 2381, 2384, 2386, 2388, 2391, 2393 |
| \itf@ttl@createsection | 606, 629 | \newmarks | 2003 |
| \itf@ttl@getformat | 481 | \newmuskip | 115 |
| \itf@ttl@getspacing | 481 | \newskip | 114, 177 |
| \itf@ttl@newsection | 489, 625 | \newtoks | 116 |
| \itf@ttl@sectionformat | 480 | \normalbaselineskip | 1242 |
| \itf@ttl@setkey | 502, 506, 508, 509, 510, 511, 512, 523, 524, 525, 526, 527 | \normalcolor | 2151, 3248, 3274, 3283, 3291 |
| \itf@ttl@setup | 481 | | |
| \itf@ttl@titleclass | 573, 633, 635 | P | |
| \itf@ttl@titleclass@i | 576, 577, 579, 591 | \pagesetup | 10, 1135, 1179, 1972 |
| \itf@ttl@titleclass@ii | 587, 592, 605 | \pagestyle | 1150, 1158, 1181, 1981 |
| \itf@ttl@update | 484, 529 | \paragraph | 2266 |
| \itf@undefinedpagestyle | 1159, 1163, 1166, 1169 | \patchcmd | 83, 101, 106, 828, 2381 |
| \itf@units@mu | 201, 2096, 2317 | \pdfstringdefDisableCommands | 255 |
| \itf@units@mutopt | 188, 229 | \pgfkeys@case@three | 3021, 3027, 3112 |
| \itf@units@pt | 220 | \pgfkeys@case@three@handleall | 3022 |
| \itf@units@pttому | 188, 214 | \pgfkeys@exp@call | 3029, 3068 |
| \itf@units@scale | 151, 3441, 3461, 3479, 3541, 3562, 3587 | \pgfkeys@global@temp | 3043, 3050, 3059, 3066 |
| \itf@units@scale@normalize | 153, 154, 158, 160, 179 | \pgfkeys@ifexecutehandler | 3018, 3041, 3053, 3057, 3070, 3074, 3123 |
| \itf@units@scale@normalize@calc | 161 | \pgfkeys@unpack | 3020, 3026 |
| \itf@units@scale@normalizeNoCalc | 159, 160 | \pgfkeysaddvalue_(bug_correction) | 3041 |
| \itf@units@scaleNoCalc | 159, 3398, 3420, 3498, 3520 | \pgfkeysEaddvalue | 2171, 2177, 2586, 3053 |
| \itf@usetikzcode | 407, 427 | \pgfkeysEsetvalue | 3053 |
| \itf@usetikzsetup | 415, 417 | \pgfkeysmeaning | 3033 |
| | | \pgfkeysnovalue | 2081, 3098, 3133, 3242, 3250, 3258, 3265 |
| L | | \pgfkeysvalueof@unexpanded | 3037, 3193, 3207, 3303, 3396, 3496 |
| \label | 2427 | \pgfkeysvaluerequired | 3103 |
| \lastchapteranchor | 1125 | \pgfmathdeclarefunction | 313 |
| \lastchapterbookmark | 1124 | \pgfmathparse | 2615, 2643, 2662 |
| \lastchaptertitle | 1122 | \pgfmathresult | 2663 |
| \lastchaptertos | 1123 | \preappto | 3627 |
| \lastsection | 1012 | \pretocmd | 75, 96, 1004, 2203, 2360, 2367, 2374 |
| \lastsectionname | 1040, 1062, 1114 | | |
| \listfigurename | 2233, 2234, 2368 | R | |
| \listofsetup | 9, 2051, 2211, 2387, 2394, 2401 | \R | 2480, 2484, 2680 |
| \listtablename | 2239, 2240, 2375 | \r | 2480, 2482, 2682 |
| | | \raisedhyperdef | 6, 802, 2412 |
| | | \ratio | 168, 169 |

| | | | | | |
|---|-----------------------------|------------------------------|------------------------------|--|--|
| \rem@pt | 275 | \thispagestyle | 1162 | | |
| \remove@to@nnil | 128, 2447, 2457, 2460, 2621 | \tikzset | 304 | | |
| \removefrom | 23, 3649 | \titleclass | 448, 581, 584, 609 | | |
| \removefromstart | 2546, 2548, 3673 | \titleformat | 448 | | |
| \ReNewEnviron | 2956, 2957 | \titlespacing | 448 | | |
| \RenewEnviron | 2957 | \tocloftpagestyle | 2071 | | |
| \repeat | 112 | \tocsetup | 8, 8, 2051, 2366 | | |
| \repeatcell | 12, 2465, 2602, 2634, 2637 | \toks | 2809, 2815, 2833, 2837, 2849 | | |
| \row | 2480, 2597 | \toksdef | 2474 | | |
| S | | | | | |
| \scr@load@hook | 2986, 3008, 3010 | \topmarks | 2020, 2033 | | |
| \scr@set@load@hook_(optimisation) | 2990 | \truncate | 2786, 2787 | | |
| \sectionformat | 4, 467, 777 | \TruncateMarker | 2780 | | |
| \selectfont | 252 | \trunctext | 17, 2780 | | |
| \set@fontsize | 266, 269, 276 | \ttl@finmarks | 1017, 1029, 1044, 1066, 1102 | | |
| \setlength | 164 | \ttl@getkeys | 531, 532 | | |
| \setlist | 16 | \ttl@spacing@ii | 566 | | |
| \setlist_(extension,_star_form) | 2745 | \ttlh@runinifempty | 462 | | |
| \skip | 188, 201, 220 | \ttlp@append | 444, 450 | | |
| \strip@pt | 2860 | U | | | |
| \strut | 1177 | \unless | 111 | | |
| \subparagraph | 2267 | \unskip | 2784, 2806, 2829 | | |
| \subsubfigure | 2269 | \usepgfmodule | 404, 433 | | |
| \subsubsection | 2265 | \usetikz | 3, 408, 409, 2643 | | |
| \subtable | 2271 | \usetikzlibrary | 393, 405, 434 | | |
| T | | | | | |
| \table | 2270 | \vsize | 2782 | | |
| \textfont | 192, 198 | \vskip | 1201, 1220, 1221 | | |
| \textheight | 792 | W | | | |
| \the@dimen | 2799, 2843, 2852 | \write | 829, 942, 947 | | |
| \thebotmarks | 2006 | X | | | |
| \thefirstmarks | 11, 2006 | \xpreappto | 3627 | | |
| \thepage | 961, 978 | \xremovefrom | 3649 | | |
| \thetopmarks | 2006 | \xremovefromstart | 3673 | | |