The interfaces* package

interfaces with keys for other packages (FC)

<florent.chervet at free.fr>

2010/11/15 – version 2.1 – 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 base on pgfkeys, hence the "key=value" syntax.

interfaces is mainly based on pgfkeys[1] and scrlfile 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} ***after*** \usepackage{titlesec}. A better way is to add \usepackage{interface-base} 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-scrlfile, interfaces-etoolbox[2] and interfaces-pgfkeys: those are the *system-layer* of interfaces.

Example

```
\pagesetup[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
 }
\pagesetup[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}{lastsectiontitle},% interfaces-hypbmsec
   mark=\markstheс{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\MiniSection{% new sectionning command
   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 documentation, run (thrice): `pdflatex interfaces.dtx`
To get the index, run: `makeindex -s gind.ist interfaces.idx`
⟶ To get the package, run: `etex interfaces.dtx`

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 LaTeX . . . 3
2 Interface TikZ . . . 3
3 Interface titlesec: \sectionformat . . . 4
4 Interface hyperref . . . 6
5 Interface bookmark . . . 6
6 Interface hypbmsec . . . 7
7 Interface tocloft: \tocsetup . . . 8
8 Interface fancyhdr: \pagesetup . . . 9
9 Interface marks . . . 10
10 Interface makecell: \repeatcell . . . 11
10.1 Conversion to scaled points . . . 12
10.2 Variable length cell . . . 12
10.3 Empty rows . . . 12
10.4 color palette . . . 13
10.5 tabularcalc-like tables (with dcolumn package) and pgfmath . . . 13
10.6 Double entry array: the Body Mass Index . . . 14
11 Interface enumitem . . . 15
12 Interface truncate . . . 16
13 Interface appendix . . . 17
14 Interface embedfile: \embedthisfile . . . 17
15 Interface environ . . . 17
16 Interface umrand . . . 17
17 Interface pgfkeys . . . 18
17.1 /.blank . . . 18
17.2 /.is private . . . 19
17.3 /.store in cs, /.estore in cs . . . 19
17.4 Boolean keys: /.true, /.false, /.on /.off . . . 19
17.5 /.is multiple choice . . . 19
17.6 /.is enum . . . 20
17.7 /.store as code . . . 20
17.8 /.store as code in cs . . . 20
17.9 /.store as color . . . 20
17.10 /.store as color in cs . . . 21
17.11 /.store as number . . . 21
17.12 /.store as number in cs . . . 21
17.13 /.store as glue . . . 21
17.14 /.store as dimen . . . 21
17.15 /.store as glue in cs . . . 21
17.16 /.store as dimen in cs . . . 22
17.17 /.estore as dimen in cs . . . 22
18 Interface etoolbox . . . 22
18.1 New macros provided with \providecommand . . . 22
19 Interface scrlfile . . . 22
19.1 Star forms for \BeforeFile and \AfterFile . . . 22
19.2 Allowing \BeforeFile and \AfterFile when \@input is invoked . . . 23
20 List of corrected or patched commands . . . 23
20.1 LaTeX . . . 23
20.2 bookmark . . . 23
20.3 colortbl . . . 24
20.4 hypbmsec . . . 24
20.5 enumitem . . . 24
20.6 pgfkeys . . . 24
21 Implementation . . . 25
21.1 Interfaces . . . 25
21.1.1 Requirements
21.1.2 The package options / list of interfaces
21.2 Interfaces-base . . . 25
21.2.1 Bug Fixes
21.2.2 Interfaces patches
21.2.3 Helper macros
21.2.4 Numbers
21.2.5 Units and math units
21.2.6 Default color
21.3 Interface LaTeX . . . 30
21.3.1 \changefont
21.4 Interface TikZ . . . 31
21.4.1 Catcode verifications
21.4.2 List of pgf/TikZ packages and libraries
21.4.3 pgfkeys definitions for \usetikz
21.5 Interface titlesec: \sectionformat . . . 34
21.5.1 titlesec modifications
21.5.2 \sectionformat
21.5.3 Keys for \sectionformat
21.5.4 \bottomtitles
21.6 Interface hyperref . . . 40
21.7 Interface bookmark . . . 41
21.7.1 star keys* definitions
21.7.2 ref and pageref keys
21.8 Interface hypbmsec . . . 44
21.8.1 interfaces-hypbmsec
21.8.2 interfaces-hypbmsec.code.tex
21.9 Interface fancyhdr: \pagesetup . . . 46
21.9.1 \pagesetup
21.9.2 Keys for \pagesetup
21.10 marks . . . 52
21.11 Interface tocloft: \tocsetup . . . 54
21.11.1 Specific handlers for tocloft
21.12 Interface makecell: \repeatcell . . . 60
21.12.1 countvalues
21.12.2 \repeatcell
21.12.3 colortbl: patch for \CT@cellc
21.13 Interface enumitem . . . 66
21.14 Interface truncate . . . 67
21.15 interface fancyvrb . . . 69
21.16 Interface appendix . . . 69
21.17 Interface embedfile: \embedthisfile . . . 70
21.18 Interface environ . . . 71
21.19 Interface umrand . . . 71
21.20 Interface scrlfile . . . 71
21.21 Interface pgfkeys . . . 72
21.21.1 interfaces-pgfkeys
21.22 Interface etoolbox . . . 85
22 History . . . 86
[2010/11/15 v2.1] . . . 86
[2010/10/31 v2.0] . . . 87
[2010/10/27 v1.2] . . . 87
[2010/10/20 v1.0] . . . 87
23 References . . . 87
README . . . 87
24 Index . . . 88

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 \usetizlibrary }

\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.

\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 TEX 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 substract and the * key scales.

MEMO

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

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

bookmark/bold	bold bookmark text	the text of the bookmark for that sectionning command will be **bold**
bookmark/italic	italic bookmark text	the text of the bookmark for that sectionning command will be in *italic*
bookmark/color	colored bookmark text	the text of the bookmark for that sectionning command will be colorized
bookmark/off	disable the bookmark	no bookmark for that sectionning command alternatively you may set bookmark/level to a huge value...
bookmark/style	use a *style*	the *style* must be previouly declared with \bookmarkdefinestyle
bookmark/level	level of the bookmark	defines \toclevel@⟨*sectionname*⟩, see bookmark 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* and color* 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 acertain 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

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

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

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

One swallow does not a summer make, interface-titlesec does not provides 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 (*ie.* 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 sets the value of \bottomtitlespace (see titlesec documentation):

```
\bottomtitles{.2}              set \bottomtitlespace to .2\textheight
\bottomtitles=.2\textheight    is the same
```

You can 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 requires interfaces-titlesec, and therefore the titlesec package: when using interfaces with automatic module loading, (or while invoking `\LoadInterface{hypbmsec}`), this module is loaded only after both titlesec.sty and hypbmsec.sty.

Package hypbmsec provides extended sectionning commands with the syntax:

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

Where:
- `(out)` is used for the bookmark (outline)
- `[toc]` for tha 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:

`\lastsectiontitle`	expands to the title of the last `\section`
`\lastsectionbookmark`	expands to the text of the bookmark for the last `\section`
`\lastsectiontoc`	expands to the entry of the last `\section` in the table of contents

Macros for each sectionning command are defined:

`\lastparttitle` `\lastchaptertitle`
`\lastsectiontitle` `\lastsubsectiontitle` `\lastsubsubsectiontitle`
`\lastparagraphtitle` `\lastsubparagraphtitle`

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

Assignation 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, *ie.* if `\sectionformat\chapter` or `\titleformat\chapter` has been invoked. Otherwise, `\lastchaptertitle`, `\lastchapterbookmark` and `\lastchaptertoc` 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 interface-marks (page 10 under ⇝`\ifmarksequal`). With this feature, when a new `\section` is created, `\lastsubsection` is reset to **`\relax`** therefore, `\marksthecs{⟨subsection⟩}{⟨lastsubsection⟩}` will not mark anything.

7 Interface ☞ tocloft : \tocsetup, \lofsetup and \lotsetup

Setting the keys overwrites the previous value, unless the + key (append code/add for dimensions), the - key (substract) 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 ... However 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|substract|scale)
- title/bottom |title/bottom+ *etc.* = vertical space after the title
- title/font|title/font* = the title font (set|append code)
- title/after|title/after* = hook to execute just after the title (\cftaftertoctitle...)
- parskip|parskip+ *etc.*|parsep = modifies (or add to) \cftparskip
- dotsep = modifies \cftdotsep if no dimen is specified, *mu* is assumed, otherwise, conversion to *mu* ≃ 1*em*/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|twocolums/before = hooks just after \begin{multicols} ...
- multicols/end|twocolums/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
- ¶/number/width|number/width+|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*|aftersnumb = code before the title (*ie.* after the section number box)
- ¶/leader|¶/leaders = modifies the leader: the default is \cftdotfill{\cft¶dotsep}
- ¶/dotsep = the space (in *mu*) between dots in the leader. Default is \cftdotsep and blank value is the same as \cftnodots
- ¶/pagefont|¶/page/font|¶/pagenumber/font = modifies the font used to typeset the page number
- ¶/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, but there is no sub-keys for section, subsection *etc.*.

8 Interface ☞ fancyhdr : \pagesetup

\pagesetup[⟨*pagestyle-name*⟩]{

norule			set \headrulewidth and \footrulewidth to 0pt
rule|rule/width[* + -] =...,			set \headrulewidth and \footrulewidth to a given value
rule/color =...,	rule/font =...,		To change the rule font/color
headrule =...,	footrule =...,		redefines \headrule and \footrule (no value restores default)
head/rule/width =...,	foot/rule/width =...,	foot/rule/skip =...,	dimensions
head/rule/color =...,	foot/rule/color =...,		for head and foot rules separately
head/rule/font =...,	foot/rule/font =...,		*idem*
head =...,	foot =...,		header and footer (\fancyhead/\fancyfoot)
left =...,	center =...,	right =...,	by position
head/left =...,	head/center =...,	head/right =...,	for one side documents
foot/left =...,	foot/center =...,	foot/right =...,	*idem*
head/left/even =...,	head/left/odd =...,		for two side documents
etc.	*etc.*	*etc.*	
foot/right/even =...,	foot/right/odd =...,		
offset =...,			changes \fancyhfoffset
head/offset =...,	foot/offset =...,		
left/offset =...,	right/offset =...,		
head/left/even/offset =...,	head/left/odd/offset =...,		for two side documents
etc.	*etc.*	*etc.*	
foot/right/even/offset =...,	foot/right/odd/offset =...,		for two side documents
pagestyle =...,			load the settings of another page style

}

\pagesetup* [⟨*pagestyle-name*⟩]{key = value}

\pagesetup resets all headers and footers, rules etc. and modifies the current (active) page style depending on the key=value pairs. \pagesetup[⟨*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).

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

\pagesetup* [⟨*page-style*⟩] defines or redefines ⟨*page-style*⟩,; the normal form starts from scratch, while the starred form \pagesetup* 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, substraction and scaling . And for code (font keys) the + and * keys are provided: both append their content to the previous value.

9 Interface marks

interface-marks provides a few macros to make easier the use of ε-TEX \marks registers. ε-TEX defines 32 768 marks registers, making the use of them far more comfortable than "old" LATEX \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*⟩}
\markstheccs{⟨*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).

\markstheccs{⟨*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 \markstheccs suitable to use with \lastsectiontitle *etc.*. at \sectionbreak (see interfaces-hypbmsec p. 7).

\thefirstmarks{⟨*named-mark*⟩}
\thebotmarks{⟨*named-mark*⟩}
\thetopmarks{⟨*named-mark*⟩}

\thefirstmarks{⟨*chapter*⟩} expands to \firstmarks\marks@⟨*chapter*⟩ (this is expandable), *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}\hskip.4em\textendash\hskip.4em#1}}
\pagesetup{head/right=\thefirstmarks{section}}
```

\getthemarks\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, *ie.* 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=\markstheccs{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,
- $\pgfmath$ (⇝\usetikz{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).

\repeatcell[⟨*alignment*⟩]{⟨*columns*⟩}{← number of columns *or* comma separated list of values

text = the text to be inserted (repeated) in each cell
font = the font inside the cell
rows = the number of rows (default is 1 row) *or* a comma separated list of value: {one,two,three,four,etc.} Empty values ***are not*** ignored.
cr|end = the end-of-row sequence (default is \\ but you may want \crcr\hline or \tabularnewline...) **end of row sequence *is never* inserted after the last row.**
text/row1|text/row2|text/row3 *etc.* = If you want to change the text on the specified rows
text/col1|text/col2|text/col3 *etc.* = If you want to change the text on the specified columns
font/row1|font/row2|font/row3 *etc.* = If you want to change the font on the specified rows
font/col1|font/col2|font/col3 *etc.* = If you want to change the font on the specified columns
cr/row1|cr/row2 |cr/row3 *etc.* = If you want to change the end-of-line sequence on the specified rows
transpose = Do you wish to invert rows and columns ? Then transpose=true ☺.
any other key = can be used to make calculations using \numexpr or $\pgfmathparse$ or $$\FPeval$$. Pay attention to put your formula inside curly brackets {$formula$} if it contains a comma , or an equal = 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, ..., *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{⟨*n*⟩}	to the value of the list for row {⟨*n*⟩}

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{⟨m⟩}	to the value of the list for column {⟨m⟩}

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...*

Important note: The mechanism of \halign does not allow to insert tab alignment characters into the arguments of \repeatcell. Therefore, to be able to define a text key containing a tabular or array, the **&** alignment tab character is activated (catcode 13) at the beginning of \repeatcell. This activation has no effect if \repeatcell is in the replacement text of one of your macro, because the category code of **&** is fixed at the time your macro was read. Thus, use the control sequence \tab instead of **&** in this case. \tab is only defined during the time \repeatcell does its job.

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{tabular}{|>{\bfseries}c|*{9}{|r}|}                                    \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{tabular}
```

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{tabular}{|*{3}{>{\centering\arraybackslash}m{.12\linewidth}}|}     \hline
Mother & Father & Children                    \\ \hline
 #1    &  #2    & \makecell{\repeatcell{#3}{text=\c, transpose} \\ \hline
\end{tabular}}
\FAMILY{Johanne}{Georg}{Priscilla,John,Tom}
\FAMILY{Nut}{Geb}{Isis,Osiris,Seth,Nephthys}
```

Mother	Father	Children
Johanne	Georg	Priscilla John Tom

Mother	Father	Children
Nout	Geb	Isis Osiris Seth Nephthys

10·3 Empty rows

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

1					
2					
3					
4					

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

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

`$\pgfmath$` 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 `$$\FPeval$$` (fp package)

10.6 Double entry array: the Body Mass Index

```
\def\weightlist{weight in kg, 30, 45, 50, 55, 60, 65, 70, 80, 90, 95, 100, 110, 120}
\def\heightlist{height in cm, 190, 180, 170, 160, 150, 140, 130}
\colorlet{denutrition}{Crimson} \colorlet{skinniness}{yellow} \colorlet{normal}{LawnGreen}
\colorlet{obesity}{Moccasin} \colorlet{serious obesity}{DarkOrange} \colorlet{morbid obesity}{Fuchsia}
\newcommand*\mycolor[1]{\cellcolor{%
 \ifdim#1pt<17pt denutrition\else \ifdim#1pt<19pt skinniness\else \ifdim#1pt<26pt normal\else
 \ifdim#1pt<30pt obesity\else     \ifdim#1pt<35pt serious obesity\else   morbid obesity\fi\fi\fi\fi}}

\begin{tabu}{*{\countvalues\heightlist}{|c}|} \hline
\repeatcell \heightlist {%
   rows = \weightlist,
   height = \ifnum\column>1 \c\else 1\fi,
   weight = \ifnum\row>1 \r\else 1\fi,
   BMI = {$$round(\weight/((\height/100)^2),1)$$},
   text = \mycolor{\BMI}\BMI,
   text/row1 = \c,      font/row1 = \color{NavyBlue},        cr/row1 = \\ \hline,
   text/col1 = \r,      font/col1 = \color{DarkViolet},
}              \\ \hline
\end{tabu}\quad
\begin{tabu}{|c|}
\repeatcell{denutrition,skinniness,normal,obesity,serious obesity,morbid obesity}{%
       text=\cellcolor{\c}{\c},transpose,font=\large\bfseries\bottopstrut}
\end{tabu}
```

height in cm / weight in kg	190	180	170	160	150	140	130
30	8.3	9.3	10.4	11.7	13.3	15.3	17.8
45	12.5	13.9	15.6	17.6	20.0	23.0	26.6
50	13.9	15.4	17.3	19.5	22.2	25.5	29.6
55	15.2	17.0	19.0	21.5	24.4	28.1	32.5
60	16.6	18.5	20.8	23.4	26.7	30.6	35.5
65	18.0	20.1	22.5	25.4	28.9	33.2	38.5
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

denutrition
skinniness
normal
obesity
serious obesity
morbid obesity

weight in kg / height in cm	30	45	50	55	60	65	70	80	90	95	100	110	120
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⟩]{
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⟩]{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 $\mathcal{AMS}$ \intertext in math formulæ, \interitem allows to put some non intented 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 (*ie.* if ⟨*width*⟩≤ width of ⟨*text*⟩). Example:

truncate: \truncate{⟨*5cm*⟩}{⟨*Hello world!*⟩} → Hello word!

trunctext: \trunctext{⟨*5cm*⟩}{⟨*Hello world!*⟩} → Hello world!

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

\htruncadjust{⟨*total width*⟩}[⟨$x_1, x_2, \ldots, x_n$⟩]{⟨$\texttt{text}_1, \texttt{text}_2, \ldots, \texttt{text}_n$⟩}

This macro will measure the width of $\texttt{text}_1$,$\texttt{text}_2$,...,$\texttt{text}_n$ and adjust them for truncation:

text	width	adusted width	γ (truncated?)	Returned width
text_1	w_1	$w'_i = \min\left(w_1, \frac{x_1}{\sum x_i} \times \langle \textit{total width} \rangle\right)$	$\gamma_1 = \begin{cases} 1 & \text{if } w'_1 < w_1 \\ 0 & \text{otherwise} \end{cases}$	$\omega_1 = w'_1 + \gamma_1 \cdot \Phi \cdot \frac{x_1}{\sum \gamma_i x_i}$
text_2	w_2	$w'_i = \min\left(w_2, \frac{x_2}{\sum x_i} \times \langle \textit{total width} \rangle\right)$	$\gamma_2 = \begin{cases} 1 & \text{if } w'_2 < w_2 \\ 0 & \text{otherwise} \end{cases}$	$\omega_2 = w'_2 + \gamma_2 \cdot \Phi \cdot \frac{x_2}{\sum \gamma_i x_i}$
text_3	w_3	$w'_i = \min\left(w_3, \frac{x_3}{\sum x_i} \times \langle \textit{total width} \rangle\right)$	$\gamma_3 = \begin{cases} 1 & \text{if } w'_3 < w_3 \\ 0 & \text{otherwise} \end{cases}$	$\omega_3 = w'_3 + \gamma_3 \cdot \Phi \cdot \frac{x_3}{\sum \gamma_i x_i}$
text_4	w_4	$w'_i = \min\left(w_4, \frac{x_4}{\sum x_i} \times \langle \textit{total width} \rangle\right)$	$\gamma_4 = \begin{cases} 1 & \text{if } w'_4 < w_4 \\ 0 & \text{otherwise} \end{cases}$	$\omega_4 = w'_4 + \gamma_4 \cdot \Phi \cdot \frac{x_4}{\sum \gamma_i x_i}$
		Φ =freespace=⟨*total width*⟩$-\sum w'_i$		$\sum \omega_i \leqslant \langle \textit{total width} \rangle$

Another example of a tabular made with ⇝\repeatcell

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

\trunctext[⟨*TruncateMarker*⟩]{\htruncwidth{⟨*i*⟩}}{⟨$\textit{text}_i$⟩}.

x_i can be any real number provided that $\sum_i x_i \leqslant 16\,383$=\maxdimen in points.

Measuring the widths occur inside a \hbox, thus $\texttt{text}_1$,..., $\texttt{text}_n$ 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 $\{\texttt{text}_1, \ldots, \texttt{text}_n\}$. In other words, $\texttt{text}_1$,...,$\texttt{text}_n$ 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 quite...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 embeded (*ie.* attached) to the .pdf file are the ones specified by \includeonly.

\embedthisfile* embeds only the main file.

This macro could disapear from interface-embedfile if it is added to the main package... Depending on Mr Oberdiek 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

/.blank	replace blank value by another value
/.is private	throws an error if the key is used in an assignment
/.store in cs /.estore in cs	like /.store in but with \csname...\endcsname
/.true /.false /.on /.off	/.is choice shortcut for boolean keys
/.is multiple choice	like /.is choice but more than on choice can be selected at a time
/.is enum	/.is choice shortcut for enumerated keys also provides the /key/.number key to get the numeric index of the selected choice.
/.store as code /.estore as code	store value (*expanded value*) in the key. Created + and * keys to append code
/.store as code in cs /.estore as code in cs	store value (*expanded value*) in a cs-name rather than in the key. Created + and * keys to append code
/.store as color /.store as color command	store the color (*the color command*) in the key.
/.store as color in cs /.store as color command in cs	the same, store in the cs-name rather than in the key
/.store as number /.estore as number	store an integer in the key (- + and * keys are provided to add, substract and multiply)
/.store as number in cs /.estore as number in cs	the same, but stores into a cs-name rather than in the key
/.store as glue /.estore as glue	store a glue in the key (+ - and * keys are provided to add, substract or scale)
/.store as glue in cs /.estore as glue in cs	the same, store in a given named control sequence
/.store in skip cs	store the glue inside a \skip register rather than in a macro
/.store as dimen /.estore as dimen	store a dimen in the key (with + - and * keys for addition, substraction and scaling)
/.store as dimen in cs /.estore as dimen in cs	the same, store in a given named control sequence rather than in the key itself
/.store in dimen cs	store the dimension into a \dimen register rather than in a macro

The command \pgfkeysmeaning is provided (with \providecommand): 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 interface-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* ⇝ `\sectionformat` *(in interface-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:

```
color=\color{yellow}        color=\normalcolor
color=blue                  color=[rgb]{1,0,0}
```

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

```
color=yellow                color=default   defined by interfaces if not already
color=blue                  color=[rgb]{1,0,0}      defined \AtBeginDocument
```

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

```
color=\normalcolor          color=\color{yellow}
color=\color{blue}          color=\color[rgb]{1,0,0}
```

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 ⇝\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 sourrounded by \glueexpr...\relax... and is stored inside the key.

Furthermore:

the + key adds to the length: \glueexpr old value + ⟨ *argument* ⟩ \relax.
the - key substracts: \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 \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 substract) 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 LaTeX to store the font size (and size+ adds, size- substracts, 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 disapear from interfaces ⌣.

18·1 † New macros provided with \providecommand

\removefrom ⟨*hook*⟩{⟨*code*⟩}	\removefromstart⟨*hook*⟩{⟨*code*⟩}
\gremovefrom⟨*hook*⟩{⟨*code*⟩}	\gremovefromstart⟨*hook*⟩{⟨*code*⟩}
\eremovefrom⟨*hook*⟩{⟨*code*⟩}	\eremovefromstart⟨*hook*⟩{⟨*code*⟩}
\xremovefrom⟨*hook*⟩{⟨*code*⟩}	\xremovefromstart⟨*hook*⟩{⟨*code*⟩}

\removefrom removes the first occurence 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 occurence 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*** interface-titlesec and interface-bookmark are loaded.

20·1 † LaTeX

❶

Patched command	Type	Importance	Module	ref. page
\@pkgextension **\@clsextension** **\@ifpackageloaded** **\@ifclassloaded**		minor optimisation or correction	interfaces-LaTeX	71

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.

❷

Patched command	Type	Importance	Module	ref. page
\addcontentsline	*prepend*	important but no side effect	interface-titlesec interface-bookmark	39

\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@ttl@BKM, placed at the beginning of \addcontentsline.

❸

Patched command	Type	Importance	Module	ref. page
\@input		important but no side effect	interface-scrlfile	72

The \input primitive (saved by LaTeX in 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

❹

Patched command	Type	Importance	Module	ref. page
\bookmark	*modif*	important	interface-bookmark	41

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 (*ie.* \aftergroup) ?

20·3 † colortbl

❺

Patched command	Type	Importance	Module	ref. page
\CT@cellc **\endarray**	*corr.*	important but no side effect	interface-makecell	66

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 13 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]#3{%  #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}%
}
```

\endarray: missed \@arrayleft in the redefinition of colortbl, then compatibility with delarray is broken. Bug fixed by interfaces-base.

Same problem for arydshln who redefines \endarray, taking care of colortbl but not of delarray. Bug fixed by interfaces-base.

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).

20·4 † hypbmsec

❻

Patched command	Type	Importance	Module	ref. page
\hbs@do **\hbs@process**	*prepend*	important but no side effect	interfaces-hypbmsec interfaces-titlesec	44

In order to get \lastsectiontitle, \lastsectionbookmark and \lastsectiontoc updated at the right place (*ie. 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

20·5 † enumitem

❼

Patched command	Type	Importance	Module	ref. page
\setlist	*redef*	minor optimisation or correction	interfaces-enumitem	66

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

20·6 † pgfkeys.code.tex

❽

Patched command	Type	Importance	Module	ref. page
\pgfkeys@unpack **\pgfkeys@case@three**	*redef*	important	interfaces-pgfkeys	74

\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    [2010/11/15 v2.1 - pgfkeys (pgf 2.0 et 2.1 macros for interfaces (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 \itf@tp{fancyvrb}
24 \if\relax\@ptionlist{interfaces.sty}\relax % no option given
25    \ExecuteOptions\@declaredoptions        % all interfaces are active
26 \fi
27 \ProcessOptions
28 ⟨/package⟩
```

21.2 Interfaces-base

```
29 ⟨*base⟩
30 \ProvidesPackage{interfaces-base}
31    [2010/10/31 v2.0 Basic macros for interfaces package (FC)]
32 \RequirePackage{etex,ltxcmds,etoolbox,pgfkeys,scrlfile}
33 \AtEndOfPackage{%
34    \RequirePackage{interfaces-etoolbox}%
35    \RequirePackage{interfaces-scrlfile}%
36    \RequirePackage{interfaces-pgfkeys}
37    \RequirePackage{interfaces-tikz}%
38    \RequirePackage{interfaces-umrand}% There is a \BeforePackage here !
39 }% AtEndOfPackage
```

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

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

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


57 \newrobustcmd*\itf@Check@Package[1]{\ltx@ifpackageloaded{#1}
58    {}
59    {\PackageError{interfaces-#1}
60       {The package `#1' must be loaded before `interfaces-#1'
61       \MessageBreak Please refer to `interfaces.pdf' document
62       \MessageBreak for correct usage of the `interfaces' package}\@ehd
63    }%
64 }% \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 LaTeX's \@ifpackageloaded – this macros is not restricted to the document preamble.

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

21.2.1 Bug Fixes

David Carlisle's colortbl is not compatible with David Carlisle's delarray!!

Bug Fixed by interfaces-base. At the same time, the bug exist as well in arydshln and we fix it.

\endarray

```
68 \AfterPackage*{colortbl}{%
69    \renewcommand*\endarray{%
70       \crcr \egroup \egroup \@arrayright \gdef\@preamble{}\CT@end}%
71 }% AfterPackage colortbl
72 \AfterPackage*{arydshln}{%
73    \ltx@ifpackageloaded{colortbl}
74       {% colortbl
75       \def\endarray{%
76        \adl@endarray \egroup \adl@arrayrestore \CT@end \egroup %<original>
77        \@arrayright     % <FC>
78        \gdef\@preamble{}% <FC>
79                    }%
80       }
81       {% no colortbl
82       \def\endarray{%
83        \adl@endarray \egroup \adl@arrayrestore \egroup %<original>
84        \@arrayright     % <FC>
85        \gdef\@preamble{}% <FC>
86                     }%
87       }%
88 }% AfterPackage arydshln
```

21.2.2 Interfaces patches

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

```
\AfterPackage{colortbl}{%
   {\ltx@ifpackageloaded{arydshln}
      {\PackageWarning{interfaces-base}
         {The package arydshln should be loaded after colortbl
         \MessageBreak because both redefine \string\@classz}}%
      {}}%
}% 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.

```
\newrobustcmd*\itf@preapptocmd[5]{%
   \pretocmd#1{#2}%
      {\apptocmd#1{#3}%
         {#4}%
         {#5}}%
      {#5}%
}% \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...

```
\def\itf@patch{\@makeother\#\futurelet\itf@tp\itf@patch@}
\def\itf@patch@{\csname itf@patch@%
      \ifx\itf@tp\patchcmd inside\else
      \ifx\itf@tp\itf@preapptocmd inside\else
      boundary\fi\fi
   \endcsname
}% \itf@patch@
\def\itf@patch@boundary#1#2#3#4#5{%
   \catcode`\#=6%
   \let\itf@tp#2%
   #1\itf@tp{#3}%
   {\let#2\itf@tp
    \AtBeginDocument{#4}% in order to keep this info even after \dump
   }
   {#5}%
}% \itf@patch@boundary (for \pretocmd and \apptocmd)
\def\itf@patch@inside#1#2#3#4#5#6{%
   \catcode`\#=6%
   \let\itf@tp#2%
   #1\itf@tp{#3}{#4}%
   {\let#2\itf@tp
    \AtBeginDocument{#5}%
   }
   {#6}%
}% \itf@patch@inside (for \patchcmd)
```

21.2.3 Helper macros

```
\newskip\itf@sk
\newmuskip\itf@musk
\newtoks\itf@tk
\newcount\itf@cnt
```

\itf@strip@at
\itf@strip@dot

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

```
%%   H E L P E R   M A C R O S
\begingroup\makeatother
   \csgdef{itf@strip@at}#1@{}
\endgroup
\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\macrototest
\macrototest 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$\tau$ against \relax with \ifcat. If false τ cannot be a macro then we leave.
Then we test if τ is expandable: \ifx$\tau$=\expandonce$\tau$, τ is not expandable and we leave. (may be \def$\tau${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.

```
\def\itf@expandonceifcs#1{\expandafter\itf@exp@ndonceifcs\expandafter#1#1Q\@nnil}
\def\itf@exp@ndonceifcs#1#2#3\@nnil{%
   \def\itf@tp{#2}%
   \ifx\itf@tp#1%
   \ifcat\relax\noexpand#2%
      \expandafter\expandafter\expandafter\remove@to@nnil
         \expandafter\ifx\expandafter#2#2\@nnil\else\@nnil\let#1#2\fi
   \fi\fi
}% \itf@exp@ndonceifcs
```

21.2.4 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 LaTeX\@defaultunits.

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

```
%%   G E T   A   N U M B E R (if possible)
%% USAGE: \itf@number\count<text>\relax\@nnil (like \@defaultunits)
\def\itf@number#1#2\relax\@nnil{% \relax is to mimick the syntax of \@defaultunits
   \afterassignment\itf@number@#1\number0#2\relax\itf@number@
}
\def\itf@number@#1\relax#2\itf@number@{%
   \ifblank{#1}
      {\itf@number@@{}#2\itf@number@@\@nnil
      \csname ltx@\ifx\@let@token\itf@number@@
         first\else second\fi oftwo\endcsname}
      \ltx@secondoftwo
}% \itf@number@
\def\itf@number@@#1{\futurelet\@let@token\itf@number@@@}
\def\itf@number@@@{%
   \csname \ifx\@let@token\relax itf@number@@%
   \else remove@to@nnil%
   \fi \endcsname
}% \itf@number@@@
```

21.2.5 Units and math units

\itf@strip@mu This is the equivalent of LaTeX's \strip@pt but for math units:

```
%%   U N I T S   F O R   D I M E N S
\def\itf@strip@mu{\expandafter\itf@rem@mu\the}
{%
\@makeother\M\@makeother\U
\lowercase{%
   \gdef\itf@rem@mu#1.#2MU{#1\ifnum#2>\z@.#2\fi}%
}}%
```

\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.

```
%% \skip = \itf@units@mutopt \muskip
\def\itf@units@pttomu#1{\gluetomu\glueexpr
   \glueexpr#1\relax*1800/%
      \expandafter\itf@strip@dot\the\dimexpr
            \fontdimen6\textfont2*100\relax.\@nil
   \relax
}% \itf@units@pttomu
\def\itf@units@mutopt#1{\glueexpr
   \mutoglue\muexpr#1\relax*
      \expandafter\itf@strip@dot\the\dimexpr
            \fontdimen6\textfont2*100\relax.\@nil/1800
   \relax
}% \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.

```
%% \muskip = \itf@units@mu{ \skip or \muskip or number}
{%
\@makeother\P\@makeother\T\@makeother\S\@makeother\B
\@makeother\C\@makeother\M\@makeother\D\@makeother\I\@makeother\N
\lowercase{%
\gdef\itf@units@mu#1#2{\begingroup
   \edef\itf@tp{#2}%
   \@tfor\x:={PT}{pt}{PC}{pc}{IN}{in}{BP}{bp}%
             {CM}{cm}{MM}{mm}{DD}{dd}{CC}{cc}{SP}{sp}\do{%
      \@expandtwoargs\in@\x\itf@tp\ifin@\@break@tfor\fi
   }%
   \expandafter\endgroup
   \ifin@
      #1=\itf@units@pttomu{#2}%
   \else
      \@defaultunits#1 #2mu\relax\@nnil
   \fi
}% \itf@units@mu
}}
```

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

```
%% \skip = \itf@units@pt( \muskip or \skip or number )
{%
\@makeother\M\@makeother\U
\lowercase{%
\gdef\itf@units@pt#1#2{\begingroup
   \@expandtwoargs\in@{MU}{#2}%
   \ifin@\else\@expandtwoargs\in@{mu}{#2}\fi
   \expandafter\endgroup
```

```
211    \ifin@
212       #1=\itf@units@mutopt{#2}%
213    \else
214       \@defaultunits#1 #2pt\relax\@nnil
215    \fi
216 }}}% \itf@units@pt
```

\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.

```
217 %% glue * 1.5 = glue * 98302 / 65536
218 %% glue * 3/2 = glue * 3/2
219 \def\itf@units@scale#1!{\itf@units@scale@normalize#1/\@nnil/\@nil}
220 \def\itf@units@scale@normalize#1/#2/#3\@nil{%
221    \ifx#2\@nnil
222       \number\dimexpr#1pt\relax/65536
223    \else#1/#2
224    \fi
225 }% \itf@units@normalize
```

21.2.6 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 xolor, but it works with color).

```
226 \AtBeginDocument{%
227    \ltx@ifpackageloaded{xcolor}
228       {\@ifundefinedcolor{default}
229          {\colorlet{default}{.}}
230          {}}%
231       {\ltx@ifpackageloaded{color}
232          {\@ifundefined{\string\color @default}
233             {\expandafter\def\csname\string\color @default\endcsname{\default@color}}
234             {}}%
235       {}}%
236 }
```

```
237 ⟨/base⟩
```

21.3 † Interface LaTeX

```
238 ⟨*LaTeX⟩
239 \ProvidesPackage{interfaces-LaTeX}
240    [2010/10/31 v2.0 pgfkeys macros for LaTeX (FC)]
241 \RequirePackage{interfaces-base}
242 \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.

```
\newrobustcmd*\changefont[1]{\itf@pgfkeys
   \pgfqkeys{/itf/LaTeX/changefont}{#1}\selectfont
}% \changefont
\AfterPackage*{hyperref}
   {\pdfstringdefDisableCommands{\let\changefont\@gobble}}


\pgfqkeys{/itf/LaTeX/changefont}{%
   encoding/.code=\fontencoding{#1},encoding/.value required,%
      enc/.alias=encoding,%
   family/.code=\fontfamily{#1},family/.value required,%
      fam/.alias=family,%
   series/.code=\fontseries{#1},series/.value required,%
      ser/.alias=series,%
   shape/.code=\fontshape{#1},shape/.value required,%
      sha/.alias=shape,%
   size/.estore as dimen in cs=f@size,%
   size/.append code=\set@fontsize\f@linespread\f@size\f@baselineskip,%
      siz/.alias*=size,%
   baselineskip/.store as glue in cs=f@baselineskip,%
   baselineskip/.append code=\set@fontsize\f@linespread\f@size\f@baselineskip,%
      skip/.alias*=baselineskip,%
   linespread/.estore as dimen in cs=f@linespread,%
   linespread/.add code={\ifx\f@linespread\@empty\let\f@linespread\@ne\fi}%
                        {\edef\f@linespread{\expandafter\rem@pt\f@linespread}%
                         \set@fontsize\f@linespread\f@size\f@baselineskip},%
   linespread+/.prefix code=\ifx\f@linespread\@empty\let\f@linespread\@ne\fi,%
   linespread-/.prefix code=\ifx\f@linespread\@empty\let\f@linespread\@ne\fi,%
   linespread*/.prefix code=\ifx\f@linespread\@empty\let\f@linespread\@ne\fi,%
      spread/.alias*=linespread,%
}% \pgfkeys definitions for \changefont
\itf@AfterPackage{color}{%
   \pgfkeysifdefined{/itf/LaTeX/changefont/color/.@cmd}{}
      {\pgfkeys{/itf/LaTeX/changefont/color/.code=\color{#1}}}%
}% AfterPackage color
\itf@AfterPackage{xcolor}{%
   \pgfkeysifdefined{/itf/LaTeX/changefont/color/.@cmd}{}
      {\pgfkeys{/itf/LaTeX/changefont/color/.code=\color{#1}}}%
}% AfterPackage xcolor

⟨/LaTeX⟩
```

21.4 † Interface TikZ (☞pgfmanual)

```
⟨*tikz⟩
\ProvidesPackage{interfaces-tikz}
   [2010/10/31 v2.0 load tikz libraries easily - pgf 2.0 & 2.1 (FC)]
\RequirePackage{interfaces-base}
\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.

```
%% Some catcode checking
\AfterPackage*{tikz}{%
   \AtEndPreamble{%
      \tikzset{every picture/.add style={%
            /utils/exec={\@makeother\!\@makeother\:}}{}}%
   }%
}%
```

21.4.2 List of pgf/TikZ packages and libraries

\alltikzpackages Record "modules" that must be loaded with \usepackage:

```
\newcommand*\alltikzpackages{%
   tikz,pgf,pgfcore,pgfcalendar,pgffor,pgfkeys,pgfpages,pgfrcs,tikzexternal,xxcolor,%
   pgfplots,pgfsys,pgfmath,%
   pgfpic2e,fp,%          + fp required for fixedpointarithmetic
}% \alltikzpackages
```

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

```
\newcommand*\allpgfmodules{datavisualization,%
decorations,%
matrix,%
oo,%
parser,%
plot,%
shapes,%
snakes,%
sorting%
}% \allpgfmodules
```

\alltikzlibraries Record modules that must be loaded with \usetizlibrary.

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

```
\newcommand*\alltikzlibraries{3d,%
arrows,%
automata,%
backgrounds,%
calc,%
calendar,%
chains,%
circuits,%
circuits.ee,%
circuits.ee.IEC,%
circuits.logic,%
circuits.logic.CDH,%
circuits.logic.IEC,%
circuits.logic.US,%
datavisualization,%
datavisualization.3d,%
datavisualization.barcharts,%
datavisualization.formats.functions,%
datavisualization.polar,%
datavisualization.sparklines,%
decorations,%
decorations.footprints,%
decorations.fractals,%
decorations.markings,%
decorations.pathmorphing,%
decorations.pathreplacing,%
decorations.shapes,%
decorations.text,%
er,%
external,%
fadings,%
fit,%
fixedpointarithmetic,%
folding,%
fpu,%
graphs,%
```

```
graphs.basic,%
intersections,%
lindenmayersystems,%
matrix,%
mindmap,%
patterns,%
petri,%
plothandlers,%
plotmarks,%
positioning,%
scopes,%
shadings,%
shadows,%
shapes,%
shapes.arrows,%
shapes.callouts,%
shapes.gates.logic.IEC,%
shapes.gates.logic.US,%
shapes.geometric,%
shapes.misc,%
shapes.multipart,%
shapes.symbols,%
snakes,%
spy,%
svg.path,%
through,%
topaths,%
trees,%
turtle%
}% \alltikzlibraries
```

21.4.3 pgfkeys definitions for \usetikz

```
\pgfqkeys{/itf/tikz}{%
   basic/.code={%
      \usetikzlibrary{calc,%
                      arrows,%         for arrows
                      backgrounds,%    for framed
                      scopes%          for scopes and shortcut : { [scope] ... }
      }%
   },%
   basic/.value forbidden,%   \alltikzpackages
   base/.alias=basic,%
   all/.code={%
      \expandafter\RequirePackage\expandafter{\alltikzpackages}%
      \expandafter\usepgfmodule\expandafter{\allpgfmodules}% see pgfcorescopes.code.tex
      \expandafter\usetikzlibrary\expandafter{\alltikzlibraries}% from:  tikz.code.tex
   },%
   .unknown/.code=\itf@usetikzcode\pgfkeyscurrentname
}% \pgfkeys for \usetikz
```

\usetikz

```
\newrobustcmd*\usetikz[1]{%
   \RequirePackage{tikz}%
   \ifblank{#1}
      {\pgfqkeys{/itf/tikz}{all}}
      {\pgfqkeys{/itf/tikz}{#1}}%
   \itf@pgfkeys% overwritten by \input primitive otherwise
}% \usetikz
```

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

```
\newrobustcmd*\itf@usetikzcode[1]{\begingroup
   \edef\x{%
      \noexpand\in@{,#1,}{,\alltikzpackages,}%
      \noexpand\ifin@  \noexpand\usepackage{#1}%
      \noexpand\else
         \noexpand\in@{,#1,}{,\allpgfmodules,}%
         \noexpand\ifin@ \noexpand\usepgfmodule{#1}%
         \noexpand\else  \noexpand\usetikzlibrary{#1}% \usetikzlibrary by default
         \noexpand\fi
      \noexpand\fi
   }\expandafter\endgroup\x
}% \itf@usetikzcode
```

⟨/tikz⟩

21.5 Interface ☞ titlesec : \sectionformat

⟨*titlesec⟩

```
\ProvidesPackage{interfaces-titlesec}
   [2010/11/15 v2.1 pgfkeys macros for titlesec (FC)]
\RequirePackage{interfaces-base}
\RequirePackage{auxhook}% for fix for \ttlp@append (odd/even pages, not implemented in interf
\itf@Check@Package{titlesec}
\itf@pgfkeys
```

21.5.1 titlesec modifications

```
%%   t i t l e s e c   "p a t c h e s"
\robustify\titleclass   \robustify\titleformat  \robustify\titlespacing
\AddLineBeginMainAux{%
   \string\providecommand\string\ttlp@append[2]{}%
}
\BeforeFile*{ttlkeys.def}{%
   \catcode32 9\relax                  % ignore spaces
   \count@\endlinechar\endlinechar\m@ne % and line feeds
}
\AfterFile*{ttlkeys.def}{% restore spaces and line feeds
   \catcode32 10\relax
   \endlinechar\count@
}
```

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

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

21.5.2 \sectionformat

\sectionformat \sectionformat\subsection* [⟨*shape*⟩]{⟨*key=value*⟩}

```
\newrobustcmd*\sectionformat[1]{\@ifstar
   {\itf@sectionformat{#1}{/*}}
   {\itf@sectionformat{#1}{}}%
}% \sectionformat
\def\itf@sectionformat#1#2{\@testopt
   {\expandafter\expandafter\expandafter%
      \itf@sectionf@rmat\expandafter\@gobble\string#1\@nil{#2}}{}%
}% \sectionformat
\def\itf@sectionf@rmat#1\@nil#2[#3]#4{\itf@pgfkeys
```

```
445    \ifblank{#3}
446       {\pgfqkeys{/itf/titlesec/#1#2}
447             {/itf/titlesec/@setup={#1}{#2},#4,/itf/titlesec/@update={#1}{#2}}}
448       {\pgfqkeys{/itf/titlesec/#1#2}
449             {/itf/titlesec/@setup={#1}{#2},#4,shape={#3},/itf/titlesec/@update={#1}{#2}}}%
450 }% \itf@ttl@sectionformat
```

\itf@ttl@setup
\itf@ttl@getformat
\itf@ttl@getspacing

Macros used to retrieve the values set by \titleformat and/or \titlespacing. This allows to use \sectionformat concurrently with \titleformat and \titlespacing provided by titlesec.

```
451 %% private keys @setup and @update
452 \pgfqkeys{/itf/titlesec}{%
453    @setup/.code 2 args=\itf@ttl@setup{#1}{#2},%
454    @update/.code 2 args=\expandafter\itf@ttl@update
455                    \csname#1\endcsname{/itf/titlesec/#1}{#2}%
456 }
457 \newrobustcmd*\itf@ttl@setup[2]{%
458    \pgfkeysifdefined{/itf/titlesec/#1#2/shape}{}
459       {\itf@ttl@newsection{#1}{#2}}%
460    \ifcsundef{ttls@#1#2}{}% no previous spacing settings
461       {\expandafter\expandafter\expandafter%
462       \itf@ttl@getspacing\csname ttls@#1#2\endcsname{/itf/titlesec/#1#2}}%
463    \ifcsundef{ttlf@#1#2}{}% no previous format settings
464       {\expandafter\expandafter\expandafter%
465       \itf@ttl@getformat\csname ttlf@#1#2\endcsname{/itf/titlesec/#1#2}}%
466 }% \itf@ttl@setup
467 \def\itf@ttl@setkey#1#2{\begingroup
468    \def\x{\pgfkeysvalueof{#1}}\def\y{#2}%
469    \expandafter\endgroup\ifx\x\y\else
470       \pgfkeyssetvalue{#1}{#2}\fi
471 }% \itf@ttl@setkey
472 \def\itf@ttl@getspacing#1#2#3#4#5#6{%
473    \itf@ttl@setkey{#6/left}{#1}%
474    \itf@ttl@setkey{#6/right}{#2}%
475    \itf@ttl@setkey{#6/top}{#3}%
476    \itf@ttl@setkey{#6/bottom}{#4}%
477    \itf@ttl@setkey{#6/indent first}{#5}%
478 }% \itf@ttl@getspacing
479 \def\itf@ttl@getformat#1#2{%
480    \csname ltx@\ifx\pgfkeysvalueof#1first\else second\fi oftwo\endcsname
481       {\itf@ttl@get@format{}}
482       {\itf@ttl@get@format{#1}{#2}}%
483 }% \itf@ttl@getformat
484 \def\itf@ttl@get@format#1#2#3#4#5#6#7{%
485    \if\relax\detokenize{#1}\relax\else
486    \pgfkeysEsetvalue{#7/shape}{\expandafter\itf@strip@at\string#1}%
487    \fi
488    \itf@ttl@setkey{#7/font}{#2}%
489    \itf@ttl@setkey{#7/label}{#3}%
490    \itf@ttl@setkey{#7/sep}{#4}%
491    \itf@ttl@setkey{#7/before}{#5}%
492    \itf@ttl@setkey{#7/after}{#6}%
493 }% \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).

```
494 \newrobustcmd*\itf@ttl@update[3]{%
495    \ifblank{#3}
496       {\ttl@getkeys{#1}{titlesec}}
497       {\ttl@getkeys{name=#1,numberless}{titlesec}}%
498    \pgfkeysgetvalue{#2#3/shape}\itf@tp
499    \ifcsundef{ttlh@\itf@tp}
```

```
      {\begingroup \makeatletter
      \InputIfFileExists{\itf@tp.tss}{}{%
      \PackageError{titlesec}{Unknown shape}%
         {Shapes are defined in files with extension tss\MessageBreak
         Either you have misspelled the shape\MessageBreak
         or there is no a \itf@tp.tss file}}%
      \endgroup}
      {}%
   \ifttl@explicit\@temptokena{\csdef{ttlf@\ttl@a}##1}%
   \else\@temptokena{\csdef{ttlf@\ttl@a}}%
   \fi
   \the\@temptokena{%
      \pgfkeysvalueof{#2#3/shape@cs}%
      {\pgfkeysvalueof{#2#3/font}}%
      {\pgfkeysvalueof{#2#3/label}}%
      {\pgfkeysvalueof{#2#3/sep}}%
      {\pgfkeysvalueof{#2#3/before}}%
      {\pgfkeysvalueof{#2#3/after}}}%
   \csname ttl@compat\ttl@a\endcsname
   \ttl@spacing@ii
      {\pgfkeysvalueof{#2#3/indent first}}%
      {\pgfkeysvalueof{#2#3/left}}%
      {\pgfkeysvalueof{#2#3/top}}%
      {\pgfkeysvalueof{#2#3/bottom}}%
      [\pgfkeysvalueof{#2#3/right}]%
}% \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.

```
%%-------------------------------------------------------------------------
%%   t i t l e c l a s s
\newrobustcmd*\itf@ttl@titleclass{\@ifstar
   {\itf@ttl@titleclass@i*}
   {\itf@ttl@titleclass@i{}}%
}% \itf@ttl@titleclass
\def\itf@ttl@titleclass@i#1#2#3[#4]#5\@nil{\ifblank{#4}
   {%
      \expandafter\titleclass\csname#2\endcsname{#3}%
   }
   {%
      \expandafter\titleclass\csname#2\endcsname{#3}[{#4}]% #4 = sup-level
      \if\relax#1\relax % not starred
         \expandafter\expandafter\expandafter%
            \itf@ttl@titleclass@ii\expandafter\@gobble\string#4\relax{#2}%  subsection\relax{
      \fi
   }%
   \expandafter\providecommand\csname l@#2\endcsname[2]{}% no entry in toc
}% \itf@ttl@titleclass@i
\def\itf@ttl@titleclass@ii#1\relax#2{%
   \ifcsundef{c@#2}
      {\newcounter{#2}[{#1}]}% creates counter for #1
      {\@addtoreset{#2}{#1}}%
   \csxdef{the#2}{%
      \expandafter\noexpand\csname the#1\endcsname
      .\noexpand\@arabic\expandafter\noexpand\csname c@#2\endcsname
   }% \the#2
   \ifcsundef{l@#1}{}{%
      \expandafter\providecommand\csname l@#2\endcsname[2]{%
         \csname l@#1\endcsname{##1}{##2}%
      }% \l@#2 = \l@newsection is (by default) the same as \l@sup-level
```

```
557    }%
558 }% \itf@ttl@titleclass@ii
```

\itf@ttl@createsection If \sectionformat is invoked with an unknown sectionning command (*ie.* 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.

```
559 %%-------------------------------------------------------------------------
560 %%   C r e a t i n g   n e w   s e c t i o n s
561 \def\itf@ttl@createsection#1{% sectionning command undefined: temporarily defined
562    \expandafter\titleclass\csname#1\endcsname{straight}[*]%     in straight class
563    \itf@AfterPackage{hypbmsec}{%
564       \AfterPreamble{%
565          \csletcs{hbs@#1}{#1}%
566          \csxdef{#1}{\noexpand\hbs@do\expandafter\noexpand\csname hbs@#1\endcsname}%
567       }%
568    }%
569    \itf@AfterPackage{interfaces-tocloft}{%
570       \itf@cft@newsection{#1}{toc}%
571    }%
572    \itf@AfterPackage{interfaces-hypbmsec}{%
573       \expandafter\let\csname last#1title\endcsname\relax
574       \expandafter\let\csname last#1bookmark\endcsname\relax
575       \expandafter\let\csname last#1toc\endcsname\relax
576    }%
577 }% \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.

```
578 %%-------------------------------------------------------------------------
579 %%   D e f i n i n g   k e y s   f o r   e a c h   s e c t i o n
580 \def\itf@ttl@newsection#1#2{% #1=section, #2=/* or empty
581    \ifcsundef{#1}% sectionning command undefined: temporarily defined in straight class
582       {\itf@ttl@createsection{#1}}
583       {}%
584    \ifblank{#2}{%
585       \pgfqkeys{/itf/titlesec/#1}{% these keys do not exist for starred/numberless sections
586          class/.code=\itf@ttl@titleclass{#1}##1[]\@nil,% with \@addtoreset
587             class/.value required,%                    if sup-level is given
588             class*/.code=\itf@ttl@titleclass*{#1}##1[]\@nil,% without \@addtoreset
589             class*/.value required,%                      even if sup-level is given
590          shape/.initial=,% initial shapes set by titlesec.sty for not numbered sections.
591          the#1/.store in cs=the#1,%
592             the#1/.default=\pgfkeysvalueof{/itf/titlesec/#1/label},%
593       }}
594       {% starred (numberless) unless titlesec, we do not put the default shape to hang,
595       \pgfkeysifdefined{/itf/titlesec/#1/shape}%   but to the shape of the not starred
596          {\pgfkeysgetvalue{/itf/titlesec/#1/shape}\itf@tp}%        sectionning command
597          {\def\itf@tp{hang}}%
598       \pgfkeyslet{/itf/titlesec/#1#2/shape}\itf@tp%
599       }%
600    \pgfqkeys{/itf/titlesec/#1#2}{%
601       shape/.value required,%
602       shape@cs/.initial=
603          \ifdefined\itf@last@from@next\itf@last@from@next{#1}\fi % interfaces-hypbmsec
```

```
604         \csname ttlh@\pgfkeysvalueof{/itf/titlesec/#1#2/shape}\endcsname,% constant
605         shape@cs/.is private=interface-titlesec,% constant
606      font/.store as code,%
607         format/.alias*=font,%
608      label/.store as code,%
609      sep/.estore as glue,%
610         labelsep/.alias*=sep,%
611      before/.store as code,%
612      after/.store as code,%
613      indent first/.true=\pgfkeyssetvalue{/itf/titlesec/#1#2/indent first}{\@ne},%
614      indent first/.false=\pgfkeyssetvalue{/itf/titlesec/#1#2/indent first}{\z@},%
615         identfirst/.alias=indent first,%
616         indent first=true,%
617      left/.estore as glue,%
618      top/.estore as glue,%
619      bottom/.estore as glue,%
620      right/.estore as glue,%
621      break/.code=\ifblank{##1}
622                     {\cslet{#1break}\@empty}
623                     {\protected\csdef{#1break}{##1}},%
624         break*/.code=\protected\apptocs{#1break}{##1},%
625         break+/.alias=break*,%
626      mark/.code=\ifblank{##1}
627                  {\cslet{#1mark}\@gobble}
628                  {\protected\long\csdef{#1mark}####1{##1}},%
629         mark/.default=,%
630   }% \pgfkeys
631   \ifblank{#2}{% no default marks for stared (numberless) \section*
632   \itf@AfterPackage{interfaces-marks}{% requires interfaces-marks
633      \pgfkeys{/itf/titlesec/#1/mark/.default=\marksthe{#1}{##1}}%
634   }% AfterPackage interfaces-mark
635   \itf@AfterPackage{interfaces-bookmark}{% requires interfaces-bookmark
636   \pgfqkeys{/itf/titlesec/#1}{% does not exist for starred sections
637      bookmark/.is multiple choice,%
638         bookmark/level/.estore in cs=toclevel@#1,% depth of the bookmark
639            bookmark/level/.default=\ifcsvoid{ttll@#1}{99}{\csuse{ttll@#1}},%
640            bookmark/level/.blank=\ifcsvoid{toclevel@#1}{\ifcsvoid{ttll@#1}{99}{\csuse{ttll@#
641            bookmark/toclevel/.alias=bookmark/level,%
642         bookmark/bold/.code=\ifblank{##1}
643            {\pgfkeyslet{/itf/titlesec/#1/bookmark/bold}\@empty}
644            {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/bold}{bold=##1,}},%
645            bookmark/bold/.default=true,%
646         bookmark/bold*/.code=\ifblank{##1}
647            {\pgfkeyslet{/itf/titlesec/#1/bookmark/bold*}\@empty}
648            {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/bold*}{bold*=##1,}},%
649            bookmark/bold*/.default=true,%
650         bookmark/italic/.code=\ifblank{##1}
651            {\pgfkeyslet{/itf/titlesec/#1/bookmark/italic}\@empty}
652            {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/italic}{italic=##1,}},%
653            bookmark/italic/.default=true,%
654         bookmark/italic*/.code=\ifblank{##1}
655            {\pgfkeyslet{/itf/titlesec/#1/bookmark/italic*}\@empty}
656            {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/italic}{italic*=##1,}},%
657            bookmark/italic*/.default=true,%
658         bookmark/color/.code=\ifblank{##1}
659            {\pgfkeyslet{/itf/titlesec/#1/bookmark/color}\@empty}
660            {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/color}{color=##1,}},%
661            bookmark/color/.default=,%
662         bookmark/color*/.code=\ifblank{##1}
663            {\pgfkeyslet{/itf/titlesec/#1/bookmark/color*}\@empty}
664            {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/color*}{color*=##1,}},%
665            bookmark/color*/.default=,%
666         bookmark/on/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on}{disable=false,},%
667            bookmark/on/.value forbidden,%
668         bookmark/off/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on}{disable=true,},%
```

```
            bookmark/off/.value forbidden,%
            bookmark/none/.alias=bookmark/off,%
         bookmark/style/.code=\ifblank{##1}
               {\pgfkeyslet{/itf/titlesec/#1/bookmark/style}\@empty}
               {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/style}{style=##1,}},%
            bookmark/style/.default=,%
         bookmark/.unknown/.code=\pgfkeysEsetvalue{/itf/titlesec/#1/bookmark/other}{\pgfkeysc
         bookmark/bold/.initial=,%
         bookmark/bold*/.initial=,%
         bookmark/italic/.initial=,%
         bookmark/italic*/.initial=,%
         bookmark/color/.initial=,%
         bookmark/color*/.initial=,%
         bookmark/on/.initial=,%
         bookmark/style/.initial=,%
   }}% \AfterPackage*{interfaces-bookmark}
   }% not starred (numberless) \section
   {}% starred/numberless \section*: do not create the keys
}% \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.

```
%% bookmarks styles for sectionning commands:
\@makeother\#
\itf@AfterPackage{interfaces-bookmark}{%
   \itf@patch\itf@preapptocmd\addcontentsline              % #2=section name in
      {\begingroup\itf@ttl@BKM{/itf/titlesec/#2/bookmark}}%   \addcontentsline
      {\endgroup}
      {\PackageInfo{interfaces-titlesec}
         {Command \string\addcontentsline\space successfully patched!}}
      {\PackageWarning{interfaces-titlesec}
         {Patch of \string\addcontentsline\space failed
         \MessageBreak bookmark-style specs for sections
         \MessageBreak will be unavailable}}%
}% AfterPackage interfaces-bookmark
\catcode`\#=6
\def\itf@ttl@BKM#1{%
   \pgfkeysifdefined{#1/.@cmd}
      {%
      \edef\itf@tp{% do nothing if the keys are not set (shortcut)
         \pgfkeysvalueof{#1/bold}%
         \pgfkeysvalueof{#1/bold*}%
         \pgfkeysvalueof{#1/italic}%
         \pgfkeysvalueof{#1/italic*}%
         \pgfkeysvalueof{#1/color}%
         \pgfkeysvalueof{#1/color*}%
         \pgfkeysvalueof{#1/on}%
         \pgfkeysvalueof{#1/style}}%
      \ifx\itf@tp\@empty
      \else
         \global\pgfkeyslet{#1/bold*}\@empty
         \global\pgfkeyslet{#1/italic*}\@empty
         \global\pgfkeyslet{#1/color*}\@empty
         \expandafter\bookmarksetup\expandafter{\itf@tp}%
         \pgfkeysifdefined{/itf/bookmark/*}%
            {% \bookmarksetup{color*=} overwrites \sectionformat...{bookmark/color*=}
            \pgfkeysgetvalue{/itf/bookmark/*}\itf@tp
            \expandafter\bookmarksetup\expandafter{\itf@tp}%
            }
            {}%
      \fi}
      {}%
```

```
}% \itf@ttl@BKM
```

21.5.4 \bottomtitles

\bottomtitles This is a "skip command": it can be assigned a number or a skip.

```
%%-------------------------------------------------------------------------
%%   b o t t o m t i t l e s
\newrobustcmd*\bottomtitles{\@ifnextchar\bgroup
   \itf@setbottomtitlespace
   {\afterassignment\itf@set@bottomtitle@space\itf@sk}%
}% \bottomtitles
\def\itf@setbottomtitlespace#1{%
   \@defaultunits\itf@sk#1\textheight\relax\@nnil
   \itf@set@bottomtitle@space
}% \itf@setbottomtitlespace
\def\itf@set@bottomtitle@space{\edef\bottomtitlespace{\the\itf@sk}}
```

⟨/titlesec⟩

21.6 † Interface ☞ hyperref

⟨*hyperref⟩

```
\ProvidesPackage{interfaces-hyperref}
   [2010/10/27 v1.2 provides raised hyperdef (FC)]
\RequirePackage{interfaces-base}
\itf@Check@Package{hyperref}
```

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

```
\newrobustcmd\raisedhyperdef[4][]{\begingroup
   \ifblank{#1}
      {}
      {%
      \expandafter\def\expandafter\HyperRaiseLinkHook\expandafter{%
         \expandafter\def\expandafter\HyperRaiseLinkHook
                                    \expandafter{\HyperRaiseLinkHook}% restore current definiti
         \HyperRaiseLinkHook
         \advance\noexpand\HyperRaiseLinkLength by\dimexpr#1\relax\relax}%
      }%
   \ifx \\#2\\\def \Hy@AnchorName {#3}\else
      \def \Hy@AnchorName {#2.#3}%
   \fi
   \ltx@leavevmode
   \Hy@raisedlink{\hyper@anchorstart{\Hy@AnchorName}\hyper@anchorend}%
   \endgroup
   #4%
}% \raisedhyperdef
```

⟨/hyperref⟩

21.7 † Interface ☞ bookmark

```
765 ⟨*bookmark⟩
766 \ProvidesPackage{interfaces-bookmark}
767    [2010/11/15 v2.1 more control on bookmarks (FC)]
768 \RequirePackage{interfaces-base}
769 \RequirePackage{interfaces-hyperref}
770 \RequirePackage{refcount}
771 \itf@Check@Package{bookmark}
772 \itf@pgfkeys
```

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

```
773 \expandafter\itf@patch\expandafter\patchcmd\csname\string\bookmark\endcsname
774    {\immediate\write\@mainaux}{\itf@BKM@write}
775    {\PackageInfo{interfaces-bookmark}
776       {Command \expandafter\string\csname\string\bookmark\endcsname\space
777       successfully patched!}}
778    {\PackageWarning{interfaces-bookmark}
779       {Command \expandafter\string\csname\string\bookmark\endcsname\space
780       could not be patched\MessageBreak
781       Loading of interfaces-bookmark is canceled}%
782       \endinput}
```

21.7.1 star keys* definitions

```
783 \global\let\itf@BKM@enable\@ne % 0 = disable*, 1=enable, \@M = disable
784 \pgfqkeys{/itf/bookmark}{%
785    disable/.true=\global\let\itf@BKM@enable\@M,%
786    disable/.false=\global\let\itf@BKM@enable\@ne,%
787    disable*/.true=\global\let\itf@BKM@enable\z@,%
788    disable*/.false=\global\let\itf@BKM@enable\@ne,%
789    atend/.true=\ifnum\itf@BKM@enable>0\edef\itf@BKM@enable{\number-\itf@BKM@enable}\fi,%
790    atend/.false=\ifnum\itf@BKM@enable<0\edef\itf@BKM@enable{\number-\itf@BKM@enable}\fi,%
791    bold*/.true=\pgfqkeys{/itf/bookmark}{@save={bold=true}},%
792    bold*/.false=\pgfqkeys{/itf/bookmark}{@save={bold=false}},%
793    bold*/@blank/.code=\pgfqkeys{/itf/bookmark/@restore}{bold},%
794    bold*/.blank=@blank,%
795    italic*/.true=\pgfqkeys{/itf/bookmark}{@save={italic=true}},%
796    italic*/.false=\pgfkeys{/itf/bookmark}{@save={italic=false}},%
797    italic*/@blank/.code=\pgfqkeys{/itf/bookmark/@restore}{italic},%
798    italic*/.blank=@blank,%
799    color*/.code={\ifblank{#1}
800       {\pgfqkeys{/itf/bookmark/@restore}{color}}
801       {\pgfqkeys{/itf/bookmark}{@save={color={#1}}}}%
802    },%
803    @save/.code={%
804       \pgfkeysifdefined{/itf/bookmark/@save/color}{}
805          {\pgfkeyslet{/itf/bookmark/@save/color}\BKM@color}%
806       \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{}
807          {\pgfkeyslet{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}%
808       \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}{}
809          {\pgfkeysEsetvalue{/itf/bookmark/@save/grouplevel}{\the\currentgrouplevel}}%
810       \pgfkeysaddvalue{/itf/bookmark/*}{}{,#1}%
811       \bookmarksetup{#1}%
812    },%
813    @restore/color/.code=%
814       \pgfkeysifdefined{/itf/bookmark/@save/color}
815          {\pgfkeysgetvalue{/itf/bookmark/@save/color}\BKM@color}
816          {},%
817    @restore/FontStyle/.code=%
818       \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}
819          {\pgfkeysgetvalue{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}
```

```
820         {},%
821    @restore/.code={%
822 %%    \pgfqkeys{/itf/bookmark/@restore}{color,FontStyle}% repeating the following lines
823       \pgfkeysifdefined{/itf/bookmark/@save/color}% is quicker than setting a /.style ...
824          {\pgfkeysgetvalue{/itf/bookmark/@save/color}\BKM@color}
825          {}%
826       \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}
827          {\pgfkeysgetvalue{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}
828          {}%
829       \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}
830          {% shortcut: \bookmarksetup{bold*} called inside main group (level 0)
831          \ifnum\pgfkeysvalueof{/itf/bookmark/@save/grouplevel}=\z@
832             \global\pgfkeyslet{/itf/bookmark/@save/FontStyle}\@undefined
833             \global\pgfkeyslet{/itf/bookmark/@save/color}\@undefined
834             \global\pgfkeyslet{/itf/bookmark/@save/grouplevel}\@undefined
835             \global\pgfkeyslet{/itf/bookmark/*}\@undefined
836             \global\let\BKM@color\BKM@color
837             \global\let\BS@BKM@FontStyle\BS@BKM@FontStyle
838          \else % restore after each group as needed
839             \pgfkeyslet{/itf/bookmark/@save/FontStyle}\@undefined
840             \pgfkeyslet{/itf/bookmark/@save/color}\@undefined
841             \pgfkeyslet{/itf/bookmark/*}\@undefined
842             \ifnum\currentgrouplevel>\z@
843                \aftergroup\itf@BKM@restore
844             \else
845                \pgfkeyslet{/itf/bookmark/@save/grouplevel}\@undefined
846          \fi\fi
847          }
848          {}%
849    },%
850    @restore/bold/.code={% bold is "hidden" in \BS@BKM@FontStyle
851       \begingroup
852          \edef\itf@tp{\bookmarkget{bold}}%
853          \pgfqkeys{/itf/bookmark/@restore}{FontStyle}%
854          \count@=\bookmarkget{bold}%
855          \edef\itf@tp{\endgroup
856             \ifnum\itf@tp=\count@
857             \else \noexpand\bookmarksetup{%
858                         bold\ifcase\count@ =false\fi}\fi
859          }\itf@tp},%
860    @restore/italic/.code={% italic is "hidden" in \BS@BKM@FontStyle
861       \begingroup
862          \edef\itf@tp{\bookmarkget{italic}}%
863          \pgfqkeys{/itf/bookmark/@restore}{FontStyle}%
864          \count@=\bookmarkget{italic}%
865          \edef\itf@tp{\endgroup
866             \ifnum\itf@tp=\count@
867             \else \noexpand\bookmarksetup{%
868                         italic\ifcase\count@ =false\fi}\fi
869          }\itf@tp},%
870 }
871 \def\itf@BKM@restore{%
872    \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}
873       {\pgfkeysalso{/itf/bookmark/@restore}}
874       {}%
875 }% \itf@BKM@restore
876 %% Plug the BKM keys in the interfaces pgfkeys (not efficient but quickly written...)
877 \define@key{BKM}{disable}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/disable={#1}}}
878 \define@key{BKM}{disable*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/disable*={#1}}}
879 \define@key{BKM}{bold*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/bold*={#1}}}
880 \define@key{BKM}{italic*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/italic*={#1}}}
881 \define@key{BKM}{color*}[]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/color*={#1}}}
882 \define@key{BKM}{atend}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/atend={#1}}}%
```

`\itf@BKM@write` 1) check if the bookmark is disable

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

```
\def\itf@BKM@write#1{%
   \aftergroup\itf@BKM@restore
   \ifcase\itf@BKM@enable
      \global\let\itf@BKM@enable\@ne
   \or
      \immediate\write\@mainaux{#1}%
   \else
      \ifcase -\itf@BKM@enable  % (case -0 = 0 tested before)
      \or                       % (case -1 = enable but At End)
         \let\itf@BKM@enable\@ne
         \edef\itf@tp{\immediate\write\@mainaux{#1}}%
         \expandafter\BookmarkAtEnd\expandafter{\itf@tp}%
      \fi
   \fi
}% \itf@BKM@write
```

21.7.2 ref and pageref keys

\itf@BKM@pageref

```
%% pageref=  key for \bookmark (using refcount : \getpagerefnumber)
\def\itf@BKM@pageref[#1]{%
   \expandafter\itf@BKM@@pageref
      \csname r@#1\endcsname{#1}%
}% \itf@BKM@pageref
\def\itf@BKM@@pageref#1#2{%
   \ifx#1\relax
      \protect\G@refundefinedtrue
      \@latex@warning{%
         Hyper reference `#2' on page \thepage \space undefined
         \MessageBreak for bookmark destination page
       }%
   \fi
   \edef\BKM@page{\getrefbykeydefault{#2}{page}{1}}%
}% \itf@BKM@@pageref
\define@key{BKM}{pageref}{%
   \Hy@babelnormalise\itf@BKM@pageref[{#1}]}
```

\itf@BKM@ref

```
%% ref=  key for \bookmark
\def\itf@BKM@ref[#1]{%
   \expandafter\itf@BKM@@ref
      \csname r@#1\endcsname{#1}%
}% \itf@BKM@ref
\def\itf@BKM@@ref#1#2{%
   \ifx#1\relax
      \protect\G@refundefinedtrue
      \@latex@warning{%
         Hyper reference `#2' on page \thepage \space undefined
         \MessageBreak for bookmark destination
       }%
      \def\BKM@dest{??}%
   \else
      \edef\BKM@dest{\expandafter\@fourthoffive#1\@empty\@empty}%
   \fi
}% \BKM@@ref
\define@key{BKM}{ref}{%
   \Hy@babelnormalise\itf@BKM@ref[{#1}]}
```

```
934 ⟨/bookmark⟩
```

21.8 Interface hypbmsec

21.8.1 interfaces-hypbmsec

```
935 ⟨*hypbmsec⟩
936 \ProvidesPackage{interfaces-hypbmsec}
937       [2010/11/15 v2.1 lastsection etc. (FC)]
938 \RequirePackage{interfaces-base}
939 \itf@Check@Package{hypbmsec}
940 \AfterPackage*{interfaces-titlesec}{% requires interface-titlesec
941    \input{interfaces-hypbmsec.code.tex}%
942 }%
943 ⟨/hypbmsec⟩
```

21.8.2 interfaces-hypbmsec.code.tex

```
944 ⟨*hypbmsec.code.tex⟩
945 \ProvidesFile{interfaces-hypbmsec.code.tex}
946    [2010/10/27 v1.2 code for interfaces-hypbmsec (FC)]
```

\hbs@do (patch)

```
947 \itf@patch\pretocmd\hbs@do{%
948    \edef\itf@tp{\expandafter\itf@strip@at\string#1}%
949    }%
950    {\PackageInfo{interfaces-hypbmsec}
951       {Command \string\hbs@do\space successfully patched}}
952    {\PackageWarning{interfaces-hypbmsec}
953       {Command \string\hbs@do\space could not be patched
954       \MessageBreak loading of ‘interfaces-hypbmsec‘ aborted
955       \MessageBreak \string\lastsection\space will not work}%
956     \endinput}
```

\hbs@process (patch)

```
957 \itf@patch\pretocmd\hbs@process{%
958    \itf@hbs@next{#1}%
959    }%
960    {\PackageInfo{interfaces-hypbmsec}
961       {Command \string\hbs@process\space successfully patched}}
962    {\PackageWarning{interfaces-hypbmsec}
963       {Command \string\hbs@process\space could not be patched
964       \MessageBreak loading of ‘interfaces-hypbmsec‘ aborted
965       \MessageBreak \string\lastsection\space will not work}%
966       \endinput}%
```

\itf@hbs@next

```
967 \long\def\itf@hbs@next#1{%
968    \csdef{itf@next\itf@tp title}{#1}%
969    \expandafter\let\csname itf@next\itf@tp toc\expandafter\endcsname
970       \csname
971          \ifx\hbs@tocstring\relax itf@next\itf@tp title\else
972                                    hbs@tocstring\fi
973       \endcsname
974    \expandafter\let\csname itf@next\itf@tp bookmark\expandafter\endcsname
975       \csname
976          \ifx\hbs@bmstring\relax itf@next\itf@tp toc\else
977                                   hbs@bmstring\fi
978       \endcsname
979    \expandafter\expandafter\expandafter
```

```
980    \expandafter\expandafter\expandafter
981    \expandafter\itf@hbs@chkclass
982          \expandafter\expandafter\expandafter\string
983             \expandafter\hbs@seccmd\string\ttl@\@nil
984       {\expandafter\itf@last@from@next\expandafter{\itf@tp}}%
985 }% \itf@hbs@next
986 \long\expandafter\def\expandafter\itf@hbs@chkclass
987             \expandafter#\expandafter1\string\ttl@#2\@nil{%
988    \ifblank{#1}{\if@nobreak\itf@hbs@nobreakfix#2\@nnil\fi}%
989 }% \itf@hbs@chkclass
990 \def\itf@hbs@nobreakfix#1#{\itf@hbs@n@breakfix}
991 \def\itf@hbs@n@breakfix#1#2\@nnil{%
992    \unless\ifx\lastsectionname\relax
993    \unless\ifnum\csname ttll@#1\endcsname > \csname ttll@\lastsectionname\endcsname\relax
994       \@nobreakfalse
995    \fi\fi
996 }% \itf@hbs@nobreakfix
```

\itf@last@from@next

```
997 \def\itf@last@from@next#1{%
998    \gdef\lastsectionname{#1}%
999    \global\expandafter\let\csname last#1title\expandafter\endcsname
1000                          \csname itf@next#1title\endcsname
1001    \global\expandafter\let\csname last#1toc\expandafter\endcsname
1002                          \csname itf@next#1toc\endcsname
1003    \global\expandafter\let\csname last#1bookmark\expandafter\endcsname
1004                          \csname itf@next#1bookmark\endcsname
1005    \let\@elt\itf@resetlastsections
1006    \csname cl@#1\endcsname
1007 }% \itf@last@from@next
1008 \def\itf@resetlastsections#1{%
1009    \expandafter\let\csname last#1title\endcsname\relax
1010    \expandafter\let\csname last#1toc\endcsname\relax
1011    \expandafter\let\csname last#1bookmark\endcsname\relax
1012    \csname cl@#1\endcsname
1013 }% \itf@resetlastsections


1014 \let\lastsectionname\relax
1015 \@for\itf@tp:=part,section,subsection,subsubsection,paragraph,subparagraph%
1016    \do{\expandafter\let\csname last\itf@tp title\endcsname\relax
1017       \expandafter\let\csname last\itf@tp toc\endcsname\relax
1018       \expandafter\let\csname last\itf@tp bookmark\endcsname\relax
1019    }%
1020 \AtBeginDocument{\ifundef\chapter{}
1021    {\let\lastchaptertitle\relax
1022     \let\lastchaptertoc\relax
1023     \let\lastchapterbookmark\relax}%
1024 }%

1025 ⟨/hypbmsec.code.tex⟩
```

21.9 Interface fancyhdr : \pagesetup

```
⟨*fancyhdr⟩
\ProvidesPackage{interfaces-fancyhdr}
   [2010/11/15 v2.1 pgfkeys macros for fancyhdr (FC)]
\RequirePackage{interfaces-base}
\itf@Check@Package{fancyhdr}
\itf@pgfkeys
\robustify\fancyhf
\robustify\fancyhfoffset
```

21.9.1 \pagesetup

\pagesetup

```
\newrobustcmd*\pagesetup{%
   \let\fancy@gbl\relax % Just in case...
   \@ifstar
      {\itf@pagesetup{}}
      {\itf@pagesetup{reset,}}%
}% \pagesetup
\def\itf@pagesetup#1{\@testopt
   {\itf@p@gesetup{#1}}{}%
}% \itf@pagesetup
\def\itf@p@gesetup#1[#2]#3{\ifblank{#2}
   {% modifies the current page style
      \itf@pgfkeys\pgfqkeys{/itf/fancyhdr}{#1#3}\relax\ps@@fancy}%
   {% defines or modifies a given page style
      \ifblank{#1}
         \csappto% star form: keep existing value (if not overwritten)
         \csdef% no star form: overwrite (+ reset at \pagestyle time)
         {ps@#2}{\itf@pgfkeys\pgfqkeys{/itf/fancyhdr}{#1#3}\relax\ps@@fancy}%
   }%
}% \itf@p@gesetup
\pgfqkeys{/itf/pagesetup}{%
   @current/.is private=interfaces-fancyhdr,%
   @style/.is private=interfaces-fancyhdr,%
}
```

\pagestyle

```
\renewrobustcmd*\pagestyle[1]{\ifcsundef{ps@#1}
   {\itf@undefinedpagestyle{#1}}
   {\csname ps@#1\endcsname}%
}% \pagestyle
```

\thispagestyle

```
\renewrobustcmd*\thispagestyle[1]{\ifcsundef{ps@#1}
   {\itf@undefinedpagestyle{#1}}
   {\global\@specialpagetrue\gdef\@specialstyle{#1}}%
}% \thispagestyle
\newcommand*\itf@undefinedpagestyle[1]{%
   \@latex@error{Undefined page style: #1
   \MessageBreak your command will be ignored}\@eha
}% \itf@undefinedpagestyle
```

21.9.2 Keys for \pagesetup

```
\pgfkeys{%
   /handlers/.fancy@hf/.code=%
      \edef\itf@tp{\pgfkeyscurrentpath}%
      \pgfkeysalso{\pgfkeyscurrentpath/.store as code=}%
      \pgfkeysalso{\pgfkeyscurrentpath/.append code/.expanded=%
         \csdef{f@ncy#1}{%
            \noexpand\pgfkeysvalueof{\itf@tp/font}
            \noexpand\pgfkeysvalueof{\itf@tp}\noexpand\strut}},%
}% handler for fancyhdr interface
\pgfqkeys{/itf/fancyhdr}{% keys for \pagesetup
   code/.code={#1},% execution at expansion of \ps@... => \pgfqkeys{...,code={...},...}
   pagestyle/.style={code=\pagestyle{#1}},%
      page style/.alias=pagestyle,%
      load style/.alias=pagestyle,%
   head/rule/.store as dimen in cs=headrulewidth,%
   head/rule/.default=.4pt,%
   head/rule/.blank=\z@,%
      head/rulewidth/.alias*=head/rule,%
      headrulewidth/.alias*=head/rule,%
      head/rule/width/.alias*=head/rule,%
   head/rule/font/.store as code,%
   head/rule/color/.style={head/rule/font+=\color{#1}},%
   headruler/.code=%
      \let\headrule\pgfkeyscurrentvalue
      \ifx\headrule\@empty\else
         \preto\headrule{\bgroup\pgfkeysvalueof{/itf/fancyhdr/head/rule/font}}%
         \appto\headrule{\egroup}%
      \fi,%
      headruler/.default=%
         \if@fancyplain\let\headrulewidth\plainheadrulewidth\fi
         \hrule\@height\headrulewidth \@width\headwidth
         \vskip-\headrulewidth\relax,%
      headrule/.alias=headruler,%
      headrule,% sets default
   foot/rule/.store as dimen in cs=footrulewidth,%
      foot/rule/.default=.4pt,%
      foot/rule/.blank=\z@,%
      foot/rulewidth/.alias*=foot/rule,%
      footrulewidth/.alias*=foot/rule,%
      foot/rule/width/.alias*=foot/rule,%
   foot/rule/font/.store as code,%
   foot/rule/color/.style={foot/rule/font+=\color{#1}},%
   footruler/.code=%
      \let\footrule\pgfkeyscurrentvalue
      \ifx\footrule\@empty\else
         \preto\footrule{\bgroup\pgfkeysvalueof{/itf/fancyhdr/head/rule/font}}%
         \appto\footrule{\egroup}
      \fi,%
      footruler/.default=%
         \if@fancyplain\let\footrulewidth\plainfootrulewidth\fi
         \vskip-2\footrulewidth
         \hrule\@height\footrulewidth \@width\headwidth\vskip\footrulewidth,%
      footrule/.alias=footruler,%
      footrule,% sets default
   rule/.style={head/rule={#1},foot/rule={#1}},%
   rule+/.style={head/rule+={#1},foot/rule+={#1}},%
   rule-/.style={head/rule-={#1},foot/rule-={#1}},%
   rule*/.style={head/rule*={#1},foot/rule*={#1}},%
      rules/.alias*=rule,rulewidth/.alias*=rule,%
      rule/width/.alias*=rule,rules/width/.alias*=rule,%
   norule/.style={rule=\z@},%
   norule/.value forbidden,%
```

```
1131      norules/.alias=norule,%
1132   rule/font/.style={head/rule/font={#1},foot/rule/font={#1}},%
1133   rule/font+/.style={head/rule/font+={#1},foot/rule/font+={#1}},%
1134      rule/font*/.alias=rule/font+,%
1135      rules/font/.alias=rule/font,
1136      rules/font+/.alias=rule/font+,%
1137      rules/font*/.alias=rule/font+,%
1138   rule/color/.style={head/rule/color={#1},foot/rule/color={#1}},%
1139      rules/color/.alias=rule/color,%
1140   foot/rule/skip/.store as glue in cs=footruleskip,%
1141      foot/rule/skip/.default=.3\normalbaselineskip,%
1142      foot/rule/skip/.blank=\z@,%
1143      foot/ruleskip/.alias*=foot/rule/skip,%            => puis les alias&style: /.alias 2
1144      footruleskip/.alias*=foot/rule/skip,%                                        crée aussi
1145   font/.style={head/font={#1},foot/font={#1}},%
1146      font+/.style={head/font+={#1},foot/font+={#1}},%
1147      font*/.alias=font+,%
1148   color/.style={head/color={#1},foot/color={#1},head/rule/color={#1},foot/rule/color={#1}},%
1149   head/.style={head/left={#1},head/center={#1},head/right={#1}},%
1150      head/font/.style={head/left/font={#1},head/center/font={#1},head/right/font={#1}},%
1151      head/font+/.style={head/left/font+={#1},head/center/font+={#1},head/right/font+={#1}},%
1152      head/font*/.alias=head/font+,%
1153      head/color/.style={head/left/color={#1},head/center/color={#1},head/right/color={#1},he
1154   nohead/.style={head=,head/font=,head/rule=,head/rule/font=},
1155   foot/.style={foot/left={#1},foot/center={#1},foot/right={#1}},%
1156      foot/font/.style={foot/left/font={#1},foot/center/font={#1},foot/right/font={#1}},%
1157      foot/font+/.style={foot/left/font+={#1},foot/center/font+={#1},foot/right/font+={#1}},%
1158      foot/font*/.alias=foot/font+,%
1159      foot/color/.style={foot/left/color={#1},foot/center/color={#1},foot/right/color={#1},fo
1160   nofoot/.style={foot=,foot/font=,foot/rule=,foot/rule/font=},
1161   left/.style={head/left={#1},foot/left={#1}},%
1162      left/font/.style={head/left/font={#1},foot/left/font={#1}},%
1163      left/font+/.style={head/left/font+={#1},foot/left/font+={#1}},%
1164      left/font*/.alias=left/font+,%
1165      left/color/.style={head/left/color={#1},foot/left/color={#1}},%
1166   center/.style={head/center={#1},foot/center={#1}},%
1167      center/font/.style={head/center/font={#1},foot/center/font={#1}},%
1168      center/font+/.style={head/center/font+={#1},foot/center/font+={#1}},%
1169      center/font*/.alias=center/font+,%
1170      center/color/.style={head/center/color={#1},foot/center/color={#1}},%
1171   right/.style={head/right={#1},foot/right={#1}},%
1172      right/font/.style={head/right/font={#1},foot/right/font={#1}},%
1173      right/font+/.style={head/right/font+={#1},foot/right/font+={#1}},%
1174      right/font*/.alias=right/font+,%
1175      right/color/.style={head/right/color={#1},foot/right/color={#1}},%
1176   head/left/.style={head/left/even={#1},head/left/odd={#1}},%
1177      head/left/font/.style={head/left/even/font={#1},head/left/odd/font={#1}},%
1178      head/left/font+/.style={head/left/even/font+={#1},head/left/odd/font+={#1}},%
1179      head/left/font*/.alias=head/left/font+,%
1180      head/left/color/.style={head/left/even/color={#1},head/left/odd/color={#1}},%
1181   head/center/.style={head/center/even={#1},head/center/odd={#1}},%
1182      head/center/font/.style={head/center/even/font={#1},head/center/odd/font={#1}},%
1183      head/center/font+/.style={head/center/even/font+={#1},head/center/odd/font+={#1}},%
1184      head/center/font*/.alias=head/center/font+,%
1185      head/center/color/.style={head/center/even/color={#1},head/center/odd/color={#1}},%
1186   head/right/.style={head/right/even={#1},head/right/odd={#1}},%
1187      head/right/font/.style={head/right/even/font={#1},head/right/odd/font={#1}},%
1188      head/right/font+/.style={head/right/even/font+={#1},head/right/odd/font+={#1}},%
1189      head/right/font*/.alias=head/right/font+,%
1190      head/right/color/.style={head/right/even/color={#1},head/right/odd/color={#1}},%
1191   foot/left/.style={foot/left/even={#1},foot/left/odd={#1}},%
1192      foot/left/font/.style={foot/left/even/font={#1},foot/left/odd/font={#1}},%
1193      foot/left/font+/.style={foot/left/even/font+={#1},foot/left/odd/font+={#1}},%
1194      foot/left/font*/.alias=foot/left/font+,%
1195      foot/left/color/.style={foot/left/even/color={#1},foot/left/odd/color={#1}},%
```

```
1196  foot/center/.style={foot/center/even={#1},foot/center/odd={#1}},%
1197     foot/center/font/.style={foot/center/even/font={#1},foot/center/odd/font={#1}},%
1198     foot/center/font+/.style={foot/center/even/font+={#1},foot/center/odd/font+={#1}},%
1199     foot/center/font*/.alias=foot/center/font+,%
1200     foot/center/color/.style={foot/center/even/color={#1},foot/center/odd/color={#1}},%
1201  foot/right/.style={foot/right/even={#1},foot/right/odd={#1}},%
1202     foot/right/font/.style={foot/right/even/font={#1},foot/right/odd/font={#1}},%
1203     foot/right/font+/.style={foot/right/even/font+={#1},foot/right/odd/font+={#1}},%
1204     foot/right/font*/.alias=foot/right/font+,%
1205     foot/right/color/.style={foot/right/even/color={#1},foot/right/odd/color={#1}},%
1206  head/left/odd/.fancy@hf=olh,%
1207  head/left/odd/font/.store as code,%
1208  head/left/odd/color/.style={head/left/odd/font+=\color{#1}},%
1209        odd/head/left/.alias=head/left/odd,%
1210        odd/head/left/font/.alias*=head/left/odd/font,%
1211        odd/head/left/color/.alias=head/left/odd/color,%
1212        head/odd/left/.alias=head/left/odd,%
1213        head/odd/left/font/.alias*=head/left/odd/font,%
1214        head/odd/left/color/.alias=head/left/odd/color,%
1215  head/center/odd/.fancy@hf=och,%
1216  head/center/odd/font/.store as code,%
1217  head/center/odd/color/.style={head/center/odd/font+=\color{#1}},%
1218        odd/head/center/.alias=head/center/odd,%
1219        odd/head/center/font/.alias*=head/center/odd/font,%
1220        odd/head/center/color/.alias=head/center/odd/color,%
1221        head/odd/center/.alias=head/center/odd,%
1222        head/odd/center/font/.alias*=head/center/odd/font,%
1223        head/odd/center/color/.alias=head/center/odd/color,%
1224  head/right/odd/.fancy@hf=orh,%
1225  head/right/odd/font/.store as code,%
1226  head/right/odd/color/.style={head/right/odd/font+=\color{#1}},%
1227        odd/head/right/.alias=head/right/odd,%
1228        odd/head/right/font/.alias*=head/right/odd/font,%
1229        odd/head/right/color/.alias=head/right/odd/color,%
1230        head/odd/right/.alias=head/right/odd,%
1231        head/odd/right/font/.alias*=head/right/odd/font,%
1232        head/odd/right/color/.alias=head/right/odd/color,%
1233  head/left/even/.fancy@hf=elh,%
1234  head/left/even/font/.store as code,%
1235  head/left/even/color/.style={head/left/even/font+=\color{#1}},%
1236        even/head/left/.alias=head/left/even,%
1237        even/head/left/font/.alias*=head/left/even/font,%
1238        even/head/left/color/.alias=head/left/even/color,%
1239        head/even/left/.alias=head/left/even,%
1240        head/even/left/font/.alias*=head/left/even/font,%
1241        head/even/left/color/.alias=head/left/even/color,%
1242  head/center/even/.fancy@hf=ech,%
1243  head/center/even/font/.store as code,%
1244  head/center/even/color/.style={head/center/even/font+=\color{#1}},%
1245        even/head/center/.alias=head/center/even,%
1246        even/head/center/font/.alias*=head/center/even/font,%
1247        even/head/center/color/.alias=head/center/even/color,%
1248        head/even/center/.alias=head/center/even,%
1249        head/even/center/font/.alias*=head/center/even/font,%
1250        head/even/center/color/.alias=head/center/even/color,%
1251  head/right/even/.fancy@hf=erh,%
1252  head/right/even/font/.store as code,%
1253  head/right/even/color/.style={head/right/even/font+=\color{#1}},%
1254        even/head/right/.alias=head/right/even,%
1255        even/head/right/font/.alias*=head/right/even/font,%
1256        even/head/right/color/.alias=head/right/even/color,%
1257        head/even/right/.alias=head/right/even,%
1258        even/head/right/font/.alias*=head/right/even/font,%
1259        even/head/right/color/.alias=head/right/even/color,%
1260  foot/left/odd/.fancy@hf=olf,%
```

```
1261  foot/left/odd/font/.store as code,%
1262  foot/left/odd/color/.style={foot/left/odd/font+=\color{#1}},%
1263        odd/foot/left/.alias=foot/left/odd,%
1264        odd/foot/left/font/.alias*=foot/left/odd/font,%
1265        odd/foot/left/color/.alias=foot/left/odd/color,%
1266        foot/odd/left/.alias=foot/left/odd,%
1267        foot/odd/left/font/.alias*=foot/left/odd/font,%
1268        foot/odd/left/color/.alias=foot/left/odd/color,%
1269  foot/center/odd/.fancy@hf=ocf,%
1270  foot/center/odd/font/.store as code,%
1271  foot/center/odd/color/.style={foot/center/odd/font+=\color{#1}},%
1272        odd/foot/center/.alias=foot/center/odd,%
1273        odd/foot/center/font/.alias*=foot/center/odd/font,%
1274        odd/foot/center/color/.alias=foot/center/odd/color,%
1275        foot/odd/center/.alias=foot/center/odd,%
1276        foot/odd/center/font/.alias*=foot/center/odd/font,%
1277        foot/odd/center/color/.alias=foot/center/odd/color,%
1278  foot/right/odd/.fancy@hf=orf,%
1279  foot/right/odd/font/.store as code,%
1280  foot/right/odd/color/.style={foot/right/odd/font+=\color{#1}},%
1281        odd/foot/right/.alias=foot/right/odd,%
1282        odd/foot/right/font/.alias*=foot/right/odd/font,%
1283        odd/foot/right/color/.alias=foot/right/odd/color,%
1284        odd/foot/right/.alias=foot/right/odd,%
1285        odd/foot/right/font/.alias*=foot/right/odd/font,%
1286        odd/foot/right/color/.alias=foot/right/odd/color,%
1287  foot/left/even/.fancy@hf=elf,%
1288  foot/left/even/font/.store as code,%
1289  foot/left/even/color/.style={foot/left/even/font+=\color{#1}},%
1290        even/foot/left/.alias=foot/left/even,%
1291        even/foot/left/font/.alias*=foot/left/even/font,%
1292        even/foot/left/color/.alias=foot/left/even/color,%
1293        foot/even/left/.alias=foot/left/even,%
1294        foot/even/left/font/.alias*=foot/left/even/font,%
1295        foot/even/left/color/.alias=foot/left/even/color,%
1296  foot/center/even/.fancy@hf=ecf,%
1297  foot/center/even/font/.store as code,%
1298  foot/center/even/color/.style={foot/center/even/font+=\color{#1}},%
1299        even/foot/center/.alias=foot/center/even,%
1300        even/foot/center/font/.alias*=foot/center/even/font,%
1301        even/foot/center/color/.alias=foot/center/even/color,%
1302        foot/even/center/.alias=foot/center/even,%
1303        foot/even/center/font/.alias*=foot/center/even/font,%
1304        foot/even/center/color/.alias=foot/center/even/color,%
1305  foot/right/even/.fancy@hf=erf,%
1306  foot/right/even/font/.store as code,%
1307  foot/right/even/color/.style={foot/right/even/font+=\color{#1}},%
1308        even/foot/right/.alias=foot/right/even,%
1309        even/foot/right/font/.alias*=foot/right/even/font,%
1310        even/foot/right/color/.alias=foot/right/even/color,%
1311        foot/even/right/.alias=foot/right/even,%
1312        foot/even/right/font/.alias*=foot/right/even/font,%
1313        foot/even/right/color/.alias=foot/right/even/color,%
1314  offset/.style={head/offset={#1},foot/offset={#1}},%
1315     offset+/.style={head/offset+={#1},foot/offset+={#1}},%
1316     offset-/.style={head/offset-={#1},foot/offset-={#1}},%
1317     offset*/.style={head/offset*={#1},foot/offset*={#1}},%
1318  head/offset/.style={head/left/offset={#1},head/right/offset={#1}},%
1319     head/offset+/.style={head/left/offset+={#1},head/right/offset+={#1}},%
1320     head/offset-/.style={head/left/offset-={#1},head/right/offset-={#1}},%
1321     head/offset*/.style={head/left/offset*={#1},head/right/offset*={#1}},%
1322        offset/head/.alias*=head/offset,%
1323  foot/offset/.style={foot/left/offset={#1},foot/right/offset={#1}},%
1324     foot/offset+/.style={foot/left/offset+={#1},foot/right/offset+={#1}},%
1325     foot/offset-/.style={foot/left/offset-={#1},foot/right/offset-={#1}},%
```

```
1326      foot/offset*/.style={foot/left/offset*={#1},foot/right/offset*={#1}},%
1327         offset/foot/.alias*=foot/offset,%
1328   left/offset/.style={foot/left/offset={#1},head/left/offset={#1}},%
1329      left/offset+/.style={foot/left/offset+={#1},head/left/offset+={#1}},%
1330      left/offset-/.style={foot/left/offset-={#1},head/left/offset-={#1}},%
1331      left/offset*/.style={foot/left/offset*={#1},head/left/offset*={#1}},%
1332         offset/left/.alias*=left/offset,%
1333   right/offset/.style={foot/right/offset={#1},head/right/offset={#1}},%
1334         right/offset+/.style={foot/right/offset+={#1},head/right/offset+={#1}},%
1335         right/offset-/.style={foot/right/offset-={#1},head/right/offset-={#1}},%
1336         right/offset*/.style={foot/right/offset*={#1},head/right/offset*={#1}},%
1337            offset/right/.alias*=right/offset,%
1338   head/left/offset/.style={head/left/even/offset={#1},head/left/odd/offset={#1}},%
1339      head/left/offset+/.style={head/left/even/offset+={#1},head/left/odd/offset+={#1}},%
1340      head/left/offset-/.style={head/left/even/offset-={#1},head/left/odd/offset-={#1}},%
1341      head/left/offset*/.style={head/left/even/offset*={#1},head/left/odd/offset*={#1}},%
1342         offset/head/left/.alias*=head/left/offset,%
1343         offset/left/head/.alias*=head/left/offset,%
1344         left/head/offset/.alias*=head/left/offset,%
1345   head/right/offset/.style={head/right/even/offset={#1},head/right/odd/offset={#1}},%
1346      head/right/offset+/.style={head/right/even/offset+={#1},head/right/odd/offset+={#1}},%
1347      head/right/offset-/.style={head/right/even/offset-={#1},head/right/odd/offset-={#1}},%
1348      head/right/offset*/.style={head/right/even/offset*={#1},head/right/odd/offset*={#1}},%
1349         offset/head/right/.alias*=head/right/offset,%
1350         offset/right/head/.alias*=head/right/offset,%
1351         right/head/offset/.alias*=head/right/offset,%
1352   foot/left/offset/.style={foot/left/even/offset={#1},foot/left/odd/offset={#1}},%
1353      foot/left/offset+/.style={foot/left/even/offset+={#1},foot/left/odd/offset+={#1}},%
1354      foot/left/offset-/.style={foot/left/even/offset-={#1},foot/left/odd/offset-={#1}},%
1355      foot/left/offset*/.style={foot/left/even/offset*={#1},foot/left/odd/offset*={#1}},%
1356         offset/foot/left/.alias*=foot/left/offset,%
1357         offset/left/foot/.alias*=foot/left/offset,%
1358         left/foot/offset/.alias*=foot/left/offset,%
1359   foot/right/offset/.style={foot/right/even/offset={#1},foot/right/odd/offset={#1}},%
1360      foot/right/offset+/.style={foot/right/even/offset+={#1},foot/right/odd/offset+={#1}},%
1361      foot/right/offset-/.style={foot/right/even/offset-={#1},foot/right/odd/offset-={#1}},%
1362      foot/right/offset*/.style={foot/right/even/offset*={#1},foot/right/odd/offset*={#1}},%
1363         offset/foot/right/.alias*=foot/right/offset,%
1364         offset/right/foot/.alias*=foot/right/offset,%
1365         right/foot/offset/.alias*=foot/right/offset,%
1366   head/left/even/offset/.store in skip cs=f@ncyO@elh,%
1367      head/left/even/offset/.append code=\fancy@setoffs,%
1368         offset/head/left/even/.alias*=head/left/even/offset,%
1369         left/even/head/offset/.alias*=head/left/even/offset,%
1370   head/left/odd/offset/.store in skip cs=f@ncyO@olh,%
1371      head/left/odd/offset/.append code=\fancy@setoffs,%
1372         offset/head/left/odd/.alias*=head/left/odd/offset,%
1373         offset/head/left/odd/.alias*=head/left/odd/offset,%
1374         left/odd/head/offset/.alias*=head/left/odd/offset,%
1375   head/right/even/offset/.store in skip cs=f@ncyO@erh,%
1376      head/right/even/offset/.append code=\fancy@setoffs,%
1377         offset/head/right/even/.alias*=head/right/even/offset,%
1378         right/even/head/offset/.alias*=head/right/even/offset,%
1379   head/right/odd/offset/.store in skip cs=f@ncyO@orh,%
1380      head/right/odd/offset/.append code=\fancy@setoffs,%
1381         offset/head/right/odd/.alias*=head/right/odd/offset,%
1382         right/odd/head/offset/.alias*=head/right/odd/offset,%
1383   foot/left/even/offset/.store in skip cs=f@ncyO@elf,%
1384      foot/left/even/offset/.append code=\fancy@setoffs,%
1385         offset/foot/left/even/.alias*=foot/left/even/offset,%
1386         left/even/foot/offset/.alias*=foot/left/even/offset,%
1387   foot/left/odd/offset/.store in skip cs=f@ncyO@olf,%
1388      foot/left/odd/offset/.append code=\fancy@setoffs,%
1389         offset/foot/left/odd/.alias*=foot/left/odd/offset,%
1390         left/odd/foot/offset/.alias*=foot/left/odd/offset,%
```

```
1391    foot/right/even/offset/.store in skip cs=f@ncyO@erf,%
1392       foot/right/even/offset/.append code=\fancy@setoffs,%
1393          offset/foot/right/even/.alias*=foot/right/even/offset,%
1394          right/even/foot/offset/.alias*=foot/right/even/offset,%
1395    foot/right/odd/offset/.store in skip cs=f@ncyO@orf,%
1396       foot/right/odd/offset/.append code=\fancy@setoffs,%
1397          offset/foot/right/odd/.alias*=foot/right/odd/offset,%
1398          right/odd/foot/offset/.alias*=foot/right/odd/offset,%
1399    reset/.style={norule,head=,foot=,offset=\z@,foot/rule/skip,rule/font=,font=},%
1400       reset/.value forbidden,%
1401 }% keys for \pagesetup (stored into token \itf@tk)
1402 \AtBeginDocument{%
1403 \ifdim\headwidth<\z@
1404 %% This catches the case that \headwidth hasn't been initialized and the
1405 %% case that the user added something to \headwidth in the expectation that
1406 %% it was initialized to \textwidth. We compensate this now. This loses if
1407 %% the user intended to multiply it by a factor. But that case is more
1408 %% likely done by saying something like \headwidth=1.2\textwidth.
1409 %% The doc says you have to change \headwidth after the first call to
1410 %% \pagestyle{fancy}. This code is just to catch the most common cases were
1411 %% that requirement is violated.
1412     \global\advance\headwidth123456789sp\global\advance\headwidth\textwidth
1413 \fi}
1414 ⟨/fancyhdr⟩
```

21.10 † marks

```
1415 ⟨*marks⟩
1416 \ProvidesPackage{interfaces-marks}
1417    [2010/10/27 v1.2 easier use of e-TeX marks (FC)]
1418 \RequirePackage{interfaces-base}
```

\marksthe

```
1419 \newrobustcmd\marksthe[2]{%
1420    \ifcsname marks@#1\endcsname\else
1421       \itf@marks@new{#1}%
1422    \fi
1423    \expandafter\marks\csname marks@#1\endcsname{\unexpanded{#2}}%
1424 }% \marksthe
```

\marksthecs

```
1425 \newrobustcmd*\marksthecs[2]{%
1426    \ifcsname marks@#1\endcsname\else
1427       \itf@marks@new{#1}%
1428    \fi
1429    \ifcsundef{#2}
1430       {}
1431       {\expandafter\marks\csname marks@#1\endcsname{%
1432          \unexpanded\expandafter\expandafter\expandafter{%
1433                \csname#2\endcsname}}}%
1434 }% \marksthecs
```

\itf@marks@new

```
1435 \def\itf@marks@new#1{%
1436    \expandafter\newmarks\csname marks@#1\endcsname
1437    \PackageInfo{interfaces-marks}
1438       {New marks `#1' defined}
1439 }% \itf@marks@warning
```

\thefirstmarks
\thebotmarks
\thetopmarks

```
\newcommand*\thefirstmarks[1]{%
   \ifcsname marks@#1\endcsname
      \expandafter\firstmarks\csname marks@#1\endcsname
   \fi
}% \thefirstmarks
\newcommand*\thebotmarks[1]{%
   \ifcsname marks@#1\endcsname
      \expandafter\botmarks\csname marks@#1\endcsname
   \fi
}% \thebotmarks
\newcommand*\thetopmarks[1]{%
   \ifcsname marks@#1\endcsname
      \expandafter\topmarks\csname marks@#1\endcsname
   \fi
}% \thetopmarks
```

\getthemarks

```
%% \getthemarks\firstmarks{section}\macro
\newrobustcmd*\getthemarks[3]{%
   \ifcsname marks@#2\endcsname
      \expandafter\itf@getthemarks\expandafter#1\csname marks@#2\endcsname{#3}%
   \else \let#3\@undefined
   \fi
}% \getthemarks
\def\itf@getthemarks#1#2#3{%
   \expandafter\def\expandafter#3\expandafter{#1#2}%
}% \itf@getthemarks
```

\getthefirstmarks
\getthebotmarks
\getthetopmarks

```
\newrobustcmd*\getthefirstmarks[2]{\getthemarks\firstmarks{#1}{#2}}
\newrobustcmd*\getthebotmarks[2]{\getthemarks\botmarks{#1}{#2}}
\newrobustcmd*\getthetopmarks[2]{\getthemarks\topmarks{#1}{#2}}
```

\ifmarksequal

```
%% \ifmarksequal\firstmarks\botmarks{subsection}{ true }{ false }
\newrobustcmd*\ifmarksequal[3]{%
   \begingroup
      \getthemarks{#1}{#3}\@tempa
      \getthemarks{#2}{#3}\@tempb
   \expandafter\endgroup
   \csname ltx@\ifdefined\@tempa\ifdefined\@tempb
      \ifx\@tempa\@tempb first\else
      second\fi\else
      second\fi\else
      second\fi
   oftwo\endcsname
}% \ifmarksequal
```

⟨/marks⟩

21.11 Interface tocloft : \tocsetup, \lofsetup and \lotsetup

```
⟨*tocloft⟩
\ProvidesPackage{interfaces-tocloft}
   [2010/10/27 v1.2 pgfkeys macros for tocloft (FC)]
\RequirePackage{interfaces-base}
\itf@Check@Package{tocloft}
\itf@pgfkeys
```

\tocsetup
\lofsetup
\lotsetup
\listofsetup

The user commands.

```
\newrobustcmd\tocsetup[1]{%
   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/toc}{#1}}%
}% \tocsetup
\newrobustcmd\lofsetup[1]{%
   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/lof}{#1}}%
}% \lofsetup
\newrobustcmd\lotsetup[1]{%
   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/lot}{#1}}%
}% \lotsetup
\newrobustcmd\listofsetup[2]{%
   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/#1}{#2}}%
}% \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.

```
\def\itf@cft@newlistof#1{%
   \pgfqkeys{/itf/tocloft/#1}{%
      pagestyle/.code=\tocloftpagestyle{##1},%
      parskip/.store in skip cs=cftparskip,%
         parsep/.alias=parskip,%
      mark/.store in cs=cftmark#1,%
         marks/.alias=mark,%
      label/.store as code,%
      ref/.store as code,%
         ref/.initial=\csname cft#1title\endcsname,% overwritten (toc,lof,lot)
         ref/.default=\csname cft#1title\endcsname,% overwritten (toc,lof,lot)
         ref/.blank=\pgfkeysnovalue,%
      title/.store as code in cs=cft#1title,% overwritten below for toc, lof and lot
      title/top/.store in skip cs=cftbefore#1titleskip,%
         beforetitleskip/.alias*=title/top,%
      title/bottom/.store in skip cs=cftafter#1titleskip,%
         aftertitleskip/.alias*=title/bottom,%
      title/font/.store as code in cs=cft#1titlefont,%
         title/format/.alias*=title/font,%
      title/after/.store as code in cs=cftafter#1title,%
      title/command/.code=\csdef{@cftmake#1title}####1{##1},%
      dotsep/.code=\itf@units@mu\itf@musk{##1}%    converts to mu
                   \edef\cftdotsep{\itf@strip@mu\itf@musk},%
         dotsep/.default=4.5mu,%
         dotsep/.blank=\cftnodots,%
      dot/.store in=\cftdot,%
      before/.store as code,%
      after/.store as code,%
      columns/before@hook/.is private=interfaces-tocloft,%
      columns/after@hook/.is private=interfaces-tocloft,%
      columns/before@hook/.initial=,%
      columns/after@hook/.initial=,%
      columns/.code={\ltx@ifpackageloaded{multicol}{%
         \pgfkeysEsetvalue\pgfkeyscurrentkey{\number##1}%
         \ifnum\number##1>\@ne
            \pgfkeysgetvalue{/itf/tocloft/#1/columns/before@hook}\itf@tp
            \ifx\itf@tp\@empty
               \pgfkeyssetvalue{/itf/tocloft/#1/columns/before@hook}{%
```

```
                \begingroup
                \multicolsep\pgfkeysvalueof{/itf/tocloft/#1/multicolsep}\relax
                \begin{multicols}{\pgfkeysvalueof{/itf/tocloft/#1/columns}}%
                \columnseprule\pgfkeysvalueof{/itf/tocloft/#1/columns/rule}\relax
                \columnsep\pgfkeysvalueof{/itf/tocloft/#1/columns/sep}\relax
                \pgfkeysgetvalue{/itf/tocloft/#1/columns/rule/color}\columnseprulecolor
                \pgfkeysvalueof{/itf/tocloft/#1/multicols/begin}%
             }%
             \pgfkeyssetvalue{/itf/tocloft/#1/columns/after@hook}{%
                \pgfkeysvalueof{/itf/tocloft/#1/multicols/end}%
                \end{multicols}
                \endgroup
             }%
          \fi
       \else
          \pgfkeyslet\pgfkeyscurrentkey\@ne
          \pgfkeyslet{/itf/tocloft/#1/columns/before@hook}\@empty
          \pgfkeyslet{/itf/tocloft/#1/columns/after@hook}\@empty
       \fi}
       {\PackageError{interfaces-tocloft}
          {Key 'columns': package multicol is required}\@eha}%
    },%
    columns/.default=2,columns/.blank=1,%
       multicol/.alias=columns,%
       multicols/.alias=columns,%
    twocolumns/.true=\pgfkeysalso{columns=2},%
    twocolumns/.false=\pgfkeysalso{columns=1},%
    columns/rule/.store as dimen,%
    columns/rule/.default=\columnseprule,%
    columns/rule,% assign default
       columnseprule/.alias*=columns/rule,%
       columns/rule/width/.alias*=columns/rule,%
    columns/sep/.store as dimen,%
    columns/sep/.default=\columnsep,%
    columns/sep/.blank=10\p@,%
    columns/sep,% assign default
       columnsep/.alias*=columns/sep,%
    columns/rule/color/.store as color command,%
    columns/rule/color,% assign default (\normalcolor)
       columnseprulecolor/.alias*=columns/rule/color,%
       columns/rulecolor/.alias*=columns/rule/color,%
    multicolsep/.store as glue,%
       multicolsep/.default=0pt plus4pt minus3pt,%
       multicolsep,% sets default
    multicols/begin/.initial=,multicols/begin/.default=,%
       multicols/before/.alias=multicols/begin,%
    multicols/end/.initial=,multicols/end/.default=,%
       multicols/after/.alias=multicols/end,%
}% \pgfkeys
\itf@AfterPackage{interfaces-bookmark}{%
\pgfqkeys{/itf/tocloft/#1}{%
   bookmark/.is multiple choice,%
      bookmark/bold/.true=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{,bold=true},%
      bookmark/bold/.false=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{,bold=false},%
      bookmark/italic/.true=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{,italic=true},%
      bookmark/italic/.false=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{,italic=false},%
      bookmark/color/.code=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{,color={##1}},%
      bookmark/level/.code=\pgfkeysEaddvalue{/itf/tocloft/#1/bookmark}{}{level={%
                     \ifcsname toclevel@##1\endcsname
                        \csname toclevel@##1\endcsname
                     \else##1\fi}},%
      bookmark/text/.store as code,%
         bookmark/text/.default=\contentsname,%
      bookmark/.unknown/.code=\pgfkeysEaddvalue{/itf/tocloft/#1/bookmark}{}{\pgfkeyscurren
   bookmark/.initial=,%
```

```
   }% \pgfkeys
   }% AfterPackage interfaces-bookmark
   %% while making a format \jobname is expanded to the name of the format... Therefore, the
   \AtEndPreamble{%
      \BeforeFile{\jobname.#1}{%
         \pgfkeysvalueof{/itf/tocloft/#1/before}%
         \pgfkeysvalueof{/itf/tocloft/#1/columns/before@hook}%
      }%
      \AfterFile{\jobname.#1}{%
         \pgfkeysvalueof{/itf/tocloft/#1/columns/after@hook}%
         \pgfkeysvalueof{/itf/tocloft/#1/after}%
      }%
   }%
   \ifcsname #1setup\endcsname\else
      \pgfqkeys{/itf/tocloft/#1}{%
         pagenumbers/.true=\@cftpnumon{#1},%
         pagenumbers/.false=\@cftpnumoff{#1},%
           pagenumber/.alias=pagenumbers
      }%
      \expandafter\pretocmd\csname @cftmake#1title\expandafter\endcsname
         \expandafter{\expandafter\itf@tocloft\csname cft#1title\endcsname{#1}{listof#1}}
      {\PackageInfo{interfaces-tocloft}
         {Command \expandafter\string\csname @cftmake#1title\endcsname
            successfully augmented}}
      {\PackageWarning{interfaces-tocloft}
         {Command \expandafter\string\csname @cftmake#1title\endcsname
            \space could not be patched
         \MessageBreak Keys label and bookmark won't work for \string\listofsetup}}%
   \fi
}% \itf@cft@newlistof
```

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

```
\itf@cft@newlistof{toc}
\itf@cft@newlistof{lof}
\itf@cft@newlistof{lot}

%% Specific to toc only ------------------------------------------------------
\pgfqkeys{/itf/tocloft/toc}{%
   depth/.store in counter cs=c@tocdepth,%
      depth/.append code=
         \global\c@tocdepth\c@tocdepth
         \addtocontents{toc}{\c@tocdepth\the\c@tocdepth\relax},%
      tocdepth/.alias=depth,%
   title/.store as code in cs=contentsname,%
      contentsname/.alias=title,%
   ref/.initial=\contentsname,%
      ref/.default=\contentsname,%
   pagenumbers/.style={% boolean key
      part/pagenumbers={#1},chapter/pagenumbers={#1},section/pagenumbers={#1},%
      subsubsection/pagenumbers={#1},subsubsection/pagenumbers={#1},%
      paragraph/pagenumbers={#1},subparagraph/pagenumbers={#1}},%
   pagenumber/.alias=pagenumbers,%
}%
%% Specific to lof and lot ---------------------------------------------------
\pgfqkeys{/itf/tocloft/lof}{%
   title/.store as code in cs=listfigurename,%
      listfigurename/.alias=title,%
   ref/.initial=\listfigurename,%
      ref/.default=\listfigurename,%
}%
\pgfqkeys{/itf/tocloft/lot}{%
   title/.store as code in cs=listtablename,%
      listtablename/.alias=title,%
   ref/.initial=\listtablename,%
```

```
      ref/.default=\listtablename,%
}%
\if@cftsubfigopt
\pgfqkeys{/itf/tocloft}{%
   lof/pagenumbers/.style={figure/pagenumbers={#1},subfigure/pagenumbers={#1}},%
      lof/pagenumber/.alias=lof/pagenumbers,%
   lof/depth/.store in counter cs=c@lofdepth,%
      lof/depth/.append code=\global\c@lofdepth\c@lofdepth\addtocontents{lof}{\c@lofdepth\the
   lot/pagenumbers/.style={figure/pagenumbers={#1},subfigure/pagenumbers={#1}},%
      lot/pagenumber/.alias=lot/pagenumbers,%
   lot/depth/.store in counter cs=c@lotdepth,%
      lot/depth/.append code=\global\c@lotdepth\c@lotdepth\addtocontents{lot}{\c@lotdepth\the
}%
\else
\pgfqkeys{/itf/tocloft}{%
   lof/pagenumbers/.style={figure/pagenumbers={#1}},%
      lof/pagenumber/.alias=lof/pagenumbers,%
   lot/pagenumbers/.style={figure/pagenumbers={#1}},%
      lot/pagenumber/.alias=lot/pagenumbers,%
}%
\fi
```

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

And the keys are either subsec or subsection.

```
\def\itf@cft@name#1.#2.#3\@nil{cft#1\itf@cft@secname{#2}#3}
\def\itf@cft@secname#1{%
   \expandafter\itf@cft@sec@name\csname#1\endcsname{#1}%
}% \itf@cft@secname
\def\itf@cft@sec@name#1#2{%
   \ifx#1\chapter chap\else
      \ifx#1\section sec\else
         \ifx#1\subsection subsec\else
            \ifx#1\subsubsection subsubsec\else
               \ifx#1\paragraph para\else
                  \ifx#1\subparagraph subpara\else
                     \ifx#1\figure fig\else
                        \ifx#1\subsubfigure subfig\else
                           \ifx#1\table tab\else
                              \ifx#1\subtable subtab\else
                               #2\fi
   \fi\fi\fi\fi\fi\fi\fi\fi\fi
}% \itf@cft@section@name
```

21.11.1 Specific handlers for tocloft

```
%% tocloft setup for tables sections :
\pgfkeys{%
   /handlers/.cft@def/.style={\pgfkeyscurrentpath/.store in cs=\itf@cft@name #1\@nil},%
   /handlers/.cft@code/.style={\pgfkeyscurrentpath/.store as code in cs=\itf@cft@name #1\@nil
   /handlers/.cft@length/.style={\pgfkeyscurrentpath/.store in skip cs=\itf@cft@name #1\@nil}
}
```

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

```
\def\itf@cft@newsection#1#2{% #1=part / chap / sec / subsec etc., #2 = #1 = toc | lof | lot
   \pgfqkeys{/itf/tocloft/#2/#1}{%
      skip/.cft@length=before.#1.skip,%
         beforeskip/.alias*=skip,%
         top/.alias*=skip,%
      indent/.cft@length=.#1.indent,%
      font/.cft@code=.#1.font,%
```

```
      font/.append code={\pgfkeysalso{/itf/tocloft/#2/#1/pagenumber/font={##1}}},%
         format/.alias*=font,%
      number/width/.cft@length=.#1.numwidth,%
         numwidth/.alias*=number/width,%
      presnum/.cft@code=.#1.presnum,%
         number/before/.alias*=presnum,%
      aftersnum/.cft@code=.#1.aftersnum,%
         number/after/.alias*=aftersnum,%
      aftersnumb/.cft@code=.#1.aftersnumb,%
         title/before/.alias*=aftersnumb,%
      leader/.cft@def=.#1.leader,%
         leader/.default=\cftdotfill{\csname\itf@cft@name.#1.dotsep\@nil\endcsname},%
         leaders/.alias=leader,%
      dotsep/.code={\ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \csdef{\itf@cft@name.#1.dotsep\@nil}{\cftdotsep}%
         \else
            \itf@units@mu\itf@musk{##1}%
            \csedef{\itf@cft@name.#1.dotsep\@nil}{\itf@strip@mu\itf@musk}}
         \fi,%
         dotsep/.blank=\cftnodots,%
      nodots/.code=\csdef{\itf@cft@name.#1.dotsep\@nil}{\cftnodots},%
         nodots/.value forbidden,%
      pagefont/.cft@code=.#1.pagefont,%
         page/font/.alias*=pagefont,%
         pagenumber/font/.alias*=pagefont,%
      afterpnum/.cft@code=.#1.afterpnum,%
         page/after/.alias*=afterpnum,%
         pagenumber/after/.alias*=afterpnum,%
      pagenumbers/.on=\@cftpnumon{\itf@cft@secname{#1}},%
      pagenumbers/.off=\@cftpnumoff{\itf@cft@secname{#1}},%
         pagenumber/.alias=pagenumbers,%
   }% \pgfqkeys
}% \itf@cft@newsection

\@for\itf@tp:=part,chap,chapter,sec,section,subsec,subsection,%
                  subsubsec,subsubsection,para,paragraph,subpara,subparagraph%
   \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{toc}}%
\if@cftsubfigopt % \@ifpackagewith{tocloft}{subfig} ??
   \@for\itf@tp:=fig,figure,subfig,subfigure%
      \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lof}}%
   \@for\itf@tp:=tab,table,subtab,subtable%
      \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lot}}%
\else
   \@for\itf@tp:=fig,figure%
      \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lof}}%
   \@for\itf@tp:=tab,table%
      \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lot}}%
\fi
```

@cftmaketoctitle (patch)
@cftmakeloftitle (patch)
@cftmakelottitle (patch)
newlistentry (patch)

```
\itf@patch\pretocmd\@cftmaketoctitle
   {\itf@tocloft\contentsname{toc}{tableofcontents}}
   {\PackageInfo{interfaces-tocloft}
      {Command \string\@cftmaketoctitle\space successfully augmented}}
   {\PackageWarning{interfaces-tocloft}
      {Command \string\@cftmaketoctitle\space could not be patched
      \MessageBreak label and bookmark keys for \string\tocsetup\space won't work}}%
\itf@patch\pretocmd\@cftmakeloftitle
   {\itf@tocloft\listfigurename{lof}{listoffigures}}
   {\PackageInfo{interfaces-tocloft}
      {Command \string\@cftmakeloftitle\space successfully augmented}}
   {\PackageWarning{interfaces-tocloft}
      {Command \string\@cftmakeloftitle\space could not be patched
```

```
      \MessageBreak label and bookmark keys for \string\lofsetup\space won't work}}%
\itf@patch\pretocmd\@cftmakelottitle
   {\itf@tocloft\listtablename{lot}{listoftables}}
   {\PackageInfo{interfaces-tocloft}
      {Command \string\@cftmakelottitle\space successfully augmented}}
   {\PackageWarning{interfaces-tocloft}
      {Command \string\@cftmakelottitle\space could not be patched
      \MessageBreak label and bookmark keys for \string\lotsetup\space won't work}}%
\itf@patch\apptocmd\newlistentry
   {\itf@cft@newlistof{#2}}
   {\PackageInfo{interfaces-tocloft}
      {Command \string\newlistentry\space successfully augmented}}
   {\PackageWarning{interfaces-tocloft}
      {Command \string\newlistentry\space could not be patched
      \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.

```
\def\itf@tocloft#1#2#3{%
   \expandafter\def\expandafter#1\expandafter{%
      \expandafter\itf@cft@label
            \expandafter#1\expandafter{#1}{#2}{#3}}%
}% \itf@tocloft
\long\def\itf@cft@label#1#2#3#4{%
   \def#1{#2}% restore original
   \begingroup
   \pgfkeysifdefined{/itf/tocloft/#3/bookmark}
      {%
      \raisedhyperdef{interfaces}{#4}{#2}%
      \pgfkeysgetvalue{/itf/tocloft/#3/bookmark/text}\itf@tp
      \ifx\itf@tp\@empty\else
         \pgfkeysgetvalue{/itf/tocloft/#3/bookmark}\@tempa
         \expandafter\bookmark\expandafter[%
            \expandafter{\@tempa,dest=interfaces.#4}]%
            {\pgfkeysvalueof{/itf/tocloft/#3/bookmark/text}}%
         \global\BKM@startatroottrue
      \fi}
      {#2}%
   \pgfkeysgetvalue{/itf/tocloft/#3/label}\itf@tp
   \ifx\itf@tp\@empty\else
      \def\@currentHref{interfaces.#4}%
      \pgfkeysgetvalue{/itf/tocloft/#3/ref}\@currentlabel
      \def\@currentlabelname{#2}%
      \label{\pgfkeysvalueof{/itf/tocloft/#3/label}}%
   \fi
   \endgroup
   \gdef\itf@tp{\def#1{#2}\let\itf@tp\@undefined}% restore original
   \aftergroup\itf@tp % simple group introduced by tocloft to print \contentsname
}% \itf@cft@label
```

1822 ⟨/tocloft⟩

21.12 Interface makecell : \repeatcell

```
1823 ⟨*makecell⟩
1824 \ProvidesPackage{interfaces-makecell}
1825    [2010/10/27 v1.2 more control on tabular cells (FC)]
1826 \RequirePackage{interfaces-base}
1827 \itf@Check@Package{makecell}
1828 \edef\itf@AtEnd{\catcode126 \the\catcode126}% 126 = ~
1829 \AtEndOfPackage{\itf@AtEnd}
1830 \catcode126 \active % 126 = ~
1831 \itf@pgfkeys
```

21.12.1 countvalues

\countvalues

```
1832 \def\countvalues#1{\number\numexpr
1833    \expandafter\itf@countvalues\expandafter0#1,\countvalues\@nnil
1834 }% \countvalues
1835 \def\itf@countvalues#1#2,#3\@nnil{%
1836    \ifx\countvalues#3%
1837       +#1\expandafter\remove@to@nnil
1838    \else
1839       +1\expandafter\itf@countvalues
1840    \fi1#3\@nnil
1841 }% \itf@countvalues
```

\itf@getvalue

```
1842 \def\itf@getvalue#1#2{%
1843    \romannumeral-`\q\expandafter\itf@getv@lue\expandafter0#1,\itf@getvalue\@nil{#2-1}\@nnil
1844 }% \itf@getvalue
1845 \def\itf@getv@lue#1#2,#3\@nil#4\@nnil{%
1846    \ifnum\numexpr#1=\numexpr#4\relax
1847       #2\expandafter\remove@to@nnil
1848    \else\ifx\itf@getvalue#3%
1849       \expandafter\expandafter\expandafter\remove@to@nnil
1850    \else
1851       \expandafter\expandafter\expandafter\itf@getv@lue
1852    \fi\fi{#1+1}#3\@nil#4\@nnil
1853 }% \itf@getv@lue
```

21.12.2 \repeatcell

\repeatcell

```
1854 \newif\ifitf@celltranspose
1855 \def\itf@ampactive{%
1856    \def\tab{&}%
1857    \lccode`\~=38\lowercase{\def~{&}}%
1858 }% \itf@ampactive
1859 \newrobustcmd*\repeatcell{\begingroup
1860    \let\tab\relax
1861    \catcode38\active
1862    \itf@repeatcell
1863 }% \repeatcell
1864 \newrobustcmd*\itf@repeatcell[3][]{%
1865    \ifblank{#1}% alignment option ?
1866       {\let\itf@makecell\@firstofone}
1867       {\def\itf@makecell##1{\noexpand\makecell[{#1}]{##1}}}%
1868    \countdef\itf@col=100% local
1869    \countdef\itf@row=101% local
1870    \countdef\itf@cnt=102% local
```

```
\toksdef\itf@cell=1% local
\itf@pgfkeys
\pgfqkeys{/itf/makecell/repeatcell}{@reset={#2},#3}%
\toks@{}% local
\ifnum\itf@row>\z@ % number
   \edef\itf@rowloop{\noexpand\@whilenum\itf@row<\the\itf@row}%
   \def\itf@cell@init{\def\r{\row}\let\R\@empty}%
\else % list
   \def\itf@rowloop{\@for\r:=\itf@rows}%
   \expandafter\def\expandafter\itf@cell@init\expandafter{%
      \expandafter\def\expandafter\R\expandafter{%
         \expandafter\itf@getvalue\expandafter{\expandafter\@empty\itf@rows}}}%
\fi
\ifnum\itf@col>\z@ % number
   \edef\itf@colloop{\noexpand\@whilenum\itf@col<\the\itf@col}%
   \expandafter\def\expandafter\itf@cell@init\expandafter{%
      \itf@cell@init\def\c{\column}\let\C\@empty}%
\else % list
   \def\itf@colloop{\@for\c:=\itf@cols}%
   \expandafter\expandafter\expandafter\def
      \expandafter\expandafter\expandafter\itf@cell@init
         \expandafter\expandafter\expandafter{%
            \expandafter\itf@cell@init
            \expandafter\def\expandafter\C\expandafter{%
                  \expandafter\itf@getvalue\expandafter{\expandafter\@empty\itf@cols}}}%
\fi
\itf@cell@init % in case they are used in formulas
\ifitf@celltranspose
   \itf@col\z@
   \itf@colloop\do{%
      \ifnum\itf@col>\z@
      \pgfkeysifdefined{/itf/makecell/repeatcell/cr/row\the\itf@col}%
         {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr/row\the\itf@col}\itf@thecr}
         {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr}\itf@thecr}%
      \toks@\expandafter\expandafter\expandafter{%
                  \expandafter\the\expandafter\toks@\itf@thecr}%
      \fi
      \advance\itf@col\@ne
      \itf@row\z@
      \itf@rowloop\do{%
         \advance\itf@row\@ne
         \ifnum\itf@row>\@ne
            \toks@\expandafter{\the\toks@&}%
         \fi
         \toks@\expandafter\expandafter\expandafter{%
            \expandafter\the\expandafter\toks@
            \itf@cell@init\itf@ampactive}%
         \itf@cell@insert}%
   }%
\else
\itf@row\z@
\itf@rowloop\do{%
   \ifnum\itf@row>\z@
      \pgfkeysifdefined{/itf/makecell/repeatcell/cr/row\the\itf@row}%
         {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr/row\the\itf@row}\itf@thecr}
         {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr}\itf@thecr}%
      \toks@\expandafter\expandafter\expandafter{%
                \expandafter\the\expandafter\toks@\itf@thecr}%
   \fi
   \advance\itf@row\@ne
   \itf@col\z@
   \itf@colloop\do{%
      \advance\itf@col\@ne
      \ifnum\itf@col>\@ne
         \toks@\expandafter{\the\toks@&}%
```

```
1936          \fi
1937          \toks@\expandafter\expandafter\expandafter{%
1938                   \expandafter\the\expandafter\toks@
1939                   \itf@cell@init\itf@ampactive}%
1940          \itf@cell@insert}%
1941       }%
1942    \fi
1943    \itf@message{Code introduced by repeatcell in the \@currenvir:\the\toks@}% (to see the cod
1944    \expandafter\endgroup\the\toks@
1945    \ignorespaces
1946 }% \itf@repeatcell
```

\itf@message

```
1947 \let\itf@message\@gobble
```

Now the keys for \repeatcell:

```
1948 \def\itf@cell@rowcolkey{%
1949    \removefromstart\pgfkeyscurrentname{row}%
1950    \ifin@\else
1951       \removefromstart\pgfkeyscurrentname{col}%
1952    \fi
1953    \ifin@
1954       \itf@number\itf@cnt\pgfkeyscurrentname\relax\@nnil
1955          {\pgfkeyslet\pgfkeyscurrentkey\pgfkeyscurrentvalue}
1956          \in@false
1957    \fi
1958    \ifin@\else
1959       \pgfkeysvalueof{/errors/unknown key/.@cmd}\pgfkeyscurrentkey
1960                                                \pgfkeyscurrentvalue
1961                                                \pgfeov
1962    \fi
1963 }% \itf@cell@rowcolkey
1964 \pgfqkeys{/itf/makecell/repeatcell}{%
1965    font/.store in=\itf@cellfont,%
1966    font/.unknown/.code=\itf@cell@rowcolkey,%
1967    text/.store in=\itf@celltext,%
1968       text/.default=\ifdefined\C\C\column\fi,%
1969    text/.unknown/.code=\itf@cell@rowcolkey,%
1970    cr/.initial=\\,%
1971       cr/.default=\crcr,cr/.blank=\\,end/.alias=cr,%
1972    cr/.unknown/.code=\itf@cell@rowcolkey,%
1973    rows/.code=\itf@number\itf@row#1\relax\@nnil
1974                   {\ifnum\itf@row<\z@\itf@row\@ne\fi}
1975                   {\itf@row\z@
1976                   \def\itf@rows{#1}\itf@expandonceifcs\itf@rows},%
1977       rows/.default=1,rows/.blank=1,%
1978    @cols/.is private=interfaces-makecell,%
1979    column/.is private=interfaces-makecell,%
1980    row/.is private=interfaces-makecell,%
1981    transpose/.is if=itf@celltranspose,%
1982    .unknown/.code=%
1983       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text \itf@cell@formulaerror{}\z@ \fi
1984       \ifx\pgfkeyscurrentvalue\@empty \itf@cell@formulaerror{}\@ne \fi
1985       \pgfkeysifdefined{\pgfkeyscurrentkey/@calc}%
1986             {\itf@cell@formulaerror{}\tw@}
1987             {\pgfkeyslet{\pgfkeyscurrentkey/@calc}\relax}%
1988       \itf@cell@getformula{#1}%
1989       \pgfkeysEaddvalue{/itf/makecell/repeatcell/@calc}{}
1990          {\the\toks@\expandafter\noexpand\csname\pgfkeyscurrentname\endcsname}%
1991       \expandafter\let\csname\pgfkeyscurrentname\endcsname\@undefined,% (order of evaluation
1992    @reset/.code=% #1 are the columns
1993       \itf@celltransposefalse
```

```
      \let\itf@celltext\@empty
      \let\itf@cellfont\@empty
      \pgfkeyssetvalue{/itf/makecell/repeatcell/cr}{\\}%
      \itf@row\@ne
      \pgfkeyssetvalue{/itf/makecell/repeatcell/@calc}{%
         \itf@cell@numcalc\itf@col\column
         \itf@cell@numcalc\itf@row\row}%
      \itf@number\itf@col#1\relax\@nnil
         {\ifnum\itf@col<\z@\itf@col\@ne\fi}
         {\itf@col\z@
         \def\itf@cols{#1}\itf@expandonceifcs\itf@cols},%
}% keys for \repeatcell
```

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

```
\def\itf@cell@getformula#1{%
   \itf@cell@getf@rmula #1$$$\itf@cell@getformula$\relax\@nnil
   \expandafter\ifx\expandafter\toks@\the\toks@
      \itf@cell@formulaerror{#1}\thr@@
   \fi
}% \itf@cell@getformula
\def\itf@cell@getf@rmula #1$#2$#3$#4#5$#6{%
   \toks@{\toks@}%
   \ifx\itf@cell@getformula #4%
      \toks@{\itf@cell@numcalc{#1}}%
   \else\ifx $#4%
      \ifx\itf@cell@getformula #6%
         \ifdefined\pgfmathparse\else\itf@cell@formulaerror{#2}4\fi
         \toks@{\itf@cell@pgfcalc{#2}}%
      \else\ifx $#6%
         \ifdefined\FPeval\else\itf@cell@formulaerror{#3}5\fi
         \toks@{\itf@cell@FPcalc{#3}}%
      \fi\fi
   \fi\fi\remove@to@nnil
}% \itf@cell@getf@rmula
\def\itf@cell@formulaerror#1#2{%
   \def\itf@tp{#1}\@onelevel@sanitize\itf@tp
   \ifcase#2% 0
   \PackageError{interfaces-makecell}
      {Missing formula for key \pgfkeyscurrentname!}\@ehd
   \or %1
   \PackageError{interfaces-makecell}
      {Computed key \pgfkeyscurrentname\space wes left blank!}\@ehd
   \or %2
   \PackageError{interfaces-makecell}
      {Duplicate definition for computed key \pgfkeyscurrentname
      \MessageBreak\string\repeatcell\space will redefine the key}\@eha
   \or %3
   \PackageError{interfaces-makecell}
      {Incorrect formula in \string\repeatcell
      \MessageBreak Check: \itf@tp
      \MessageBreak Take care of commas and equal signs...}\@ehd
   \or %4
   \PackageError{interfaces-makecell}
      {Formula: \itf@tp
      \MessageBreak requires \string\pgfmathparse:\space\string\usetikz{pgfmath}}\@ehd
   \or %5
   \PackageError{interfaces-makecell}
      {Formula: \itf@tp
      \MessageBreak requires \string\FPeval:\space\string\usepackage{fp}}\@ehd
   \else
   \PackageError{interfaces-makecell}
      {Impossible value!}\@ehd
   \fi
}% \itf@cell@formulaerror
```

\itf@cell@calc
\itf@cell@numcalc
\itf@cell@pgfcalc
\itf@cell@FPcalc

```
\protected\def\itf@cell@calc#1#2{%
   \edef#1{\def\noexpand#1{#2}}%
   \toks@\expandafter\expandafter\expandafter{%
                     \expandafter\the\expandafter\toks@#1}%
   #1%
}% \itf@cell@calc
\protected\def\itf@cell@numcalc#1#2{%
   \itf@cell@calc#2{\number\numexpr#1}%
}% \itf@cell@numcalc
\protected\def\itf@cell@pgfcalc#1#2{%
   \pgfmathparse{#1}%
   \itf@cell@calc#2\pgfmathresult
}% \itf@cell@pgfcalc
\protected\def\itf@cell@FPcalc#1#2{%
   \begingroup
   \setbox\z@\vbox{%\FPeval introduces some weird code, hence the calculation in a \vbox
      \FPeval\FPresult{#1}%
      \global\let\FPresult\FPresult}%
   \endgroup
   \itf@cell@calc#2\FPresult
}% \itf@cell@FPcalc
```

\itf@cell@insert

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

```
\def\itf@cell@getifdefined#1#2{%
   \pgfkeysifdefined{/itf/makecell/repeatcell/#1}
      {\pgfkeysgetvalue{/itf/makecell/repeatcell/#1}#2}
      {\let#2\@empty}%
}%
\def\itf@cell@insert{%
   \pgfkeysvalueof{/itf/makecell/repeatcell/@calc}%
   \ifx\R\@empty\else
      \toks@\expandafter\expandafter\expandafter{%
         \expandafter\the\expandafter\toks@
         \expandafter\def\expandafter\r\expandafter{\r}}%
   \fi
   \ifx\C\@empty\else
      \toks@\expandafter\expandafter\expandafter{%
         \expandafter\the\expandafter\toks@
         \expandafter\def\expandafter\c\expandafter{\c}}%
   \fi
   \itf@cell@getifdefined{font/row\the\itf@row}\itf@cellfontrow
   \itf@cell@getifdefined{font/col\the\itf@col}\itf@cellfontcol
   \itf@cell@getifdefined{text/row\the\itf@row}\itf@cellrow
   \itf@cell@getifdefined{text/col\the\itf@col}\itf@cellcol
   \toks@\expandafter\expandafter\expandafter{%
          \expandafter\the\expandafter\toks@\itf@cellfont}%
   \ifx\itf@cellrow\@empty
      \ifx\itf@cellcol\@empty
         \itf@cell\expandafter\expandafter\expandafter{%
            \expandafter\itf@cellfontcol\itf@cellfontrow}%
         \let\itf@tp\itf@celltext
      \else
         \itf@cell\expandafter{\itf@cellfontcol}%
         \let\itf@tp\itf@cellcol
      \fi
   \else
      \ifx\itf@cellcol\@empty
         \itf@cell\expandafter{\itf@cellfontrow}%
         \let\itf@tp\itf@cellrow
      \else
         \itf@cell{}%
```

```
2115          \ifitf@celltranspose
2116          \edef\itf@tp{\noexpand\itf@diaghead(-2,1)
2117             {\expandonce\itf@cellfontrow\expandonce\itf@cellrow}
2118             {\expandonce\itf@cellfontcol\expandonce\itf@cellcol}}%
2119          \else
2120          \edef\itf@tp{\noexpand\itf@diaghead(-2,1)
2121             {\expandonce\itf@cellfontcol\expandonce\itf@cellcol}
2122             {\expandonce\itf@cellfontrow\expandonce\itf@cellrow}}%
2123          \fi
2124       \fi
2125    \fi
2126    \itf@cell\expandafter\expandafter\expandafter{%
2127                \expandafter\the\expandafter\itf@cell\itf@tp}%
2128    \edef\itf@tp{\itf@makecell{\the\itf@cell}}%
2129    \toks@\expandafter\expandafter\expandafter{%
2130                \expandafter\the\expandafter\toks@\itf@tp}%
2131 }% \itf@cell@insert
```

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

```
2132 \def\itf@diaghead(#1)#2#3{%
2133    \begingroup
2134    \edef\itf@tp{%
2135    \noexpand\diaghead({#1}){\unexpanded{#2 #3}}%
2136                    {\unexpanded{#2}}
2137                    {\unexpanded{#3}}}%
2138    \expandafter\endgroup\itf@tp
2139 }% \itf@diaghead
```

21.12.3 colortbl: patch for `\CT@cellc`

`\CT@cellc`

```
2140 \AfterPackage*{colortbl}{%
2141 \def\CT@cellc#1[#2]#3{%  #1 is either \@firstofone or \@gobble, [#2]#3 = color spec
2142    \xdef\CT@cell@color{\noexpand\CT@color#1{[{#2}]}{#3}%
2143    \global\let\noexpand\CT@cell@color\relax}%
2144 }}
```

```
2145 ⟨/makecell⟩
```

21.13 † Interface ☞enumitem : `\setlist`

```
2146 ⟨*enumitem⟩
2147 \ProvidesPackage{interfaces-enumitem}
2148    [2010/11/15 v2.1 more control on lists (FC)]
2149 \RequirePackage{interfaces-base}
2150 \itf@Check@Package{enumitem}
2151 \itf@pgfkeys
```

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

```
2152 \providerobustcmd*\interitem{%
2153    \item[]\hskip-\leftmargin
2154 }% \interitem
```

`\setlist` (extension, star form)

```
2155 \renewrobustcmd*\setlist{\begingroup \@ifstar
2156    {\let\do\csgappto \itf@setlist}
2157    {\let\do\csgdef   \itf@setlist}%
```

```
}% \setlist
\newrobustcmd*\itf@setlist[2][0]{%
   \let\enit@a\@empty
   \let\enit@b\@empty
   \@for\x:=#1\do{%
      \ifx\x\@empty\else
         \itf@number\count@\x\relax\@nnil
            {\edef\enit@b{\enit@b,\the\count@}}
            {\edef\enit@a{\enit@a,\x}}%
      \fi}%
   \ifx\enit@a\@empty\def\enit@a{list}\fi
   \ifx\enit@b\@empty\def\enit@b{0}\fi
   \@for\a:=\enit@a\do{%
      \ifx\a\@empty\else
      \@for\b:=\enit@b\do{%
         \ifx\b\@empty\else
         \ifcsundef{end\a}
            {\expandafter\itf@setlist@error\expandafter{\a}}
            {\do{enit@@\a\romannumeral\b}{,#2}}%
         \fi}%
      \fi}
   \endgroup
}% \itf@setlist
\def\itf@setlist@error#1{%
   \PackageError{interfaces-enumitem}
      {Environment `#1' is undefined
      \MessageBreak Use \string\newlist\space before \string\setlist
      \MessageBreak Your command will be ignored}\@eha
}% \itf@setlist@error
```

```
⟨/enumitem⟩
```

21.14 Interface truncate

```
⟨*truncate⟩
\ProvidesPackage{interfaces-truncate}
```

\trunctext

`\trunctext` is like `\truncate` but leaves the text to its natural width if possible.

```
\newrobustcmd\trunctext[3][\TruncateMarker]{% do nothing if width of #3 is less than #2
   \begingroup
      \hbadness\maxdimen\hfuzz\vsize
      \setbox\z@=\hbox to\dimexpr#2\relax{\ignorespaces#3\unskip}%
      \ifnum\badness>\@M
            \def\next{\truncate[{#1}]{#2}{#3}}%
      \else \def\next{#3}%
      \fi
      \expandafter\endgroup\next
}% \trunctext
```

\htruncadjust

```
\newrobustcmd\htruncadjust[1]{\@testopt
   {\itf@htruncadjust{#1}}{}%
}% \htruncadjust
\long\def\itf@htruncadjust#1[#2]#3{% #1=total width, #2=[x_1,...,x_n], #3=text_1,...,text_n
   \begingroup
   \let\itf@trunc@width\@empty
   \let\itf@istruncated\@gobble % to remove the first comma
   \let\the@dimen\relax
   \count@\z@  \dimen@\z@
   \ifblank{#2}% no optional [x_1,...,x_n] parameters
      {%
         \@for\x:=#3\do{%
```

```
            \advance \count@ \@ne
            \setbox\z@=\hbox{\ignorespaces\x\unskip}%
            \dimen\count@ = \wd\z@
            \toks\count@\expandafter{\the\dimen\count@}%
            \advance\dimen@ \dimen\count@}%
      }
      {%
         \@for\x:=#2\do{%
            \advance \count@ \@ne
            \toks\count@\expandafter{\x\p@}%
            \advance\dimen@ \x\p@
            \dimen\count@ -\@ne\p@}%
      }%
   \advance\count@\@ne \dimendef\itf@freespace=\count@
   \advance\count@\@ne \dimendef\itf@tempdim=\count@
   \advance\count@\@ne \dimendef\itf@sum=\count@
   \count@\z@     %%dimen0=\the\dimen@\quad\number\dimen@\par
   \itf@freespace\z@ % free space left when   w_i < x_i/sum(x_i) * total
   \@for\x:=#3\do{%
      \advance \count@ \@ne
      \ifdim \dimen\count@<\z@ % not measured before
            \setbox\z@=\hbox{\ignorespaces\x\unskip}%
            \dimen\count@ = \wd\z@
      \fi
      \itf@tempdim = \itf@scaledimen{#1}*\the\toks\count@/\dimen@\relax
      \ifdim \dimen\count@ > \itf@tempdim
            \dimen\count@ = \itf@tempdim
            \edef\itf@istruncated{\itf@istruncated,\the\count@}%
            \advance\itf@sum by\the\toks\count@
      \else
         \advance\itf@freespace \itf@tempdim
         \advance\itf@freespace -\dimen\count@
      \fi
      \edef\itf@trunc@width{\itf@trunc@width\the\count@{\the@dimen\the\count@}}%
   }%
%%   Free space=\the\itf@freespace\par
   \ifdim\itf@freespace > \z@ % reinsert free space proportionally
      \ifx\itf@istruncated\@gobble\else
      \@for\x:=\itf@istruncated\do{%
         \advance\dimen\x by\itf@scaledimen\itf@freespace*\the\toks\x/\itf@sum\relax
      }%
   \fi\fi
   \def\the@dimen{\the\dimen}%
   \edef\itf@trunc@width{\itf@trunc@width}%
   \expandafter\endgroup
      \expandafter\def
         \expandafter\itf@trunc@width
               \expandafter{\itf@trunc@width}%
%%   total=\the\dimexpr#1\relax
}% \itf@htruncadjust
\def\itf@scalefactor#1#2{\strip@pt\dimexpr#1*65536/#2\relax}
\def\itf@scaledimen#1*#2/#3\relax{%
   \dimexpr\itf@scalefactor{#2}{#3}\dimexpr#1\relax\relax
}% \itf@scaledimen
```

\htruncwidth

```
\newcommand*\htruncwidth[1]{% purely expandable
   \expandafter\itf@htruncwidth\itf@trunc@width\relax\relax\@nnil{#1}%
}% \htruncwidth
\def\itf@htruncwidth#1#2#3\@nnil#4{%
   \ifx\relax#1\itf@trunc@nosuchwidth{#4}%
   \else\ifnum#1=#4
      #2%
```

```
2273    \else
2274       \itf@htruncwidth#3\@nnil{#4}%
2275    \fi\fi
2276 }% \itf@htruncwidth
2277 \def\itf@trunc@nosuchwidth#1{%
2278    \PackageError{interfaces-truncate}
2279       {You requested the width for text nr#1
2280       \MessageBreak but there is no such text
2281       \MessageBreak Therefore no such width!}\@ehd
2282 }% \itf@trunc@nosuchwidth


2283 ⟨/truncate⟩
```

21.15 interface fancyvrb

```
2284 ⟨*fancyvrb⟩
2285 \ProvidesPackage{interfaces-fancyvrb}
2286    [2010/10/27 v1.2 Custom verbatim environments (FC)]
2287 \RequirePackage{interfaces-base}
2288 \itf@Check@Package{fancyvrb}
2289 \itf@pgfkeys
2290 \def\itf@tp#1{%
2291 \DefineVerbatimEnvironment{itf:VerbLines}{Verbatim}
2292    {gobble=#1,frame=lines,framesep=6pt,fontseries=m}
2293 \DefineVerbatimEnvironment{itf:Verb}{Verbatim}
2294    {gobble=#1,fontseries=m}
2295 \DefineVerbatimEnvironment{itf:Verb*}{Verbatim}
2296    {gobble=#1,fontseries=m,commandchars=$()}
2297 }
2298 \ltx@ifpackageloaded{doc}
2299             {\itf@tp{1}}
2300             {\itf@tp{0}}
2301 \AtBeginDocument{%
2302 \ifdefined\VerbLines\else
2303 \ifdefined\endVerbLines\else
2304    \letcs\VerbLines{itf:VerbLines}%
2305    \letcs\endVerbLines{enditf:VerbLines}%
2306 \fi\fi
2307 \ifdefined\Verb\else\ifdefined\endVerb\else
2308    \letcs\Verb{itf:Verb}%
2309    \letcs\endVerb{enditf:Verb}%
2310 \fi\fi
2311 \ifcsname Verb*\endcsname\else
2312 \ifcsname endVerb*\endcsname\else
2313    \csletcs{Verb*}{itf:Verb*}%
2314    \csletcs{endVerb*}{enditf:Verb*}%
2315 \fi\fi
2316 }
2317 ⟨/fancyvrb⟩
```

21.16 Interface ☞appendix and the appendices environment

```
2318 ⟨*appendix⟩
2319 \ProvidesPackage{interfaces-appendix}
2320    [2010/10/27 v1.2 easy control of appendices environment (FC)]
2321 \RequirePackage{interfaces-base}
2322 \itf@Check@Package{appendix}
2323 \itf@pgfkeys
```

\appendices

```
2324 \let\itf@appendices@pp\appendices
2325 \let\itf@endappendices@pp\endappendices
2326 \let\itf@appendices@pp\appendices
2327 \let\itf@endappendices@pp\endappendices
```

```
\renewenvironment{appendices}[1][]{%
   \pgfqkeys{/itf/appendix/appendices}{#1}%
   \itf@appendices@pp
   }
   {\itf@endappendices@pp}%

\pgfqkeys{/itf/appendix/appendices}{%
   title/.store as code in cs=appendixpagename,%
      title/.append code=\ifx\appendixpagename\@empty \@dopage@ppfalse
                          \else \@dopage@pptrue\fi,%
      title/.value required,%
   intoc/.is if=@dotoc@pp,%
   intoc/title/.store as code in cs=appendixtocname,%
      intoc/title/.append code=\ifx\appendixtocname\@empty \@dotoc@ppfalse
                                \else \@dotoc@pptrue\fi,%
      intoc/title/.default=\pgfkeysvalueof{/itf/appendix/appendices/title},%
   section/.is enum={section,chapter},%
      section/.append code={%
         \ifcase\pgfkeysvalueof{/itf/appendix/appendices/section/.number}\relax
         \@chapter@ppfalse\or\@chapter@pptrue\fi
         \if@chapter@pp\ifundef\chapter{%
            \PackageError{interfaces-appendix}
               {You asked chapter-style appendices but your
               \MessageBreak current class does not provide chapters!}\@eha
            \pgfkeysalso{/itf/appendix/appendices/section}}{}%
         \fi},%
   chapter/.style={section=chapter},%
      chapter/.value forbidden,%
}% keys for appendices environment
```

⟨/appendix⟩

21.17 Interface ☞embedfile : \embedthisfile

```
⟨*embedfile⟩
\ProvidesPackage{interfaces-embedfile}
   [2010/10/27 v1.2 easy macros to embed files into pdf (FC)]
\RequirePackage{interfaces-base}
\itf@Check@Package{embedfile}
\newrobustcmd*\embedthisfile{\@ifstar
   \embedthisfile@main
   \embedthisfile@all
}% \embedthisfile
\@onlypreamble\embedthisfile
\def\embedthisfile@main{\embedfile{\jobname.tex}}
\def\embedthisfile@all{%
   \embedthisfile@main
   \AfterPreamble{%
      \ltx@ifundefined{@listfiles}
         {\let\@listfiles\@empty}% to avoid: \global\let\@filelist\relax in \document
         {}%
   }%
   \gpreto\@enddocumenthook{%
      \let\do\itf@EmbedOne
      \expandafter\docsvlist\expandafter{\@filelist}%
   }%
}% \embedthisfile@all
\def\itf@EmbedOne#1{\itf@Embed@ne#1.tex\@nil}
\def\itf@Embed@ne#1.tex#2\@nil{%
   \ltx@ifundefined{cp@#1}
      {}
      {\ifblank{#2}
         {}
         {\embedfile{#1.tex}}}%
```

```
}% \itf@Embed@ne
⟨/embedfile⟩
```

21.18 Interface environ

```
⟨*environ⟩
\ProvidesPackage{interfaces-environ}
   [2010/07/23 v1.0 ReNewEnviron = RenewEnviron (FC)]
\RequirePackage{interfaces-base}
\AfterPackage{environ}{
   \ifundef\ReNewEnviron
      {\let\ReNewEnviron\RenewEnviron}
      {}%
}%
⟨/environ⟩
```

21.19 Interface umrand

```
⟨*umrand⟩
\ProvidesPackage{interfaces-umrand}
   [2010/07/23 v1.0 patch for umrand.sty ifnotempty (FC)]
\RequirePackage{interfaces-base}
\BeforePackage{umrand}{%
   \let\itf@newif\newif
   \def\itf@tp{\ifnotempty}%
   \def\newif#1{\def\x{#1}\ifx\x\itf@tp \expandafter\@gobble
      \else\expandafter\ltx@newif\fi{#1}}%
}
\AfterPackage{umrand}{%
   \let\newif\itf@newif
   \let\itf@newif\@undefined
}
⟨/umrand⟩
```

21.20 Interface scrlfile

```
⟨*scrlfile⟩
\ProvidesPackage{interfaces-scrlfile}
   [2010/10/27 v1.2 scrlfile hooks improved (FC)]
\RequirePackage{interfaces-base}
```

`\@pkgextension` `\@clsextension`

In LaTeX they are defined `\@onlypreamble`. So we use the ltxcmds equivalents:

```
\AfterEndPreamble{%
   \let\@pkgextension\ltx@pkgextension
   \let\@clsextension\ltx@clsextension
   \let\@ifpackageloaded\ltx@ifpackageloaded
   \let\@ifclassloaded\ltx@ifclassloaded
}
```

`\scr@load@hook`

```
\renewrobustcmd*\scr@load@hook[2]{%
   \csuse{#2-@#1}%
   \ifcsname#2-@#1-\string\keep\endcsname\else\global\csundef{#2-@#1}\fi
}% \scr@load@hook
```

`\scr@set@load@hook` (optimisation)

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

\BeforeFile
\AfterFile

Addition of star forms *: the hooks are not canceled after input (in case a file is input more than once...)

```
\renewrobustcmd*\BeforeFile{\@ifstar
   {\itf@s@LoadFile{before}}
   {\itf@LoadFile{before}}%
}% \BeforeFile
\renewrobustcmd*\AfterFile{\@ifstar
   {\itf@s@LoadFile{after}}
   {\itf@LoadFile{after}}%
}% \AfterFile

\def\itf@s@LoadFile#1#2{%
   \global\cslet{#2-@#1-\string\keep}\@empty
   \scr@set@load@hook{#1}{#2}%
}% \itf@s@LoadFile
\def\itf@LoadFile#1#2{%
   \global\csundef{#2-@#1-\string\keep}%
   \scr@set@load@hook{#1}{#2}%
}% \itf@LoadFile
```

\@input (patch)

This modification allows \@input to behaves like \InputIfFileExists as far as hooks are concerned.

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

⟨/scrlfile⟩
```

21.21 † Interface pgfkeys (☞pgfmanual)

21.21.1 interfaces-pgfkeys

```
⟨*pgfkeys⟩
\ProvidesPackage{interfaces-pgfkeys}
   [2010/10/27 v1.2 pgfkeys handlers definitions (FC)]
\RequirePackage{interfaces-base}
\ifdefined\pgfkeys@ifexecutehandler % pgf version 2.1
\def\itf@pgfkeys{%
   \global\let\pgfkeys@unpack       \itf@pgfkeys@unpack
   \global\let\pgfkeys@case@three   \itf@pgfkeys@case@three
   \global\let\pgfkeys@case@three@handleall\itf@pgfkeys@case@three@handleall
}% \itf@pgfkeys
\else
\def\itf@pgfkeys{%
   \global\let\pgfkeys@unpack       \itf@pgfkeys@unpack
   \global\let\pgfkeys@case@three   \itf@pgfkeys@case@three
   \global\let\pgfkeysaddvalue      \itf@pgfkeysaddvalue
   \global\let\pgfkeys@exp@call     \itf@pgfkeys@exp@call
}% \itf@pgfkeys
\fi
\AtBeginDocument{%
   \itf@pgfkeys
   \let\itf@pgfkeys\relax
}% AtBeginDocument
```

Some new pgfkeys macros

\pgfkeysmeaning

```
\providecommand*\pgfkeysmeaning[1]{%
   \pgfkeysifdefined{#1}
```

```
      {\expandafter\meaning\csname pgfk@#1\endcsname}
      {\meaning\@undefined}%
}%
```

\pgfkeysvalueof@unexpanded

```
\def\pgfkeysvalueof@unexpanded#1{%
   \unexpanded\expandafter\expandafter\expandafter{%
                           \csname pgfk@#1\endcsname}%
}% \pgfkeysvalueof@unexpanded
```

Some pgfkeys macros redefinition/corrections

\pgfkeysaddvalue (bug corrected)

Bug in \pgfkeysaddvalue for pgf v2.0, fixed in v2.1

```
\ifdefined\pgfkeys@ifexecutehandler \else % pgf v2.1: bug fixed
\def\itf@pgfkeysaddvalue#1#2#3{% (pgf v2.0: bug in pgfkeys.code.tex)
   \edef\pgfkeys@global@temp{%
      \unexpanded{#2}%
      \ifcsname pgfk@#1\endcsname
         \unexpanded\expandafter\expandafter\expandafter{%
                           \csname pgfk@#1\endcsname}%
      \fi
      \unexpanded{#3}}%
   \pgfkeyslet{#1}\pgfkeys@global@temp
}% \itf@pgfkeysaddvalue
\fi
```

\pgfkeysEsetvalue
\pgfkeysEaddvalue

```
\ifdefined\pgfkeys@ifexecutehandler \long \fi % pgf version 2.1
\def\pgfkeysEsetvalue#1#2{%
   \expandafter\edef\csname pgfk@#1\endcsname{#2}%
}% \pgfkeysEsetvalue
\ifdefined\pgfkeys@ifexecutehandler \long \fi % pgf v2.1
\def\pgfkeysEaddvalue#1#2#3{% \csepreappto{pgfk@#1}{#2}{#3}
   \edef\pgfkeys@global@temp{%
      #2%
      \ifcsname pgfk@#1\endcsname
         \unexpanded\expandafter\expandafter\expandafter{%
            \csname pgfk@#1\endcsname}%
      \fi
      #3}%
   \pgfkeyslet{#1}\pgfkeys@global@temp
}% \pgfkeysEaddvalue
```

\pgfkeys@exp@call

Missed the braces around the argument (in case the argument contains a comma or an = sign...):

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

Handlers /.blank

```
%% The value is now checked by \itf@pgfkeys@default@blank@required:
\ifdefined\pgfkeys@ifexecutehandler \long \fi % pgf v2.1
\def\itf@pgfkeys@unpack#1=#2=#3\pgfkeys@stop{%
  \pgfkeys@spdef\pgfkeyscurrentkey{#1}%
  \edef\pgfkeyscurrentkey{\pgfkeyscurrentkey}%
  \ifx\pgfkeyscurrentkey\pgfkeys@empty
    % Skip
  \else%
    \pgfkeys@add@path@as@needed
    \pgfkeys@spdef\pgfkeyscurrentvalue{#2}%
    \itf@pgfkeys@default@blank@required % <FC>
                    \pgfkeyscurrentkey % <FC>
                     \pgfkeys@case@one % <FC>
  \fi
}% \itf@pgfkeys@unpack
```

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

```
\def\itf@pgfkeys@default@blank@required#1#2{%
   \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text% Hmm... no value
      \pgfkeysifdefined{#1/.@def}%
         {\pgfkeysgetvalue{#1/.@def}\pgfkeyscurrentvalue}
         {}% no default, so leave it
   \fi
   \ifx\pgfkeyscurrentvalue\@empty
      \pgfkeysifdefined{#1/.@blank}
         {\pgfkeysgetvalue{#1/.@blank}\pgfkeyscurrentvalue}
         {}% no /.blank specified
   \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text% in case /.blank -> \pgfkeysnovalue
      \pgfkeysifdefined{#1/.@def}%
         {\pgfkeysgetvalue{#1/.@def}\pgfkeyscurrentvalue}
         {}% no default, so leave it
   \fi\fi
   \ifx\pgfkeyscurrentvalue\pgfkeysvaluerequired
      \pgfkeysvalueof{/errors/value required/.@cmd}\pgfkeyscurrentkey
                                                   \pgfkeyscurrentvalue
                                                   \pgfeov
   \else
      #2%
   \fi
}% \itf@pgfkeys@default@blank@required
```

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

```
%% To handle the cases of: /handler/.default= /handler/.blank= /handler/.value required,
%% it is necessary to redefine \pgfkeys@case@three
\def\itf@pgfkeys@case@three{%
   \pgfkeys@split@path
   \pgfkeysifdefined{/handlers/\pgfkeyscurrentname/.@cmd}
      {%
      \itf@pgfkeys@default@blank@required{/handlers/\pgfkeyscurrentname}%
         {\pgfkeysgetvalue{/handlers/\pgfkeyscurrentname/.@cmd}\pgfkeys@code
          \expandafter\pgfkeys@code\pgfkeyscurrentvalue\pgfeov}%
      }
     \pgfkeys@unknown % no handler, thus the key is unknown
}% \itf@pgfkeys@case@three
\ifdefined\pgfkeys@ifexecutehandler
   \let\itf@pgfkeys@case@three@handleall \itf@pgfkeys@case@three
\fi
```

Loading interfaces modifications for pgfkeys

```
\itf@pgfkeys
```

Handlers definition: /.alias, /.blank

```
\pgfkeys{%
   /handlers/.alias/.code=\itf@pgfkeys@ifaliasexists{#1}{%
      \pgfkeysedef\pgfkeyscurrentpath{\noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}}},%
      /handlers/.alias/.value required,%
   /handlers/.blank/.code=\pgfkeyssetvalue{\pgfkeyscurrentpath/.@blank}{#1},%
      /handlers/.blank/.default=\pgfkeysnovalue,%
}% /.alias   and  /.blank
\def\itf@pgfkeys@ifaliasexists#1#2{%
   \pgfkeysifdefined{\pgfkeysdefaultpath#1/.@cmd}{#2}
      {\pgfkeysifdefined{\pgfkeysdefaultpath#1}
         {#2}
         {\itf@pgfkeys@aliaserror{#1}}}%
}% \itf@pgfkeys@ifaliasexists
\def\itf@pgfkeys@aliaserror#1{%
   \PackageError{interfaces-pgfkeys}
      {You want the key \pgfkeyscurrentpath\space
      \MessageBreak to be an alias for \pgfkeysdefaultpath#1
      \MessageBreak but the key \pgfkeysdefaultpath#1
      \MessageBreak is not defined!}\@eha\@gobble
}% \itf@pgfkeys@aliaserror
```

Handlers 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).

```
\pgfkeys{
   /handlers/.alias*/.code=\itf@pgfkeys@ifaliasexists{#1}{%
      \pgfkeysedef\pgfkeyscurrentpath{\noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}}%
      \@tfor\x:=+-*\do{%
         \pgfkeysifdefined{\pgfkeysdefaultpath#1\x/.@cmd}
            {\pgfkeysedef{\pgfkeyscurrentpath\x}{%
               \noexpand\pgfkeysalso{\pgfkeysdefaultpath#1\x={##1}}}}
            {}%
      }},%
   /handlers/.alias*/.value required,%
}% /.alias*
```

Handlers definition: /.is private

```
\pgfkeys{%
   /handlers/.is private/.code=\pgfkeysdef\pgfkeyscurrentpath{%
      \PackageError{#1}{Key \pgfkeyscurrentkey\space is private
      \MessageBreak and should not be modified}\@ehd},%
   /handlers/.is private to class/.code=\pgfkeysdef\pgfkeyscurrentpath{%
      \ClassError{#1}{Key \pgfkeyscurrentkey\space is private
      \MessageBreak and should not be modified}\@ehd}%
}% /.is private
```

Handlers definition: /.true /.false /.on /.off

```
\def\itf@pgfkeys@choice[#1]#2#3{%
   \pgfkeys{\pgfkeyscurrentpath/.cd,%
      .is choice,%
      #2/.code={#3},%
      .default/.expanded=\@firstoftwo#1}%
}% \itf@pgfkeys@choice
\pgfkeys{%
   /handlers/.true/.code=\itf@pgfkeys@choice[{true}{false}]{true}{#1},%
   /handlers/.false/.code=\itf@pgfkeys@choice[{true}{false}]{false}{#1},%
   /handlers/.on/.code=\itf@pgfkeys@choice[{on}{off}]{on}{#1},%
   /handlers/.off/.code=\itf@pgfkeys@choice[{on}{off}]{off}{#1},%
}% /.true /.false
```

Handlers definition: /.store in cs /.estore in cs etc.

```
\pgfkeys{%
   /handlers/.store in cs/.code=\pgfkeysdef\pgfkeyscurrentpath{\csdef{#1}{##1}},%
   /handlers/.estore in cs/.code=\pgfkeysdef\pgfkeyscurrentpath{\csedef{#1}{##1}},%
}% /.store in cs
```

Handler definition: /.store as code

```
%% /.store as code handlers
\pgfkeys{%
   /handlers/.store as code/.code={%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
      \pgfkeyslet\pgfkeyscurrentpath\@empty
      \pgfkeyslet{\pgfkeyscurrentpath/.@def}\@empty
      \@tfor\x:=#1\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
               \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
               \noexpand\unexpanded{##1}}}}%
      \pgfkeyslet{\pgfkeyscurrentpath\x/.@def}\@empty
      }},%
   /handlers/.store as code/.default=*+,%
}% /.store as code
```

Handlers definition: /.estore as code

```
\pgfkeys{%
   /handlers/.estore as code/.code={%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{##1}}%
      \pgfkeyslet\pgfkeyscurrentpath\@empty
      \@tfor\x:=#1\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
               \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
               \noexpand\unexpanded{##1}}}}%
      }},%
   /handlers/.estore as code/.default=*+,%
}% /.estore as code
```

Handlers definition: /.store as code in cs=⟨cs-name⟩

```
\pgfkeys{%
   /handlers/.store as code in cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else\csdef{#1}{##1}\fi}%
      \@tfor\x:=*+\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
               \noexpand\ifcsname#1\endcsname
                  \noexpand\expandafter\noexpand\expandonce
                     \noexpand\csname#1\endcsname
               \noexpand\fi
               \noexpand\unexpanded{##1}}}}%
         \pgfkeyslet{\pgfkeyscurrentpath\x/.@def}\@empty
      }},%
   /handlers/.store as code in cs/.value required,%
}% /.store as code in cs
\pgfkeys{%
   /handlers/.estore as code in cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else\csedef{#1}{##1}\fi}%
      \@tfor\x:=*+\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
               \noexpand\ifcsname#1\endcsname
                  \noexpand\expandafter\noexpand\expandonce
                        \noexpand\csname#1\endcsname
               \noexpand\fi
               \noexpand\unexpanded{##1}}}}%
      }},%  default if \pgfkeysnovalue which expands to \@empty
   /handlers/.estore as code in cs/.value required,%
}% /.estore as code in cs
```

Handlers definition: /.store as color, /.store as color command

```
\protected\def\itf@pgfk@color#1#2\@nil{%
   \ifx\@let@token\color
      \def\pgfkeyscurrentvalue{#2}%
   \else\ifx\@let@token\normalcolor
      \def\pgfkeyscurrentvalue{default}%
   \else\ifx\@let@token\pgfkeysnovalue
      \def\pgfkeyscurrentvalue{default}%
   \fi\fi\fi
}% \itf@pgfk@color
\pgfkeys{%
   /handlers/.store as color/.code=%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \futurelet\noexpand\@let@token\itf@pgfk@color
                        ##1\noexpand\pgfkeysnovalue\noexpand\@nil
         \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
}% /.store as color
```

Handlers definition: /.store as color in cs=⟨cs-name⟩

```
\pgfkeys{%
   /handlers/.store as color in cs/.code=%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \futurelet\noexpand\@let@token\itf@pgfk@color
                        ##1\noexpand\pgfkeysnovalue\noexpand\@nil
         \cslet{#1}\pgfkeyscurrentvalue}%
}% /.store as color in cs
```

Handlers definition: /.store as color command

```
\protected\def\itf@pgfk@colorcmd#1\@nil{%
   \ifcat @\noexpand\@let@token
      \def\pgfkeyscurrentvalue{\color{#1}}%
   \else\ifcat 0\noexpand\@let@token
      \def\pgfkeyscurrentvalue{\color{#1}}%
   \else\ifx\@let@token\@nil
      \let\pgfkeyscurrentvalue\normalcolor
   \fi\fi\fi
}% \itf@pgfk@colorcmd
\pgfkeys{%
   /handlers/.store as color command/.code=%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \futurelet\noexpand\@let@token\itf@pgfk@colorcmd
                        ##1\noexpand\@nil
         \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
      \pgfkeyssetvalue{\pgfkeyscurrentpath/.@def}\normalcolor
}% /.store as color command
```

Handlers definition: /.store as color command in cs=⟨cs-name⟩

```
\pgfkeys{%
   /handlers/.store as color command in cs/.code=%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \futurelet\noexpand\@let@token\itf@pgfk@colorcmd
                        ##1\noexpand\@nil
         \cslet{#1}\noexpand\pgfkeyscurrentvalue}%
      \pgfkeyssetvalue{\pgfkeyscurrentpath/.@def}\normalcolor
}% /.store as color command in cs
```

Handlers definition /.store as number

```
\pgfkeys{%
   /handlers/.store as number/.code={%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\numexpr##1\relax}}%
      \pgfkeyslet\pgfkeyscurrentpath\z@
      \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
      \@tfor\x:=#1\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
               \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
               \x
               \numexpr\unexpanded{##1}\relax}}%
         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@\fi}%
      }},%
   /handlers/.store as number/.default=+-*,%
}% /.store as number
```

Handlers definition: /.estore as number

```
\pgfkeys{%
   /handlers/.estore as number/.code={%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{%
                     \noexpand\number\numexpr##1\relax}}%
      \pgfkeyslet\pgfkeyscurrentpath\z@
      \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
      \@tfor\x:=#1\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
               \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}%
               \x
               \numexpr##1\relax}}%
      }},%
   /handlers/.estore as number/.default=+-*,%
}% /.estore as number
```

/.store as number in cs Handlers definition: /.store as number in cs

```
\pgfkeys{%
   /handlers/.store as number in cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else\csdef{#1}{\numexpr##1\relax}\fi}%
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
      \@tfor\x:=+-*\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \letcs\noexpand\pgfkeys@temp{%
               \noexpand\ifcsname#1\endcsname
                  #1\noexpand\else\if*\x @ne\else z@\fi\noexpand\fi}%
         \edef\noexpand\pgfkeys@temp{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath={%
               \noexpand\expandonce\noexpand\pgfkeys@temp
               \x
               \numexpr\noexpand\unexpanded{##1}\relax}}%
         }\noexpand\pgfkeys@temp}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@\fi}%
   }},%
   /handlers/.store as number in cs/.value required,%
}% /.store as number in cs
```

Handlers definition: /.estore as number in cs

```
\pgfkeys{%
   /handlers/.estore as number in cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else\csedef{#1}{\number\numexpr##1\relax}\fi}%
   \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
   \@tfor\x:=+-*\do{%
      \pgfkeysedef{\pgfkeyscurrentpath\x}{%
         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
            \noexpand\csname\noexpand\ifcsname#1\endcsname
               #1\noexpand\else\if*\x @ne\else z@\fi\noexpand\fi\endcsname
            \x
            \numexpr##1\relax}}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@\fi}%
   }},%
   /handlers/.estore as number in cs/.value required,%
```

```
}% /.estore as number in cs
```

Handlers definition: /.store in counter cs

```
\pgfkeys{%
   /handlers/.store in counter cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else\csname#1\endcsname\numexpr##1\relax\fi}%
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
   \@tfor\x:=+-*\do{%
      \pgfkeysedef{\pgfkeyscurrentpath\x}{%
         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
            \noexpand\csname\noexpand\ifcsname#1\endcsname
            #1\noexpand\else\if*\x @ne\else z@\fi\noexpand\fi\endcsname
            \x
            \numexpr##1\relax}}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\@ne\else\z@\fi}%
   }},%
   /handlers/.store in counter cs/.value required,%
}% /.store in counter cs
```

Handlers definition: /.store as glue

```
\pgfkeys{%
   /handlers/.store as glue/.code={%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\glueexpr##1\relax}}%
      \pgfkeyslet\pgfkeyscurrentpath\z@skip
      \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@skip
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
      \@tfor\x:=#1\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
               \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
               \x
               \if*\x\noexpand\itf@units@scale##1!
               \else\glueexpr\noexpand\unexpanded{##1}\relax\fi}}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@skip\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@skip\fi}%
      }},%
   /handlers/.store as glue/.default=+-*,%
}% /.store as glue
```

Handlers definition: /.estore as glue

```
\pgfkeys{%
   /handlers/.estore as glue/.code={%
      \pgfkeysedef{\pgfkeyscurrentpath}{%
         \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{%
                     \noexpand\the\glueexpr##1\relax}}%
      \pgfkeyslet\pgfkeyscurrentpath\z@skip
      \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@skip
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
      \@tfor\x:=#1\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
               \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\x
               \if*\x \noexpand\itf@units@scale##1!
               \else\glueexpr##1\relax\fi}}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@skip\fi}%
```

```
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@skip\fi}%
      }},%
   /handlers/.estore as glue/.default=+-*,%
}% /.estore as glue
```

Handlers definition: /.store as dimen

```
\pgfkeys{%
   /handlers/.store as dimen/.code={%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\dimexpr##1\relax}}%
      \pgfkeyslet\pgfkeyscurrentpath\z@
      \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
      \@tfor\x:=#1\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
               \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
               \x
               \if*\x \noexpand\itf@units@scale##1!
               \else\dimexpr\noexpand\unexpanded{##1}\relax\fi}}}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@\fi}%
      }},%
   /handlers/.store as dimen/.default=+-*,%
}% /.store as dimen
```

Handlers definition: /.estore as dimen

```
\pgfkeys{%
   /handlers/.estore as dimen/.code={%
      \pgfkeysedef\pgfkeyscurrentpath{%
         \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{%
                  \noexpand\the\dimexpr##1\relax}}%
      \pgfkeyslet\pgfkeyscurrentpath\z@
      \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
      \@tfor\x:=#1\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
               \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\x
               \if*\x   \noexpand\itf@units@scale##1!
               \else    \dimexpr##1\relax\fi}}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@\fi}%
      }},%
   /handlers/.estore as dimen/.default=+-*,%
}% /.estore as dimen
```

Handlers definition: /.store as glue in cs=⟨*cs-name*⟩

```
\pgfkeys{%
   /handlers/.store as glue in cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else\csdef{#1}{\glueexpr##1\relax}\fi}%
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
      \@tfor\x:=+-*\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \letcs\noexpand\pgfkeys@temp{%
               \noexpand\ifcsname#1\endcsname
                  #1\noexpand\else z@skip\noexpand\fi}%
```

```
            \edef\noexpand\pgfkeys@temp{%
               \noexpand\pgfkeysalso{\pgfkeyscurrentpath={%
                  \noexpand\expandonce\noexpand\pgfkeys@temp
                  \x
                  \if*\x\noexpand\itf@units@scale##1!
                  \else\glueexpr\noexpand\unexpanded{##1}\relax\fi}}%
            }\noexpand\pgfkeys@temp}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@skip\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@skip\fi}%
   }},%
   /handlers/.store as glue in cs/.value required,%
}% /.store as glue in cs
```

Handlers definition: /.estore as glue in cs=⟨*cs-name*⟩

```
\pgfkeys{%
   /handlers/.estore as glue in cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else\csedef{#1}{\the\glueexpr##1\relax}\fi}%
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
      \@tfor\x:=+-*\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
               \noexpand\csname\noexpand\ifcsname#1\endcsname
                  #1\noexpand\else z@skip\noexpand\fi\endcsname
               \x
               \if*\x\noexpand\itf@units@scale##1!
               \else\glueexpr##1\relax\fi}}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@skip\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@skip\fi}%
   }},%
   /handlers/.etore as glue in cs/.value required,%
}% /.estore as glue in cs
```

Handlers definition: /.store in skip cs

This is a variant of /.store as glue in cs when the control sequence in a \skip (*ie.* a length). The skip is given by its name, not as a control sequence.

```
\pgfkeys{%
   /handlers/.store in skip cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else\csname#1\endcsname\glueexpr##1\relax\fi}%
      \pgfkeyslet{/\pgfkeyscurrentpath/.@blank}\z@skip
      \@tfor\x:=+-*\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
               \csname#1\endcsname  % should have been defined as a skip (not expandable)
               \x
               \if*\x\noexpand\itf@units@scale##1!
               \else\glueexpr##1\relax\fi}}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@skip\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@skip\fi}%
   }},%
   /handlers/.store in skip cs/.value required,%
}% /.store in skip cs
```

Handlers definition: /.store as dimen in cs=⟨cs-name⟩

```
\pgfkeys{%
   /handlers/.store as dimen in cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else\csdef{#1}{\dimexpr##1\relax}\fi}%
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
      \@tfor\x:=+-*\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \letcs\noexpand\pgfkeys@temp{%
               \ifcsname#1\endcsname#1\else z@\fi}%
            \edef\noexpand\pgfkeys@temp{%
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath={%
               \noexpand\expandonce\noexpand\pgfkeys@temp
               \x
               \if*\x\noexpand\itf@units@scale##1!
               \else\dimexpr\noexpand\unexpanded{##1}\fi}}%
            }\noexpand\pgfkeys@temp}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@\fi}%
   }},%
   /handlers/.store as dimen in cs/.value required,%
}% /.store as dimen in cs {csname}{+-}
```

Handlers definition: /.estore as dimen in cs=⟨cs-name⟩

```
\pgfkeys{%
   /handlers/.estore as dimen in cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else \@defaultunits\itf@sk ##1pt\relax\@nnil
               \csedef{#1}{\the\dimexpr\itf@sk\relax}\fi}%
      \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
      \@tfor\x:=+-*\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \noexpand\ifcsname#1\endcsname
               \noexpand\@defaultunits\itf@sk
                  \noexpand\csname#1\endcsname pt\relax\noexpand\@nnil
            \noexpand\else\itf@sk\z@\noexpand\fi
            \if*\x
               \itf@sk\dimexpr\itf@sk*\noexpand\itf@units@scale##1!\relax
            \else
               \noexpand\@defaultunits\@tempdima ##1pt\relax\noexpand\@nnil
               \advance\itf@sk\x\@tempdima
            \fi
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath=\itf@sk}}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@\fi}%
   }},%
   /handlers/.etore as dimen in cs/.value required,%
}% /.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.

```
\pgfkeys{%
   /handlers/.store in dimen cs/.code={%
      \pgfkeysdef\pgfkeyscurrentpath{%
         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
         \else\csname#1\endcsname\dimexpr##1\relax\fi}%
      \pgfkeyslet{/\pgfkeyscurrentpath/.@blank}\z@
      \@tfor\x:=+-*\do{%
         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
            \if*\x \else
            \noexpand\@defaultunits\itf@sk ##1pt\relax\noexpand\@nnil
            \fi
            \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
               \csname#1\endcsname  % should have been defined as a dimen (not expandable)
               \x
               \if*\x \noexpand\itf@units@scale##1!
               \else\itf@sk\fi}}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@\fi}%
   }},%
   /handlers/.store in dimen cs/.value required,%
}% /.store in dimen cs
```

Handlers definition: /.is multiple choice

```
\pgfkeys{%
   /handlers/.is multiple choice/.code=%
      \pgfkeys{\pgfkeyscurrentpath/.cd,%
         .unknown/.style={/errors/unknown choice value=\pgfkeyscurrentkey\pgfkeyscurrentvalue
         .code=\expandafter\pgfkeys\expandafter{\pgfkeyscurrentkey/.cd,####1}%
      },%
   /handlers/.is multiple choice/.value forbidden,%
}% /.is multiple choice
```

Handlers definition: /.is enum

```
\pgfkeys{%
   /handlers/.is enum/.code={%
      \itf@cnt\z@
      \@for\x:=\pgfkeyscurrentvalue\do{%
         \pgfkeysedef{\pgfkeyscurrentpath/\x}{%
            \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\x}%
            \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath/.number}{\the\itf@cnt}}%
         \ifnum\itf@cnt=\z@
            \pgfkeyslet{\pgfkeyscurrentpath/.@def}\x
            \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\x
         \fi
         \advance\itf@cnt\@ne
      }%
      \pgfkeysalso{\pgfkeyscurrentpath/.is choice}},%
   /handlers/.is enum/.value required,%
}% /.is enum
```

⟨/pgfkeys⟩

21.22 Interface ☞etoolbox - etoolbox-like commands

```
3055 ⟨*etoolbox⟩
3056 \ProvidesPackage{interfaces-etoolbox}
3057    [2010/10/27 v1.2 some more etoolbox-like commands (FC)]
3058 \RequirePackage{interfaces-base}
3059 \begingroup\catcode`\&=3
3060    \long\gdef\etb@doitem#1&{\do{#1}}
3061 \endgroup
```

\preappto
\epreappto
\gpreappto
\xpreappto

```
3062 \providerobustcmd\preappto[3]{%
3063    \edef#1{\unexpanded{#2}\ifdefined#1\expandonce#1\fi\unexpanded{#3}}%
3064 }% \preappto
3065 \providerobustcmd\epreappto[3]{%
3066    \edef#1{#2\ifdefined#1\expandonce#1\fi#3}%
3067 }% \epreappto
3068 \providerobustcmd*\gpreappto{\global\preappto}
3069 \providerobustcmd*\xpreappto{\global\epreappto}
```

\cspreappto
\csepreappto
\csgpreappto
\csxpreappto

```
3070 \providerobustcmd*\cspreappto[1]{%
3071    \csname itf@\ifcsname#1\endcsname
3072       preappto\else defpreapp\fi\expandafter\endcsname
3073                                \csname#1\endcsname
3074 }% \cspreappto
3075 \long\def\itf@preappto#1#2#3{\edef#1{#2\expandonce#1#3}}
3076 \long\def\itf@defpreapp#1#2#3{\edef#1{#2#3}}
3077 \providerobustcmd*\csepreappto[1]{%
3078    \csname itf@\ifcsname#1\endcsname
3079       preappto\else defpreapp\fi\expandafter\endcsname
3080                                \csname#1\endcsname
3081 }% \csepreappto
3082 \providerobustcmd*\csgpreappto{\global\cspreappto}
3083 \providerobustcmd*\csxpreappto{\global\csepreappto}
```

\removefrom
\eremovefrom
\gremovefrom
\xremovefrom

```
3084 \providerobustcmd\removefrom[2]{\in@false
3085    \long\def\itf@tp##1\itf@tp{%
3086       \in@true
3087       \edef#1{\unexpanded{##1}}%
3088    }\expandafter\itf@removefrom\expandafter{#1}{#2}%
3089 }% \removefrom
3090 \providerobustcmd\eremovefrom[2]{\begingroup
3091    \edef\x{\endgroup\removefrom\noexpand#1{#2}%
3092    }\x
3093 }% \eremovefrom
3094 \providerobustcmd\gremovefrom[2]{%
3095    \removefrom{#1}{#2}%
3096    \global\let#1=#1%
3097 }% \gremovefrom
3098 \providerobustcmd\xremovefrom[2]{%
3099    \eremovefrom{#1}{#2}%
3100    \global\let#1=#1%
3101 }% \xremovefrom
```

\itf@removefrom

```
\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@@
}% \itf@removefrom
```

\removefromstart
\eremovefromstart
\gremovefromstart
\xremovefromstart

```
\newrobustcmd*\removefromstart[2]{\in@false
   \def\itf@tp##1\itf@tp{%
      \in@true
      \edef#1{\unexpanded{##1}}%
   }\expandafter\itf@removefromstart\expandafter{#1}{#2}%
}% \removefromstart
\providerobustcmd*\eremovefromstart[2]{\begingroup
   \edef\x{\endgroup\removefromstart\noexpand#1{#2}%
   }\x
}% \eremovefromstart
\providerobustcmd*\gremovefromstart[2]{%
   \removefromstart{#1}{#2}%
   \global\let#1=#1%
}% \gremovefromstart
\providerobustcmd*\xremovefromstart[2]{%
   \eremovefromstart{#1}{#2}%
   \global\let#1=#1%
}% \xremovefromstart
```

\itf@removefromstart

```
\def\itf@removefromstart#1#2{% removes #2 (code) from #1 (hook)
   \def\in@@##1#2##2\in@@{##1\in@\itf@tp##2\in@@@}%
   \def\in@@@##1\in@@@@#2\in@@@{%
      \ifx\in@##1\fi
   }\expandafter\in@@@\in@@#1\itf@tp\in@@@@#2\in@@
}% \itf@removefromstart
```

⟨/etoolbox⟩

22 History

[2010/11/15 v2.1]

- Improved default and blank values for bookmark levels, based on \ttll@⟨*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 convertion `pt` to `mu` (catcodes...)
- Added microtype to the documentation...

[2010/10/20 v1.0]

- First version.

23 References

[1] *The docstrip program*; 2009/09/25 v2.5d; CTAN:macros/latex/base/.

[2] *some packages...*

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 LaTeX.

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-TeX `\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...

24 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

\# 103, 111, 120, 689, 701
\& 3059
\@@input 2447
\@addtoreset 540, 548, 586, 588
\@break@tfor 193
\@cftmakeloftitle␣(patch) 1763
\@cftmakelottitle␣(patch) 1763
\@cftmaketoctitle␣(patch) 1763
\@cftpnumoff 1619, 1745
\@cftpnumon 1618, 1744
\@clsextension 2418
\@defaultunits 147, 148, 199, 214, 736, 2988, 2994, 3000, 3018
\@elt 1005
\@expandtwoargs 193, 208, 209
\@filef@und 2447
\@filelist 2372, 2377
\@firstofone 1866, 2141
\@fourthoffive 929
\@ifclassloaded 2422
\@ifpackageloaded 2421
\@input␣(patch) 2445
\@mainaux 774, 888, 893
\@nobreakfalse 994
\@onelevel@sanitize 2027
\@onlypreamble 52, 2366
\@pkgextension 2418
\@tfor 191, 2591, 2630, 2644, 2657, 2674, 2740, 2759, 2774, 2796, 2814, 2833, 2853, 2871, 2891, 2908, 2931, 2950, 2968, 2991, 3015
\@whilenum 1876, 1885
\\bookmark␣(patch) 773
\~ 1857

A

\active 1830, 1861
\addcontentsline␣(patch) 688
\AddLineBeginMainAux 420
\addtocontents 1641, 1671, 1675
\adl@arrayrestore 76, 83
\adl@endarray 76, 83
\AfterFile 427, 1611, 2429
\aftergroup 843, 884, 1820
\AfterPackage 55, 68, 72, 89, 246, 287, 684, 940, 2140, 2393, 2409
\allpgfmodules 298, 386, 403
\alltikzlibraries 308, 387
\alltikzpackages 293, 382, 385, 400
\appendices 2324
\apptocmd 98, 118, 1784

B

\BeforeFile 423, 1607, 2429
\BeforePackage 38, 2403
\BKM@@ref 931
\BKM@color 805, 815, 824, 836
\BKM@dest 927, 929
\BKM@page 911
\bookmark 6, 773, 776, 779, 898, 915, 1805
\BookmarkAtEnd 894
\bookmarkget 852, 854, 862, 864
\bookmarksetup . . . 6, 719, 721, 723, 811, 830, 857, 867
\botmarks 1447, 1466, 1468
\bottomtitles 5, 729
\bottomtitlespace 739
\BS@BKM@FontStyle 807, 819, 827, 837, 850, 860

C

\C 187, 1887, 1894, 1968, 2089
\c 1887, 1889, 2092
\c@lofdepth 1671
\c@lotdepth 1675
\c@tocdepth 1640, 1641
\cftdot 1525
\cftdotfill 1727
\cftdotsep 1522, 1730
\cftnodots 1524, 1735, 1736
\changefont 3, 243, 271
\chapter 1020, 1690, 2347
\color 232, 233, 274, 278, 1090, 1110, 1208, 1217, 1226, 1235, 1244, 1253, 1262, 1271, 1280, 1289, 1298, 1307, 2686, 2710, 2712
\colorlet 229
\column 1887, 1968, 1999
\columnsep 1542, 1571
\columnseprule 1541, 1566
\columnseprulecolor 1543
\contentsname 1600, 1645, 1646, 1764, 1820
\count 147
\countdef 1868, 1869, 1870
\countvalues 11, 1832
\csepreappto 2500, 3070
\csgappto 2156, 2428
\csgpreappto 3070
\cspreappto 3070
\csxpreappto 3070
\CT@cell@color 2142, 2143
\CT@cellc 2140
\CT@color 2142
\CT@end 70, 76
\currentgrouplevel 809, 842

D

\default@color 233
\DefineVerbatimEnvironment 2291, 2293, 2295
\diaghead 2135
\dimen 2214, 2215, 2216, 2223, 2232, 2234, 2237, 2238, 2243, 2251, 2254
\dimendef 2225, 2226, 2227
\dump 115

E

\embedfile 2367, 2386
\embedthisfile 17, 2362, 2365, 2366
\endarray 68
\endlinechar 425, 429
\epreappto 3062
\eremovefrom 3084
\eremovefromstart 3108

F

\f@baselineskip 258, 261, 266
\f@linespread . . 258, 261, 264, 265, 266, 267, 268, 269
\f@size 258, 261, 266
\fancy@setoffs . . . . .

\itf@ttl@getformat 451
\itf@ttl@getspacing 451
\itf@ttl@newsection 459, 578
\itf@ttl@sectionformat 450
\itf@ttl@setkey 467, 471, 473, 474, 475, 476, 477, 488, 489, 490, 491, 492
\itf@ttl@setup 451
\itf@ttl@titleclass 526, 586, 588
\itf@ttl@titleclass@i 529, 530, 532, 544
\itf@ttl@titleclass@ii 540, 545, 558
\itf@ttl@update 454, 494
\itf@undefinedpagestyle . . . 1058, 1062, 1065, 1068
\itf@units@mu 184, 1521, 1732
\itf@units@mutopt 171, 212
\itf@units@normalize 225
\itf@units@pt 203
\itf@units@pttomu 171, 197
\itf@units@scale 217, 2838, 2857, 2876, 2895, 2917, 2937, 2955, 2976, 2998, 3023
\itf@units@scale@normalize 219, 220
\itf@usetikzcode 389, 398

L

\label 1816
\lastchapterbookmark 1023
\lastchaptertitle 1021
\lastchaptertoc 1022
\lastsection 955, 965
\lastsectionname 992, 993, 998, 1014
\listfigurename 1657, 1658, 1771
\listofsetup 9, 1488, 1630, 1790
\listtablename 1663, 1664, 1778
\LoadInterface 6, 7, 9, 40
\lofsetup 1488, 1776
\lotsetup 1488, 1783
\ltx@clsextension 2420
\ltx@ifclassloaded 2422
\ltx@ifpackageloaded 53, 57, 66, 73, 90, 227, 231, 1532, 2298, 2421
\ltx@pkgextension 2419

M

\makecell 1867
\marks 43, 1423, 1431
\marksthe 10, 633, 1419
\marksthecs 1425
\maxdimen 2192
\muexpr 179
\multicolsep 1539
\muskip 171, 184, 203
\mutoglue 179

N

\newcounter 547
\newlist 2184
\newlistentry␣(patch) 1763
\newmarks 1436
\newmuskip 129
\newskip 128
\newtoks 130
\normalbaselineskip 1141
\normalcolor 1576, 2688, 2714, 2723, 2731

P

\pagesetup 9, 1034, 1078, 1401
\pagestyle 1049, 1057, 1080, 1410
\paragraph 1694
\patchcmd 105, 127, 773
\pdfstringdefDisableCommands 247
\pgfkeys@case@three 2459, 2465, 2553
\pgfkeys@case@three@handleall 2460
\pgfkeys@exp@call 2467, 2510
\pgfkeys@global@temp 2485, 2492, 2501, 2508
\pgfkeys@ifexecutehandler 2456, 2483, 2495, 2499, 2511, 2515, 2564
\pgfkeys@unpack 2458, 2464
\pgfkeysaddvalue␣(bug␣correction) 2483
\pgfkeysEaddvalue 1595, 1601, 1989, 2495
\pgfkeysEsetvalue 2495
\pgfkeysmeaning 2474
\pgfkeysnovalue 1511, 2539, 2573, 2682, 2690, 2698, 2705
\pgfkeysvalueof@unexpanded 2479, 2633, 2647, 2743, 2836, 2874
\pgfkeysvaluerequired 2544
\pgfmathparse 2018, 2046, 2066
\pgfmathresult 2067
\preappto 3062
\pretocmd . . 97, 118, 947, 957, 1622, 1763, 1770, 1777

R

\R 1877, 1881, 2084
\r 1877, 1879, 2087
\raisedhyperdef 6, 746, 1801
\rem@pt 265
\remove@to@nnil 142, 1837, 1847, 1849, 2024
\removefrom 22, 3084
\removefromstart 1949, 1951, 3108
\ReNewEnviron 2394, 2395
\RenewEnviron 2395
\repeatcell 11, 1854, 2005, 2037, 2040
\row 1877, 2000

S

\scr@load@hook 2424, 2446, 2448
\scr@set@load@hook␣(optimisation) 2428
\sectionformat 4, 436, 721
\selectfont 244
\set@fontsize 258, 261, 266
\setlist 15
\setlist␣(extension,␣star␣form) 2155
\skip 171, 184, 203
\strip@pt 2262
\strut 1076
\subparagraph 1695
\subsubfigure 1697
\subsubsection 1693
\subtable 1699

T

\tab 1856, 1860
\table 1698
\textfont 175, 181
\textheight 736
\the@dimen 2207, 2245, 2254
\thebotmarks 1440
\thefirstmarks 10, 1440
\thepage 907, 924
\thetopmarks 1440
\thispagestyle 1061
\tikzset 289

\titleclass 419, 534, 537, 562
\titleformat 419
\titlespacing 419
\tocloftpagestyle 1502
\tocsetup *8*, *8*, 1488, 1769
\toks 2215, 2221, 2236, 2240, 2251
\toksdef 1871
\topmarks 1452, 1467
\truncate 2195
\TruncateMarker 2190
\trunctext *16*, 2190
\ttl@getkeys 496, 497
\ttl@spacing@ii 519
\ttlh@runinifempty 431
\ttlp@append 415, 421

U

\unless 992, 993
\unskip 2193, 2213, 2233
\usepgfmodule 386, 404
\usetikz *3*, 390, 391, 2046
\usetikzlibrary 376, 387, 405

V

\vsize 2192
\vskip 1100, 1119, 1120

W

\write 774, 888, 893

X

\xpreappto 3062
\xremovefrom 3084
\xremovefromstart 3108