

The **interfaces*** package

interfaces with keys for other packages (FC)

<florent.chervet at free.fr>

2010/10/27 – version 1.2

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 **scrfile** for its different modules are loaded \AfterPackage. **e-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-scrfile**, **interfaces-etoolbox**² and **interfaces-pgfkeys**: those are the *system-layer* of **interfaces**.

~~~~~ Example ~~~~

```
\pagestyle[document]{%
    rules=.5pt,
    foot/left/odd=\arabic{page},
    foot/right/even=\arabic{page}/\pageref{LastPage},
    head/font=\color[gray]{.85}\bfseries\small,
    head/left/even=\leftmark,
    head/left=\thefirstmarks{MiniSection},% interfaces-marks
}
\pagestyle[plain]{% for chapters...
    pagestyle=document,% preloaded settings
    head=,
    foot/left/odd = \roman{page},
    norules, head/rule=.4pt
    head/center=This is the title,
}
\sectionformat{section[hang]}{
    label=\arabic{section},
    labelsep=,
    font=\bfseries\Large,
    bookmark={bold,color=blue},% interfaces-bookmark
    top=6pt plus3pt minus2pt,
    bottom=3pt plus2pt,
    break=\marksthe{section}{lastsectiontitle},% interfaces-hypmsec
    mark=\marksthe{section}{#1},% interfaces-marks
}
\sectionformat{subsection[hang]}{
    label=, labelsep=,
    font=\bfseries,
    bookmark=italic,
    top*=1.2,% a 120% scaling
    bottom+=0pt plus4pt,% add to previous value
    mark=\markright{#1},
}
\sectionformat{\subsubsection[hang]}{
    label=, labelsep=,
    font=\bfseries,
    bookmark=italic,
    top*=1.2,% a 120% scaling
    bottom+=0pt plus4pt,% add to previous value
    mark=\markright{#1},
}
\sectionformat{\subsubsubsection[hang]}{
    label=, labelsep=,
    font=\bfseries,
    bookmark=italic,
    top*=1.2,% a 120% scaling
    bottom+=0pt plus4pt,% add to previous value
    mark=\markright{#1},
}
\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,
    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 L^AT_EX	.	.	.	3	21	Implementation	.	.	.	27
2	Interface TikZ	.	.	.	5	21.1	Interfaces	.	.	.	27
3	Interface titlesec: \sectionformat	.	.	.	5	21.1.1	Requirements	.	.	.	
4	Interface hyperref	.	.	.	7	21.1.2	The package options / list of interfaces	.	.	.	
5	Interface bookmark	.	.	.	7	21.2	Interfaces-base	.	.	.	27
6	Interface hypbmsec	.	.	.	8	21.2.1	Bug Fixes	.	.	.	
7	Interface tocloft: \tocsetup	.	.	.	9	21.2.2	Interfaces patches	.	.	.	
8	Interface fancyhdr: \pagesetup	.	.	.	10	21.2.3	Helper macros	.	.	.	
9	Interface marks	.	.	.	11	21.2.4	Numbers	.	.	.	
10	Interface makecell: \repeatcell	.	.	.	12	21.2.5	Units and math units	.	.	.	
	10.1 Conversion to scaled points	.	.	.	13	21.2.6	Default color	.	.	.	
	10.2 Variable length cell	.	.	.	13	21.3	Interface L^AT_EX	.	.	.	33
	10.3 Empty rows	.	.	.	14	21.3.1	\changefont	.	.	.	
	10.4 color palette	.	.	.	14	21.3.2	\rowfont	.	.	.	
	10.5 tabularcalc-like tables (with dcolumn package)	.	.	.	14	21.4	Interface TikZ	.	.	.	38
	and pgfmath	.	.	.	14	21.4.1	Catecode verifications	.	.	.	
	10.6 Double entry array: the Body Mass Index	.	.	.	15	21.4.2	List of pgf/TikZ packages and libraries	.	.	.	
11	Interface enumitem	.	.	.	16	21.4.3	pgfkeys definitions for \usetikz	.	.	.	
12	Interface truncate	.	.	.	17	21.5	Interface titlesec: \sectionformat	.	.	.	40
13	Interface appendix	.	.	.	18	21.5.1	titlesec modifications	.	.	.	
14	Interface embedfile: \embedthisfile	.	.	.	18	21.5.2	\sectionformat	.	.	.	
15	Interface environ	.	.	.	18	21.5.3	Keys for \sectionformat	.	.	.	
16	Interface umrand	.	.	.	19	21.5.4	\bottomtitles	.	.	.	
17	Interface pgfkeys	.	.	.	20	21.6	Interface hyperref	.	.	.	46
	17.1 /.blank	.	.	.	20	21.7	Interface bookmark	.	.	.	46
	17.2 /.is private	.	.	.	20	21.7.1	star keys* definitions	.	.	.	
	17.3 /.store in cs, /.estore in cs	.	.	.	20	21.7.2	ref and pageref keys	.	.	.	
	17.4 Boolean keys: /.true, /.false, /.on /.off	.	.	.	20	21.8	Interface hypbmsec	.	.	.	50
	17.5 /.is multiple choice	.	.	.	20	21.8.1	interfaces-hypbmsec	.	.	.	
	17.6 /.is enum	.	.	.	20	21.8.2	interfaces-hypbmsec.code.tex	.	.	.	
	17.7 /.store as code	.	.	.	21	21.9	Interface fancyhdr: \pagesetup	.	.	.	52
	17.8 /.store as code in cs	.	.	.	21	21.9.1	\pagesetup	.	.	.	
	17.9 /.store as color	.	.	.	21	21.9.2	Keys for \pagesetup	.	.	.	
	17.10 /.store as color in cs	.	.	.	21	21.10	marks	.	.	.	57
	17.11 /.store as number	.	.	.	21	21.11	Interface tocloft: \tocsetup	.	.	.	59
	17.12 /.store as number in cs	.	.	.	21	21.11.1	Specific handlers for tocloft	.	.	.	
	17.13 /.store as glue	.	.	.	21	21.12	Interface makecell: \repeatcell	.	.	.	65
	17.14 /.store as dimen	.	.	.	21	21.12.1	countvalues	.	.	.	
	17.15 /.store as glue in cs	.	.	.	21	21.12.2	\repeatcell	.	.	.	
	17.16 /.store as dimen in cs	.	.	.	21	21.12.3	colortbl: patch for \CT@cellc	.	.	.	
	17.17 /.estore as dimen in cs	.	.	.	22	21.13	Interface enumitem	.	.	.	71
18	Interface etoolbox	.	.	.	22	21.14	Interface truncate	.	.	.	72
	18.1 New macros provided with \providecommand	.	.	.	22	21.15	interface fancyvr	.	.	.	74
19	Interface scrfile	.	.	.	22	21.16	Interface appendix	.	.	.	74
	19.1 Star forms for \BeforeFile and \AfterFile	.	.	.	22	21.17	Interface embedfile: \embedthisfile	.	.	.	75
	19.2 Allowing \BeforeFile and \AfterFile when @input is invoked	.	.	.	23	21.18	Interface environ	.	.	.	76
20	List of corrected or patched commands	.	.	.	23	21.19	Interface umrand	.	.	.	76
	20.1 LaTeX	.	.	.	23	21.20	Interface scrfile	.	.	.	76
	20.2 array	.	.	.	23	21.21	Interface pgfkeys	.	.	.	77
	20.3 bookmark	.	.	.	23	21.21.1	interfaces-pgfkeys	.	.	.	
	20.4 colortbl	.	.	.	23	21.22	Interface etoolbox	.	.	.	90
	20.5 hypbmsec	.	.	.	24	22	History	.	.	.	91
	20.6 enumitem	.	.	.	24	[2010/10/27 v1.2]	91
	20.7 pgfkeys	.	.	.	24	[2010/10/20 v1.0]	92
					25	23	References	.	.	.	92
					25	README	92
					26	24	Index	.	.	.	93

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 thanks to \pdfstringdefDisableCommands.

```
\rowfont[<alignment>]{<font specification>}
```

\rowfont modifies the font for each cell in a row of a tabular or array.

\rowfont specifications have priority over column specifications, exactly like \rowcolor has priority over \columncolor in the colortbl package.

The implementation must be compatible with all packages that are compatible with the array package, which is required (\rowfont is not defined if the standard package array.sty is not loaded). In particular, \rowfont compatibility has been tested with success with colortbl, hhline, arydshln, tabularx, longtable, delarray, dcolumn, ragged2e...

The alignment of each cell in one row can also be changed to:

l	= left	or for ragged2e settings: L
c	= center	or for ragged2e settings: C
r	= right	or for ragged2e settings: R
j	= justify	or for ragged2e settings: J

Any other value for the optional <alignment> parameter is silently ignored. If ragged2e is not loaded, **L R C** and **J** are synonymous with the lowercase equivalent.

~~~~~ EXAMPLE WITH PACKAGE COLORTBL ~~~~~

```
\begin{tabular}{|l|c|c|c|c|} \hline
\begin{columncolor}{Salmon}\itshape\mdseries c|lcr|} \hline
\rowcolor{GreenYellow}\rowfont{\bfseries}  

<----- type -----> &-- center --> &-- left --> &-- center --> &-- right --> \\ \hline
normal &1&2&3&4&\\
\rowcolor{LightSkyBlue}% overwrites \columncolor
\rowfont[1]{\itshape}  

to left &1&2&3&4&\\
\rowfont[c]{\sffamily\bfseries}% overwrites \mdseries
to center &1&2&3&4&\\
\rowcolor{MediumSpringGreen}
\rowfont[r]{\large\upshape\bottomstrut}% overwrites \itshape
to right &1&2&3&4&\\
\end{tabular}
```

| <---- type ----> | <- center -> | <- left -> | <- center -> | <- right -> |
|------------------|--------------|------------|--------------|-------------|
| normal           | 1            | 2          | 3            | 4           |
| to left          | 1            | 2          | 3            | 4           |
| <b>to center</b> | <b>1</b>     | <b>2</b>   | <b>3</b>     | <b>4</b>    |
| to right         | 1            | 2          | 3            | 4           |

A very bad looking example !

Limitations for the alignment parameter:

- The alignment parameter is essentially provided to format the first title line of the tabular in “one shot” and in simple cases. Otherwise, prefer `\thead` (package `makecell`) for each title cell.
  - be aware that changing the alignment with `\rowfont` in a row whose cells contains tabulars or arrays will give poor results (but `\multicolumn`, `\makecell` and `\repeatcell` have their own alignment parameter).
  - Changing the alignment with `\rowfont` in mathematical mode (*ie.* in arrays) does not work presently
  - Avoid the `ragged2e` settings if you don’t know the package `ragged2e` very well. It contains a lot of parameters for paragraph glues and indentations...

EXAMPLE FOR PARAGRAPH COLUMNS (WITH PACKAGE COLORCTL AS WELL)

| Initial    | m column                                                                                                                          | \raggedright m column                                                                                                               | \centering m column                                                                                                               | \raggedleft m column                                                                                                               |
|------------|-----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| normal     | This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially justified</b>        | This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially raggedright</b>        | This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially centering</b>        | This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially raggedleft</b>        |
| to left    | <i>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially justified</b></i> | <i>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially raggedright</b></i> | <i>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially centering</b></i> | <i>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially raggedleft</b></i> |
| to center  | <b>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially justified</b></b> | <b>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially raggedright</b></b> | <b>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially centering</b></b> | <b>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially raggedleft</b></b> |
| to right   | <i>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially justified</b></i> | <i>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially raggedright</b></i> | <i>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially centering</b></i> | <i>This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially raggedleft</b></i> |
| to justify | This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially justified</b>        | This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially raggedright</b>        | This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially centering</b>        | This is a paragraph, not too long and not excessively short, to see what happens with <b>m columns initially raggedleft</b>        |

EXPLANATIONS FOR LATEXEN

array defines the preamble of each tabular with a pseudo code like:

\hspace{...}\d@llarbegin \the\toks i \ignorespaces \unskip \the\toks i+1 \relax\d@llarend... &  
where \toks i and \toks i+1 contain user defined commands placed before (left side of) and after (right side of) each column specification,  
for example:

$\overbrace{\{ \backslash b f s e r i e s \backslash c o l o r { r e d } \} \quad r \quad \{ \backslash c o l o r { b l a c k } \}, \backslash \$\}}$   
 $\quad \quad \quad \backslash t o k s < i > \quad \quad \quad \backslash t o k s < i + 1 >$

There are also other types of “pseudo code” for **p** and **m** columns, and for special **@** and **!** columns but the implementation is the same. For more details, please refer to the implementation part.

`interfaces` append to each such `\toksi` a control sequence named `\itf@cellleft` and to each such `\toksi + 1` a control sequence named `\itf@cellright`, similar to `\@arrayleft` and `\@arrayright` for the whole tabular. Those control sequences can be globally modified in order to change the font (with `\itf@cellleft`) in each cell in a row. `\itf@cellright` is also used when modifying the alignment for the row.

Both `\itf@cellleft` and `\itf@cellright` are equal to `\@empty`, unless `\rowfont` is used. Therefore,

This implementation is fair, and compatible with `colortbl`, `tabularx`, `arydshln`, `hhline`, `longtable`, `delarray`, `dcolumn` and certainly any package compatible with `array`.

However, you must know that `arydshln` does not like special @ and ! columns. If you encounter problems with @ or ! columns while `arydshln` is loaded (that is, the vertical lines are not correctly positionned), it's not the fault of [interfaces...](#)

## 2 Interface TikZ (☞ pgfmanual )

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

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

`\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 subtract and the `*` key scales.

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

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

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

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

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

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

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

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

|                              |                       |                                                                                                                  |
|------------------------------|-----------------------|------------------------------------------------------------------------------------------------------------------|
| <code>bookmark/bold</code>   | bold bookmark text    | the text of the bookmark for that sectioning command will be <b>bold</b>                                         |
| <code>bookmark/italic</code> | italic bookmark text  | the text of the bookmark for that sectioning command will be in <i>italic</i>                                    |
| <code>bookmark/color</code>  | colored bookmark text | the text of the bookmark for that sectioning command will be <b>colorized</b>                                    |
| <code>bookmark/off</code>    | disable the bookmark  | no bookmark for that sectioning command alternatively you may set <code>bookmark/level</code> to a huge value... |
| <code>bookmark/style</code>  | use a <i>style</i>    | the <i>style</i> must be previously declared with <code>\bookmarkdefinenestyle</code>                            |
| <code>bookmark/level</code>  | level of the bookmark | defines <code>\toclevel@{sectionname}</code> , 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 ascertain that there is no bookmark between the one corresponding to the section and the one corresponding to its first subsection, then this is the same as: “`after=\bookmarksetup{color*=red}`”.

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`:

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

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}
```

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

Where:    `[toc]` for the table and contents and the marks  
           `{title}` to typeset the section title in the document

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

`interfaces` takes advantage of this and defines:

|                                   |                                                                                 |
|-----------------------------------|---------------------------------------------------------------------------------|
| <code>\lastsectiontitle</code>    | expands to the title of the last <code>\section</code>                          |
| <code>\lastsectionbookmark</code> | expands to the text of the bookmark for the last <code>\section</code>          |
| <code>\lastsectiontoc</code>      | expands to the entry of the last <code>\section</code> in the table of contents |

Macros for each sectionning command are defined:

|                                  |                                     |                                      |
|----------------------------------|-------------------------------------|--------------------------------------|
| <code>\lastparttitle</code>      | <code>\lastchaptertitle</code>      |                                      |
| <code>\lastsectiontitle</code>   | <code>\lastsubsectiontitle</code>   | <code>\lastsubsubsectiontitle</code> |
| <code>\lastparagraphtitle</code> | <code>\lastsubparagraphtitle</code> |                                      |

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

Assignment to those macros is automatic. More precisely, `\lastsectiontitle` is defined **after** `\sectionbreak` and **before** `\sectionmark`: this means that in the fields `after` or `mark` of `\sectionformat`, `\lastsectiontitle` has been updated yet. While in the fields `before`, and `break`, `\lastsectiontitle` is not yet updated. (For chapters, this is the same if `titlesec` formats the chapter titles, *i.e.* if `\sectionformat\chapter` or `\titleformat\chapter` has been invoked. Otherwise, `\lastchaptertitle`, `\lastchapterbookmark` and `\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 11 under `\ifmarksequal`). With this feature, when a new `\section` is created, `\lastsubsection` is reset to `\relax` therefore, `\marksthecs{\subsection}{\lastsubsection}` will not mark anything.

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

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

```
\tocsetup{ depth = \setcounter{tocdepth} This is a LATEX rather than tocloft's ...
           title | contentsname = redefines \contentsname
           title/top | title/top+ | title/top- | title/top* = vertical space before the title (set | add | subtract | scale)
           title/bottom | title/bottom+ etc. = vertical space after the title
           title/font | title/font* = 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  $\approx 1em/18$  is supplied. Default value is 4.5mu and blank value is the same as
                     \cftnodots
           dot = modifies \cftdot, the dot used for leaders produced by \cftdotfill
           pagestyle = page style for the first page (\tocloftpagestyle)
           mark = the mark command: \cftmarktoc
                   example: \chaptermark\contentsname
           before | before* | after | after* = hooks to execute before or after the command \tableofcontents
           twocolumns | columns | multicol = twocolumns=true is the same as columns=2 Makes a multicolumn table
                     of contents. Package multicol is required!
           columns/sep | columns/sep+ | columnsep etc. = sets (or add) a <dimen> to \columnsep
           columns/rule | columns/rule+ | columnseprule etc. = sets (or add) a <dimen> to \columnseprule
           columns/color | columnseprulecolor = modifies \columnseprulecolor
           multicolsep | multicolsep+ etc. = modifies (or add) a <glue> to \multicolsep
                     \multicolsep is the space added before and after multicol environment. Its default value is 0pt plus 4pt minus 3pt.
           multicols/begin | twocolumns/before = hooks just after \begin{multicols} ...
           multicols/end | twocolumns/after = ... and before \end{multicols}
           label = the label for reference to the table of contents
           ref = the reference printed by \ref when referencing the table of contents
                     (default is \contentsname. In any case, \nameref prints \contentsname.)
           bookmark/text = the bookmark text for the table of contents (if empty, no bookmark is set)
           bookmark/bold | bookmark/italic | bookmark/color = the bookmark style for the table of contents
           bookmark/level = the bookmark level: either a number or section or chapter
                     Default is chapter, or section if the class does not provide chapters. }
```

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

The following keys apply to each section levels. Section level names are : part / chap or chapter / sec or section / subsec or subsection / subsubsec or subsubsection / para or paragraph / 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* | aftersnum = code before the title (ie. after the section number box)
           ¶/leader | ¶/leaders = modifies the leader: the default is \cftdotfill{\cftdotsep}
           ¶/dotsep = the space (in mu) between dots in the leader. Default is \cftdotsep and blank value
                     is the same as \cftnodots
           ¶/pagefont | ¶/page/font | ¶/pagenumber/font = modifies the font used to typeset the page number
           ¶/page/after | ¶/pagenumber/after | afternum = 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 : \pagestyle

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

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

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

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

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

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

9 Interface marks

`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” L^AT_EX `\markboth`, `\markright`, `\leftmark` and `\rightmark`. Who gets the correct – I mean *expected* – result with such commands ? Not me, I can say !

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

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

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

`\marksthecs{\{subsubsection\}}{\{cs-name\}}` will similarly mark the unexpanded content of `\{cs-name\}` by means of the marks register `\marks@{\{subsubsection\}}`. `\{cs-name\}` is really the ***name of the control sequence***, not the control sequence itself. If `\{cs-name\}` is undefined or `\relax`, nothing is marked, making `\marksthecs` suitable to use with `\lastsectontitle` etc.. at `\sectionbreak` (see `interfaces-hypbmsec` p. 8).

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

`\thefirstmarks{\{chapter\}}` expands to `\firstmarks\marks@{\{chapter\}}` (this is expandable), *i.e.* 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\}}` *i.e.* 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 (*i.e.* nothing is done).

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

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

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

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

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

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

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

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

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

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

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

10 Interface ↗ `makecell` : `\repeatcell`

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

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

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

- `\numexpr`,
- `\pgfmath` (\sim `\usetikz{pgfmath}` required in preamble)
- `$$\FPeval$$` (`\usepackage{fp}` required in preamble) \Rightarrow 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 `\crcrcr\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 $\{1, 2, 3, 4, \dots, p\}$.

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

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

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

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

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

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

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

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<br>scaled<br>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    | Mother | Father | Children                   |
|---------|--------|-------------|--------|--------|----------------------------|
|         |        | Priscilla   |        |        | Isis                       |
| Johanne | Georg  | John<br>Tom | Nout   | Geb    | Osiris<br>Seth<br>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   = \blueii,
text/row1=\cellcolor[RGB]{\red,\greeni,\bluei}(\red,\greeni,\bluei),
text/row2=\cellcolor[RGB]{\red,\greenii,\blueii}(\red,\greenii,\blueii),
text/row3=\cellcolor[RGB]{\red,\greeniii,\blueiii}(\red,\greeniii,\blueiii),
text/row4=\cellcolor[RGB]{\red,\greeniv,\blueiv}(\red,\greeniv,\blueiv)} }
\end{tabular}
```

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

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

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

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

|                         |        |         |      |        |         |
|-------------------------|--------|---------|------|--------|---------|
| $x$                     | -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    |
| $h(x) = \sqrt{x^2 + 1}$ | 4.1231 | 2.23605 | 1.0  | 2.4622 | 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{#1}}
\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\fi}

\begin{tabular}{*{\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{BIM}\BIM,
  text/row1 = \c, font/row1 = \color{NavyBlue}, cr/row1 = \\ \hline,
  text/coll1 = \r, font/coll1 = \color{DarkViolet},
} \\ \hline
\end{tabular}\quad
\begin{tabular}{|c|} \hline
\repeatcell{denutrition,skinniness,normal,obesity,serious obesity,morbid obesity}{%
  text=\cellcolor{\c}{\c}, transpose, font=\large\bfseries\bottomstrut}
\end{tabular}
```

*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  $\mathcal{AM}\mathcal{S}$  `\intertext` in math formulæ, `\interitem` allows to put some non intended text inside a list environment. Example:

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

The example give in real world:

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

And the following follows...

## 12 Interface ⇔ truncate

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

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

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

```
\trunc{⟨TruncateMarker⟩}{⟨width⟩}{⟨text⟩}
```

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

```
truncate: \truncate{⟨5cm⟩}{⟨Hello world!⟩}
trunc: \trunc{⟨5cm⟩}{⟨Hello world!⟩}
```

Hello word!

Hello world!

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

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

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

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

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

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

```
\trunc{⟨TruncateMarker⟩}{\htruncwidth{⟨i⟩}}{⟨texti⟩}.
```

$x_i$  can be any real number provided that  $\sum_i x_i \leq 16383 = \text{\maxdimen}$  in points.

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

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

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

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

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

```
\htruncadjust{3cm}{first quite long,second short,third} % natural widths
\trunc{⟨htruncwidth{1}⟩}{first quite long}
\trunc{⟨htruncwidth{2}⟩}{second short}
\trunc{⟨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 disappear 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 ~~~~~

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

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

### 17.1 Handler definition: `/.blank`

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

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

The following make blank value behave the same as default :

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

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

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

With `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:

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

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

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

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

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

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

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

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

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

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

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

### 17.11 † Handlers definition: `./store as number, ./estore as number`

### 17.12 † Handlers definition: `./store as number in cs, ./estore as number in cs`

### 17.13 † Handler definition: `./store as glue`

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

Furthermore:

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

the `-` key subtracts: `\glueexpr old value - < argument > \relax`

the `*` key scales: `\glueexpr old value * < argument > \relax`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

## 18 Interface ↗ etoolbox - etoolbox-like commands

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

### 18.1 † New macros provided with `\providetcommand`

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

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

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

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

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

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

this allows loops of the form:

`\in@true`

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

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

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

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

## 19 Interface ↗ scrfile

### 19.1 † Star forms for `\BeforeFile` and `\AfterFile`

`scrfile` 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 **scrfile**.

**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 † L<sup>A</sup>T<sub>E</sub>X

|   | Patched command                                                        | Type | Importance                          | Module           | ref. page |
|---|------------------------------------------------------------------------|------|-------------------------------------|------------------|-----------|
| ❶ | \@pkgextension<br>\@clsextension<br>\ifpackageloaded<br>\ifclassloaded |      | minor<br>optimisation or correction | interfaces-LaTeX | 76        |

Those commands are replaced by their equivalent in the package **ltcmds**:

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

This is because L<sup>A</sup>T<sub>E</sub>X defines those commands \onlypreamble, while the equivalent macros in **ltcmds** can be used inside **document** as well.

|   | Patched command  | Type    | Importance                      | Module                                   | ref. page |
|---|------------------|---------|---------------------------------|------------------------------------------|-----------|
| ❷ | \addcontentsline | prepend | important<br>but no side effect | interface-titlesec<br>interface-bookmark | 45        |

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

|   | Patched command | Type | Importance                      | Module            | ref. page |
|---|-----------------|------|---------------------------------|-------------------|-----------|
| ❸ | \@input         |      | important<br>but no side effect | interface-scrfile | 77        |

The \input primitive (saved by L<sup>A</sup>T<sub>E</sub>X in control sequence \@@input) is surrounded by \scr@load@hook{before}... \scr@load@hook{after}:

Original:

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

Modified:

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

## 20.2 † array

|   | Patched command            | Type           | Importance                      | Module                           | ref. page |
|---|----------------------------|----------------|---------------------------------|----------------------------------|-----------|
| ④ | <code>\prepnext@tok</code> | <i>prepend</i> | important<br>but no side effect | <code>interfaces-makecell</code> | 34        |

`\prepnext@tok` is replaced by `\itf@prepnext@tok` for the purpose of inserting `\itf@cell1align` and `\itfcellleft` at the left edge of each cell in a tabular and `\itf@cellright \itf@cellralign` at the right edge of each cell.

This allows `\rowfont` to change the font (and optionally the alignment) for all cells in a tabular or array row.

## 20.3 † bookmark \\bookmark

|   | Patched command        | Type         | Importance | Module                          | ref. page |
|---|------------------------|--------------|------------|---------------------------------|-----------|
| ⑤ | <code>\bookmark</code> | <i>modif</i> | important  | <code>interface-bookmark</code> | 46        |

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

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

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

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

## 20.4 † colortbl

|   | Patched command                                  | Type         | Importance                      | Module                          | ref. page |
|---|--------------------------------------------------|--------------|---------------------------------|---------------------------------|-----------|
| ⑥ | <code>\CT@cellc</code><br><code>\endarray</code> | <i>corr.</i> | important<br>but no side effect | <code>interface-makecell</code> | 71        |

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

Original:

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

Modified:

```
\def\CT@cellc#1[#2]#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).

## 20.5 † hypbmsec

|   | Patched command                                   | Type           | Importance                      | Module                                                               | ref. page |
|---|---------------------------------------------------|----------------|---------------------------------|----------------------------------------------------------------------|-----------|
| ⑦ | <code>\hbs@do</code><br><code>\hbs@process</code> | <i>prepend</i> | important<br>but no side effect | <code>interfaces-hypbmsec</code><br><code>interfaces-titlesec</code> | 50        |

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

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

## 20.6 † `enumitem`

|   | Patched command       | Type         | Importance                       | Module                           | ref. page |
|---|-----------------------|--------------|----------------------------------|----------------------------------|-----------|
| ⑧ | <code>\setlist</code> | <i>redef</i> | minor optimisation or correction | <code>interfaces-enumitem</code> | 72        |

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

## 20.7 † `pgfkeys.code.tex`

|   | Patched command                                                                                    | Type         | Importance | Module                          | ref. page |
|---|----------------------------------------------------------------------------------------------------|--------------|------------|---------------------------------|-----------|
| ⑨ | <code>\pgfkeys@unpack</code><br><code>\pgfkeys@case@three</code><br><code>\pgfkeys@exp@call</code> | <i>redef</i> | important  | <code>interfaces-pgfkeys</code> | 79        |

`\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   [2010/10/27 v1.2 - pgfkeys macros for other packages (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   [2010/10/27 v1.2 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}%
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 L<sup>A</sup>T<sub>E</sub>X’s `\@ifpackageloaded` – this macro 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.

```
89 \AfterPackage{colortbl}{%
90   {\ltx@ifpackageloaded{arydshln}%
91    {\PackageWarning{interfaces-base}%
92     {The package arydshln should be loaded after colortbl%
93      \MessageBreak because both redefine \string\@classz}}%
94   {}}%
95 }% 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.

```
96 \newrobustcmd*\itf@preapptocmd[5]{%
97   \pretocmd#1{#2}%
98   {\apptocmd#1{#3}%
99    {#4}%
100   {#5}}%
101   {#5}%
102 }% \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...

```
103 \def\itf@patch{\@makeother\#\futurelet\itf@tp\itf@patch@}
104 \def\itf@patch@{\csname itf@patch@\%
105   \ifx\itf@tp\patchcmd inside\else
106   \ifx\itf@tp\itf@preapptocmd inside\else
107   boundary\fi\fi
108   \endcsname
109 }% \itf@patch@
110 \def\itf@patch@boundary#1#2#3#4#5{%
111   \catcode`\#=6%
112   \let\itf@tp#2%
113   #1\itf@tp{#3}%
114   {\let#2\itf@tp
115     \AtBeginDocument{#4}% in order to keep this info even after \dump
116   }
117   {#5}%
118 }% \itf@patch@boundary (for \pretocmd and \apptocmd)
119 \def\itf@patch@inside#1#2#3#4#5#6{%
120   \catcode`\#=6%
121   \let\itf@tp#2%
122   #1\itf@tp{#3}{#4}%
123   {\let#2\itf@tp
124     \AtBeginDocument{#5}%
125   }
126   {#6}%
127 }% \itf@patch@inside (for \patchcmd)
```

### 21.2.3 Helper macros

```

128 \newskip\itf@sk
129 \newmuskip\itf@musk
130 \newtoks\itf@tk
131 \newcount\itf@cnt

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

132 %%    H E L P E R    M A C R O S
133 \begingroup\makeatother
134   \csgdef{\itf@strip@at}{#1@{}}
135 \endgroup
136 \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\macroptest`

`\macroptest` is expanded once and its first token  $\tau$  is placed into a temporary macro `\itf@tp`.

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

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

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

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

```

137 \def\itf@expandonceifcs{\expandafter\itf@exp@ndonceifcs\expandafter#1#1Q@nnil}
138 \def\itf@exp@ndonceifcs#1#2#3@nnil{%
139   \def\itf@tp{#2}%
140   \ifx\itf@tp#1%
141     \ifcat\relax\noexpand#2%
142       \expandafter\expandafter\expandafter\remove@to@nnil
143         \expandafter\ifx\expandafter#2#2@nnil\else\@nnil\let#1#2\fi
144   \fi\fi
145 }% \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`.

```

146 %%    G E T    A    N U M B E R (if possible)
147 %% USAGE: \itf@number\count<text>\relax@nnil (like \@defaultunits)
148 \def\itf@number#1\relax@nnil{%
149   \relax is to mimick the syntax of \@defaultunits
150   \afterassignment\itf@number@#1\number@#2\relax\itf@number@
151 }
151 \def\itf@number@#1\relax#2\itf@number@{%
152   \ifblank{#1}{%
153     {\itf@number@@{}}#2\itf@number@@@%
154     \csname ltx@\ifx\@let@token\itf@number@@
155       first\else second\fi oftwo\endcsname%
156     \ltx@secondoftwo
157 }% \itf@number@
158 \def\itf@number@@#1{\futurelet\@let@token\itf@number@@@}
159 \def\itf@number@@@{%
160   \csname \ifx\@let@token\relax\itf@number@@%
161   \else remove@to@nnil%
162   \fi\endcsname
163 }% \itf@number@@

```

## 21.2.5 Units and math units

`\itf@strip@mu` This is the equivalent of L<sup>A</sup>T<sub>E</sub>X's `\strip@pt` but for math units:

```

164 %% U N I T S F O R D I M E N S
165 \def\itf@strip@mu{\expandafter\itf@rem@mu\the}
166 {%
167 \@makeother\M\@makeother\U
168 \lowercase{%
169   \gdef\itf@rem@mu#1.#2MU{#1\ifnum#2>\z@.#2\fi}%
170 }}%

```

`\itf@units@pttomu`  $1mu$  is exactly  $1em/18$  where the  $em$  is taken from `\textfont2`.

`\itf@units@mutopt` Thus the scaling: `\fontdimen6\textfont2 * 100/1800`.

```

171 \% \skip = \itf@units@mutopt \muskip
172 \def\itf@units@pttomu#1{\gluetomu\glueexpr
173   \glueexpr#1\relax*1800%
174     \expandafter\itf@strip@dot\the\dimexpr
175       \fontdimen6\textfont2*100\relax.\@nil
176     \relax
177 }% \itf@units@pttomu
178 \def\itf@units@mutopt#1{\glueexpr
179   \mutoglue\muexpr#1\relax*
180     \expandafter\itf@strip@dot\the\dimexpr
181       \fontdimen6\textfont2*100\relax.\@nil/1800
182     \relax
183 }% \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.

```

184 \% \muskip = \itf@units@mu{ \skip or \muskip or number}
185 {%
186 \@makeother\P\@makeother\T\@makeother\S\@makeother\B
187 \@makeother\C\@makeother\M\@makeother\D\@makeother\I\@makeother\N
188 \lowercase{%
189 \gdef\itf@units@mu#1#2{\begingroup
190   \edef\itf@tp{#2}%
191   \atfor\x:={PT}{pt}{PC}{pc}{IN}{in}{BP}{bp}%
192     {CM}{cm}{MM}{mm}{DD}{dd}{CC}{cc}{SP}{sp}\do{%
193       \expandtwoargs\in@\x\itf@tp\ifin@\break@tfor\fi
194     }%
195   \expandafter\endgroup
196   \ifin@
197     #1=\itf@units@pttomu{#2}%
198   \else
199     \defaultunits#1 #2mu\relax\@nnil
200   \fi
201 }% \itf@units@mu
202 }%

```

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

```

203 \% \skip = \itf@units@pt( \muskip or \skip or number )
204 {%
205 \@makeother\M\@makeother\U
206 \lowercase{%
207 \gdef\itf@units@pt#1#2{\begingroup
208   \expandtwoargs\in@\{MU\}{#2}%
209   \ifin@\else\expandtwoargs\in@\{mu\}{#2}\fi

```

```

210  \expandafter\endgroup
211  \ifin@
212    #1=\itf@units@mutopt{\#2}%
213  \else
214    \@defaultunits#1 #2pt\relax\@nil
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 `xcolor`, 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 L<sup>A</sup>T<sub>E</sub>X

```
238 {*\LaTeX}
239 \ProvidesPackage{interfaces-LaTeX}
240   [2010/10/27 v1.2 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.

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

  

```
248 \pgfqkeys{/itf/LaTeX/changefont}{%
249   encoding/.code=\fontencoding{#1},encoding/.value required,%
250     enc/.alias=encoding,%
251   family/.code=\fontfamily{#1},family/.value required,%
252     fam/.alias=family,%
253   series/.code=\fontseries{#1},series/.value required,%
254     ser/.alias=series,%
255   shape/.code=\fontshape{#1},shape/.value required,%
256     sha/.alias=shape,%
257   size/.estore as dimen in cs=f@size,%
258   size/.append code=\set@fontsize\f@linespread\f@size\f@baselineskip,%
259     siz/.alias*=size,%
260   baselineskip/.store as glue in cs=f@baselineskip,%
261   baselineskip/.append code=\set@fontsize\f@linespread\f@size\f@baselineskip,%
262     skip/.alias*=baselineskip,%
263   linespread/.estore as dimen in cs=f@linespread,%
264   linespread/.add code={\ifx\f@linespread\empty\let\f@linespread\ne\fi}%
265     {\edef\f@linespread{\expandafter\rem@pt\f@linespread}%
266      \set@fontsize\f@linespread\f@size\f@baselineskip},%
267   linespread+/.prefix code=\ifx\f@linespread\empty\let\f@linespread\ne\fi,%
268   linespread-/.prefix code=\ifx\f@linespread\empty\let\f@linespread\ne\fi,%
269   linespread*/.prefix code=\ifx\f@linespread\empty\let\f@linespread\ne\fi,%
270     spread/.alias*=linespread,%
271 }% \pgfkeys definitions for \changefont
272 \itf@AfterPackage{color}{%
273   \pgfkeysifdefined{/itf/LaTeX/changefont/color/.@cmd}{}
274     {\pgfkeys{/itf/LaTeX/changefont/color/.code=\color{#1}}}}%
275 }% AfterPackage color
276 \itf@AfterPackage{xcolor}{%
277   \pgfkeysifdefined{/itf/LaTeX/changefont/color/.@cmd}{}
278     {\pgfkeys{/itf/LaTeX/changefont/color/.code=\color{#1}}}}%
279 }% AfterPackage xcolor
```

### 21.3.2 \rowfont

\rowfont \rowfont uses the control sequences \itf@cellalign, \itf@cellleft, \itf@cellright and \itf@cellralign which have been placed on purpose into the user-defined tokens inserted in any preamble by the `array` package.

\itf@cellalign and \itf@cellralign are used to modify the alignment. If the optional [*alignment*] parameter of \rowfont is not specified, then those control sequence expand to \@empty.

\itf@cellleft contains the font-modification information.

Placement of those control sequences into the user-tokens that are inserted in the preamble by the `array` package is explained below under the macro \itf@prepnext@tok.

```

280 \newtoks\itf@everycr@tok % better than colortbl turnaround for \everycr
281 \newcommand*\rowfont{\noalign{\ifnum0='}\fi\itf@rowfont}
282 \newrobustcmd*\itf@rowfont[2][]{%
283   \global\itf@everycr@tok\everycr
284   \global\let\itf@cellleft\itf@cellleft
285   \global\let\itf@cellright\itf@cellright
286   \gdef\itf@cellfont{#2}%
287   \ifcsname itf@cell@#1\endcsname % row alignment
288     \csname itf@cell@#1\endcsname
289   \else\lowercase{%
290     \ifcsname itf@cell@#1\endcsname
291       \csname itf@cell@#1\endcsname
292       \fi
293   }\fi
294   \toks@\expandafter{\itf@cellleft\itf@cellfont}% inside \noalign group ok
295   \xdef\itf@cellleft{\the\toks@}%
296   \global\everycr\expandafter\expandafter\expandafter{%
297     \expandafter\the\expandafter\expandafter\expandafter\itf@rowfont@reset}%
298   \ifnum0='{\fi}% end of noalign group
299 }% \rowfont

```

\itf@rowfont@reset This macro resets \itf@cellalign, \itf@cellleft, \itf@cellright, \itf@cellralign and \everycr to the value they had before the expansion of \rowfont.

It expands when a new row is inserted into the tabular or array.

```

300 \def\itf@rowfont@reset{%
301   \noalign{%
302     \global\let\itf@cellleft\itf@cellleft
303     \global\let\itf@cellright\itf@cellright
304     \global\let\itf@cellfont\@empty
305     \global\let\itf@cellalign\@empty
306     \global\let\itf@cellralign\@empty
307     \global\everycr\itf@everycr@tok
308     \global\itf@everycr@tok{}%
309   }%
310 }% \itf@rowfont@reset

```

\itf@prepnext@tok \itf@prepnext@tok will replace \prepnext@tok (in `array.sty`): its purpose is to add the control sequences \itf@cellalign, \itf@cellleft, \itf@cellright and \itf@cellralign at the right position in the “preamble” for \halign. Those control sequences are not inserted directly into the preamble, but by the means of the user-tokens placed there by the `array` package.

The package `array` defines a macro \prenext@tok to initialize each user-token inserted at both side of each “normal” column. For “special” @ and ! columns, there is only one token.

$$\overbrace{\bfseries \color{red}{r}}^{\text{\scriptsize \itf@prepnext@tok}} \quad \underbrace{< \color{black}{,} >}_{\text{\scriptsize \itf@prepnext@tok}}$$

When a column is inserted in the tabular preamble (`\@preamble`), the T<sub>E</sub>X counter `\count@` is equal to  $i + 1$  (ie. the right token) and the counter `\@tempcnta` is equal to  $i$  (ie. the left token). If the column is special (ie. @ or !) `\@tempcnta` is not updated.

Thus, when a new token is “prepared” by `\prepnext@tok`:

**either: `i = \count@ = \@tempcnta`** : the token to prepare (ie. `\toks< i + 1 >`) is the right one of a “normal” column. The switch `\ifitf@cellright` is set to `true`.

The *previous* token (`\toks< i > = \toks\count@`) is necessarily the left one of this “normal” column: we prepend `\itf@cellalign` and append `\itf@cellleft` to this token (`\toks< i >`). This token is finished and will not change afterwards.

**or: `i = \count@ = \@tempcnta + 1`** : the token to prepare (`\toks< i + 1 >`) is either the left one of a normal column, or the single one of a special @ or ! column.

If the switch `\ifitf@cellright` is true, then the *previous* token `\toks< i >` is the right one of the last inserted column (which was a “normal” column, thus):, `\itf@cellright\itf@cellralign` is appended to it, and the switch `\itif@cellright` is reset to `false`. May be `\prepnext@tok` will be expanded again (by `\save@decl`): if it happens, then again `\count@ = \@tempcnta + 1` (same case) but `\ifitf@cellright` is `false` and nothing is changed.

**else:** The token to prepare (which is `\toks< i + 1 > = \toks\count@ + 1`), cannot be the right one of a “normal” column: `\ifitf@cellright` is set to `false`.

The fact that  $|\count@ - \@tempcnta| > 1$  tells us that the previous token `\toks< i >` is necessarily the single one of a “special” @ or ! column. We don’t modify this token, as long as *special columns are always inserted as is*: `\rowcolor` has no effect on special columns, nor `\rowfont`.

Thereafter, the original initialisation sequence occurs: `\advance\count@ by\@ne` and initialize the token to prepare (`\toks\count@ = \toks< i + 1 >`) to an empty one.

```

311 \newif\ifitf@cellright
312 \def\itf@prepnext@tok{\% only if array is loaded
313   \itf@cnt = \count@
314   \advance \itf@cnt -\@tempcnta
315   \ifcase \itf@cnt          % (case 0): previous token is left
316     \itf@cellrighttrue    % consequently the token to prepare is right !
317     \toks\count@\expandafter{\% augment the previous (left) token
318       \expandafter\itf@cellalign
319       \the\toks\count@
320       \itf@cellleft}%
321   \or                      % (case 1)
322     \ifitf@cellright      % previous token is right
323       \itf@cellrightfalse% consequently the token to prepare is not right !
324       \toks\count@\expandafter{\% augment the previous (right) token
325         \the\toks\count@
326         \itf@cellright
327         \itf@cellralign}%
328   \fi
329   \else % special column: do not change the token
330     \itf@cellrightfalse % the token to prepare cannot be right !
331   \fi
332   \advance \count@ \@ne \toks\count@\{\%\ \prepnext@tok original def
333 }%\itf@prepnext@tok
334 \AfterPackage*{array}{%
335   \global\let\array@prepnext@tok\prepnext@tok % frozen original definition
336   \global\let\prepnext@tok\itf@prepnext@tok
337 }%\AfterPackage*{array}

```

`\itf@cellleft` First initialisation to `\@empty`.

```

\itf@cellalign
\itf@cellright
\itf@cellralign
\itf@cellfont
338 \let\itf@cellleft\@empty
339 \let\itf@cellright\@empty
340 \def\itf@cellalign{\itf@cellleft}\% the row font spec. applies to pre-column material
341 \let\itf@cellralign\@empty

```

```
342 \let\itf@cellfont\@empty
```

\itf@cell@l Setup macros to modify the alignment. The skips inserted to make the standard alignment specified in the tabular preamble are not the same with standard array tabulars and colortbl tabulars, hence the switch \itf@cell@r \ifitf@colortbl.

\itf@cell@j

```
343 \let\ifitf@colortbl\iffalse
344 \AfterPackage*{colortbl}{\let\ifitf@colortbl\iftrue}
345 \def\itf@cell@l{\% force alignment to left
346   \gdef\itf@cellalign{\itf@removehfil
347     \raggedright\arraybackslash
348     \itf@cellleft}%
349   \gdef\itf@cellralign{\itf@flush1\itf@ignorehfil}%
350   \toks@\expandafter{\itf@cellleft\raggedright\arraybackslash}% local (noalign group)
351   \xdef\itf@cellleft{\the\toks@}%
352 }% \itf@cell@l
353 \def\itf@cell@c{\% force alignment to center
354   \gdef\itf@cellalign{\itf@removehfil
355     \centering\arraybackslash
356     \itf@flush{.5}\itf@cellleft}%
357   \gdef\itf@cellralign{\itf@flush{.5}\itf@ignorehfil}%
358   \toks@\expandafter{\itf@cellleft\centering\arraybackslash}% local (noalign group)
359   \xdef\itf@cellleft{\the\toks@}%
360 }% \itf@cell@c
361 \def\itf@cell@r{\% force alignment to right
362   \gdef\itf@cellalign{\itf@removehfil
363     \raggedleft\arraybackslash
364     \itf@flush1\itf@cellleft}%
365   \gdef\itf@cellralign{\itf@ignorehfil}%
366   \toks@\expandafter{\itf@cellleft\raggedleft\arraybackslash}% local (noalign group)
367   \xdef\itf@cellleft{\the\toks@}%
368 }% \itf@cell@r
369 \def\itf@cell@j{\% force justification (for p, m, b columns)
370   \gdef\itf@cellalign{\itf@justify\itf@cellleft}%
371   \global\let\itf@cellralign\@empty
372   \toks@\expandafter{\itf@cellleft\itf@justify}%
373   \xdef\itf@cellleft{\the\toks@}%
374 }% \itf@cell@j
375 \def\itf@justify{%
376   \leftskip\z@skip \rightskip\leftskip \rightskip\@rightskip
377   \parfillskip\@flushglue
378 }% \itf@justify
379 \AfterPackage*{ragged2e}{%
380 \def\itf@cell@L{\% force alignment to left (ragged2e)
381   \gdef\itf@cellalign{\itf@removehfil
382     \RaggedRight\arraybackslash
383     \itf@cellleft}%
384   \gdef\itf@cellralign{\itf@flush1\itf@ignorehfil}%
385   \toks@\expandafter{\itf@cellleft\RaggedRight\arraybackslash}%
386   \xdef\itf@cellleft{\the\toks@}%
387 }% \itf@cell@L
388 \def\itf@cell@C{\% force alignment to center (ragged2e)
389   \gdef\itf@cellalign{\itf@removehfil
390     \Centering\arraybackslash
391     \itf@flush{.5}\itf@cellleft}%
392   \gdef\itf@cellralign{\itf@flush{.5}\itf@ignorehfil}%
393   \toks@\expandafter{\itf@cellleft\Centering\arraybackslash}%
394   \xdef\itf@cellleft{\the\toks@}%
395 }% \itf@cell@C
396 \def\itf@cell@R{\% force alignment to right (ragged2e)
397   \gdef\itf@cellalign{\itf@removehfil
398     \RaggedLeft\arraybackslash
399     \itf@flush1\itf@cellleft}%

```

```

400   \gdef\itf@cellralign{\itf@ignorehfil}%
401   \toks@\expandafter{\itf@cellleft\RaggedLeft\arraybackslash}%
402   \xdef\itf@cellleft{\the\toks@}%
403 }% \itf@cell@R
404 \def\itf@cell@J{%
405   \gdef\itf@cellallign{\justifying\arraybackslash\itf@cellleft}%
406   \global\let\itf@cellralign\empty
407   \toks@\expandafter{\itf@cellleft\justifying\arraybackslash}%
408   \xdef\itf@cellleft{\the\toks@}%
409 }% \itf@cell@J
410 }% AfterPackage ragged2e
411 \def\itf@flush#1{%
412   \ifitf@colortbl      % colortbl uses \hfill rather than \hfil
413     \hskip \ifnum\currentgroup>13 \stretch{#1}%
414   \else\ifdim#1pt<1pt \itf@cellskip
415   \else \stretch{#1}%
416   \fi\relax
417   \else                 % array.sty
418     \ifnum \currentgroup>13\relax
419       \hfil \hskip1sp
420     \fi
421   \fi
422 }% \itf@flush

```

`\itf@removehfil` `\itf@removehfil` removes (eventually) the infinite stretchable glue inserted *before* the cell (in the preamble of `\halign`) to make the column alignment.

`\itf@ignorehfil` removes (eventually) the infinite stretchable glue inserted *after* the cell (in the preamble of `\halign`) to make the column alignment.

```

423 \newskip\itf@cellskip
424 \let\itf@hfil\hfil
425 \let\itf@hfill\hfill
426 \let\itf@hskip\hskip
427 \def\itf@removehfil{%
428   \ifitf@colortbl
429     \unkern \itf@cellskip = \lastskip
430     \ifnum\gluestretchorder\itf@cellskip = \tw@ \hskip-\itf@cellskip
431     \else \itf@cellskip = \z@skip
432     \fi
433   \else
434     \ifdim\lastskip=1sp\unskip\fi
435     \ifnum\gluestretchorder\lastskip = \ne
436       \hfilneg % \hfilneg for array.sty but not for colortbl...
437     \fi
438   \fi
439 }% \itf@removehfil
440 \def\itf@nohfil{%
441   \def\hfil{\let\hfil\itf@hfil}% local to (alignment template) group
442 }% \itf@nohfil
443 \AfterPackage*{colortbl}{%
444   \def\itf@nohfil{%
445     \def\hfil{\let\hfil\itf@hfil}% local to (alignment template) group
446     \def\hfill{\let\hfill\itf@hfill}% (colortbl uses \hfill) pfff...
447     \def\hskip##1\relax{\let\hskip\itf@hskip}}% local
448 }% AfterPackage colortbl
449 \def\itf@ignorehfil{%
450   \aftergroup\itf@nohfil
451 }% \itf@ignorehfil

```

452

## 21.4 + Interface TikZ (pgfmanual )

```
453 {*tikz}
454 \ProvidesPackage{interfaces-tikz}
455 [2010/10/27 v1.2 load of tikz libraries easily (FC)]
456 \RequirePackage{interfaces-base}
457 \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.

```
458 % Some catcode checking
459 \AfterPackage*{tikz}{%
460   \AtEndPreamble{%
461     \tikzset{every picture/.add style={%
462       /utils/exec={\@makeother\! \@makeother\:}}}}{}}
463 }%
464 }%
```

### 21.4.2 List of pgf/TikZ packages and libraries

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

```
465 \newcommand*\alltikzpackages{%
466   pgf,pgfmath,pgfcalendar,pgfplots,pgfpages,tikz%
467 }% \alltikzpackages
```

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

```
468 \newcommand*\allpgfmodules{matrix,plot}
```

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

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

```
469 \newcommand*\alltikzlibraries{%
470   arrows,%
471   chains,%
472   er,%
473   scopes,%
474   snakes,%
475   shapes,%
476   shapes.misc,%
477   shapes.geometric,%
478   shapes.arrows,%
479   shapes.multipart,%
480   shapes.callouts,%
481   shapes.symbols,%
482   shapes.gates.logic.US,%
483   shapes.gates.logic.IEC,%
484   decorations,%
485   decorations.pathmorphing,%
486   decorations.pathreplacing,%
487   decorations.shapes,%
488   decorations.text,%
489   decorations.markings,%
490   decorations.fractals,%
```

```

491   decorations.footprints,%
492   positioning,trees,%
493   automata,%
494   patterns,%
495   fadings,%
496   shadows,%
497   through,%
498   backgrounds,%
499   petri,%
500   fit,%
501   mindmap,%
502   folding,%
503   plothandlers,%
504   plotmarks,%
505   calc,%
506   matrix%
507 %%   topath,placements,%
508 }% \alltikzlibraries

```

### 21.4.3 pgfkeys definitions for \usetikz

```

509 \pgfqkeys{/itf/tikz}{%
510   basic/.code={%
511     \usetikzlibrary{calc,%
512       arrows,% for arrows
513       backgrounds,% for framed
514       scopes% for scopes and shortcut : { [scope] ... }
515     }%
516   },%
517   basic/.value forbidden,% \alltikzpackages
518   base/.alias=basic,%
519   all/.code={%
520     \expandafter\RequirePackage\expandafter{\alltikzpackages}%
521     \expandafter\usepgfmodule\expandafter{\allpgfmodules}% see pgfcorescopes.code.tex
522     \expandafter\usetikzlibrary\expandafter{\alltikzlibraries}% from: tikz.code.tex
523   },%
524   .unknown/.code=\itf@usetikzcode\pgfkeyscurrentname
525 }% \pgfkeys for \usetikz

\usetikz

526 \newrobustcmd*\usetikz[1]{%
527   \RequirePackage{tikz}%
528   \ifblank{#1}%
529     {\pgfqkeys{/itf/tikz}{all}}%
530     {\pgfqkeys{/itf/tikz}{#1}}%
531   \itf@pgfkeys% overwritten by \input primitive otherwise
532 }% \usetikz

```

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

```

533 \newrobustcmd*\itf@usetikzcode[1]{\begingroup
534   \edef\x{%
535     \noexpand\in@{,#1,}{,\alltikzpackages,}%
536     \noexpand\ifin@ \noexpand\usepackage{#1}%
537     \noexpand\else
538       \noexpand\in@{,#1,}{,\allpgfmodules,}%
539       \noexpand\ifin@ \noexpand\usepgfmodule{#1}%
540       \noexpand\else \noexpand\usetikzlibrary{#1}% \usetikzlibrary by default
541       \noexpand\fi
542     \noexpand\fi

```

```

543     }\expandafter\endgroup\x
544 }% \itf@usetikzcode

545 </tikz>

```

## 21.5 † Interface ↗ **titlesec** : **\sectionformat**

```

546 {*titlesec}
547 \ProvidesPackage{interfaces-titlesec}
548   [2010/10/27 v1.2 pgfkeys macros for titlesec (FC)]
549 \RequirePackage{interfaces-base}
550 \RequirePackage{auxhook}% for fix for \ttlp@append (odd/even pages, not implemented in interf
551 \itf@Check@Package{titlesec}
552 \itf@pgfkeys

```

### 21.5.1 **titlesec** modifications

```

553 %% t i t l e s e c "p a t c h e s"
554 \robustify\titleclass \robustify\titleformat \robustify\titlespacing
555 \AddLineBeginMainAux{%
556   \string\providetoggle\ttlp@append[2]{}}%
557 }
558 \BeforeFile*{ttlkeys.def}{%
559   \catcode32 9\relax % ignore spaces
560   \count@\endlinechar\endlinechar\m@ne % and line feeds
561 }
562 \AfterFile*{ttlkeys.def}{% restore spaces and line feeds
563   \catcode32 10\relax
564   \endlinechar\count@
565 }

```

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

```

566 %-----%
567 %% runinifempty => like hang but no paragraph break (like runin) if title is empty
568 \def\ttlh@runinifempty{%
569   \csname ttlh@ifx\ttl@savetitle@\empty runin\else hang\fi\endcsname
570 }% \ttlh@runinifempty

```

### 21.5.2 **\sectionformat**

```

\sectionformat \sectionformat\subsection* [<shape>]{<key=value>}

571 \newrobustcmd*\sectionformat[1]{\ifstar
572   {\itf@sectionformat{#1}{/*}}
573   {\itf@sectionformat{#1}{}}%
574 }% \sectionformat
575 \def\itf@sectionformat#1#2{@testopt
576   {\expandafter\expandafter\expandafter%
577    \itf@sectionf@rmat\expandafter\gobble\string#1\@nil{#2}}{}%
578 }% \sectionformat
579 \def\itf@sectionf@rmat#1\@nil#2[#3]#4{\itf@pgfkeys
580   \ifblank{#3}%
581     {\pgfqkeys{/itf/titlesec/#1#2}%
582      {/itf/titlesec/@setup={#1}{#2},#4,/itf/titlesec/@update={#1}{#2}}}%
583     {\pgfqkeys{/itf/titlesec/#1#2}%
584      {/itf/titlesec/@setup={#1}{#2},#4,shape={#3},/itf/titlesec/@update={#1}{#2}}}%
585 }% \itf@ttl@sectionformat

```

```

\itf@ttl@setup Macros used to retrieve the values set by \titleformat and/or \titlespacing. This allows to use
\itf@ttl@getformat \sectionformat concurrently with \titleformat and \titlespacing provided by titlesec.
\itf@ttl@getspacing 586% private keys @setup and @update
587 \pgfqkeys{/itf/titlesec}{%
588   @setup/.code 2 args=\itf@ttl@setup[#1]{#2},%
589   @update/.code 2 args=\expandafter\itf@ttl@update
590           \csname#1\endcsname{/itf/titlesec/#1}{#2}%
591 }
592 \newrobustcmd*\itf@ttl@setup[2]{%
593   \pgfkeysifdefined{/itf/titlesec/#1#2/shape}{}%
594     {\itf@ttl@newsection[#1]{#2}}%
595   \ifcsundef{ttls@#1#2}{}% no previous spacing settings
596     {\expandafter\expandafter\expandafter%
597      \itf@ttl@getspacing\csname ttls@#1#2\endcsname{/itf/titlesec/#1#2}}%
598   \ifcsundef{ttlf@#1#2}{}% no previous format settings
599     {\expandafter\expandafter\expandafter%
600      \itf@ttl@getformat\csname ttlf@#1#2\endcsname{/itf/titlesec/#1#2}}%
601 }% \itf@ttl@setup
602 \def\itf@ttl@setkey#1#2{\begingroup
603   \def\x{\pgfkeysvalueof{#1}}\def\y{#2}%
604   \expandafter\endgroup\ifx\x\y\else
605     \pgfkeyssetvalue{#1}{#2}\fi
606 }% \itf@ttl@setkey
607 \def\itf@ttl@getspacing#1#2#3#4#5#6{%
608   \itf@ttl@setkey{#6/left}{#1}%
609   \itf@ttl@setkey{#6/right}{#2}%
610   \itf@ttl@setkey{#6/top}{#3}%
611   \itf@ttl@setkey{#6/bottom}{#4}%
612   \itf@ttl@setkey{#6/indent first}{#5}%
613 }% \itf@ttl@getspacing
614 \def\itf@ttl@getformat#1#2{%
615   \csname ltx@\ifx\pgfkeysvalueof#1first\else second\fi oftwo\endcsname
616     {\itf@ttl@get@format{}}%
617     {\itf@ttl@get@format{#1}{#2}}%
618 }% \itf@ttl@getformat
619 \def\itf@ttl@get@format#1#2#3#4#5#6#7{%
620   \ifrelax\detokenize{#1}\relax\else
621     \pgfkeysEsetvalue{#7/shape}{\expandafter\itf@strip@at\string#1}%
622   \fi
623   \itf@ttl@setkey{#7/font}{#2}%
624   \itf@ttl@setkey{#7/label}{#3}%
625   \itf@ttl@setkey{#7/sep}{#4}%
626   \itf@ttl@setkey{#7/before}{#5}%
627   \itf@ttl@setkey{#7/after}{#6}%
628 }% \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).

```

629 \newrobustcmd*\itf@ttl@update[3]{%
630   \ifblank{#3}
631     {\ttl@getkeys{#1}{titlesec}}
632     {\ttl@getkeys{name=#1,numberless}{titlesec}}%
633   \pgfkeysgetvalue{#2#3/shape}\itf@tp
634   \ifcsundef{tth@}\itf@tp\fi
635     {\begingroup \makeatletter
636       \InputIfFileExists{\itf@tp.tss}{}{%
637         \PackageError{titlesec}{Unknown shape}%
638         {Shapes are defined in files with extension tss\MessageBreak
639          Either you have misspelled the shape\MessageBreak
640          or there is no a \itf@tp.tss file}}%
641     \endgroup}

```

```

642      {}%
643  \iftl@explicit@\temptokena{\csdef{ttl@}{\ttl@a}##1}%
644  \else@\temptokena{\csdef{ttl@}{\ttl@a}}%
645  \fi
646  \the@\temptokena{%
647      \pgfkeysvalueof{#2#3/shape@cs}%
648      {\pgfkeysvalueof{#2#3/font}}%
649      {\pgfkeysvalueof{#2#3/label}}%
650      {\pgfkeysvalueof{#2#3/sep}}%
651      {\pgfkeysvalueof{#2#3/before}}%
652      {\pgfkeysvalueof{#2#3/after}}}}%
653  \csname ttl@compat\ttl@a\endcsname
654  \ttl@spacing@ii
655      {\pgfkeysvalueof{#2#3/indent first}}%
656      {\pgfkeysvalueof{#2#3/left}}%
657      {\pgfkeysvalueof{#2#3/top}}%
658      {\pgfkeysvalueof{#2#3/bottom}}%
659      [\pgfkeysvalueof{#2#3/right}]%
660 }% \itf@ttl@update

```

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

- change the class of an existing sectioning command;
- or:
- create a new sectioning command.

```

661 %%-----%
662 % t i t l e c l a s s
663 \newrobustcmd*\itf@ttl@titleclass{\ifstar
664   {\itf@ttl@titleclass@i*}
665   {\itf@ttl@titleclass@i{}}}%
666 }% \itf@ttl@titleclass
667 \def\itf@ttl@titleclass@i#1#2[#4]#5@nil{\ifblank{#4}
668   {%
669     \expandafter\titleclass\csname#2\endcsname{#3}%
670   }
671   {%
672     \expandafter\titleclass\csname#2\endcsname{#3}[{#4}]% #4 = sup-level
673     \if\relax#1\relax % not starred
674       \expandafter\expandafter\expandafter%
675         \itf@ttl@titleclass@ii\expandafter\@gobble\string#4\relax{#2}% subsection\relax{#2}%
676     \fi
677   }%
678   \expandafter\providecommand\csname l@#2\endcsname[2]{}% no entry in toc
679 }% \itf@ttl@titleclass@i
680 \def\itf@ttl@titleclass@ii#1\relax#2{%
681   \ifcsundef{c@#2}
682     {\newcounter{#2}[{#1}]}% creates counter for #1
683     {\@addtoreset{#2}{#1}}%
684   \csxdef{the#2}{%
685     \expandafter\noexpand\csname the#1\endcsname
686     .\noexpand\@arabic\expandafter\noexpand\csname c@#2\endcsname
687   }% \the#2
688   \ifcsundef{l@#1}{}{%
689     \expandafter\providecommand\csname l@#2\endcsname[2]{%
690       \csname l@#1\endcsname{##1}{##2}%
691     }% \l@#2 = \l@newsection is (by default) the same as \l@sup-level
692   }%
693 }% \itf@ttl@titleclass@ii

```

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

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

```
694 %-----  
695 %% Creating new sections  
696 \def\itf@ttl@createsection#1{  
  sectionning command undefined: temporarily defined  
  \expandafter\titleclass\csname#1\endcsname{straight}[*] in straight class  
  \itf@AfterPackage{hypbmsec}{%  
    \AfterPreamble{  
      \csletcs{hbs@#1}{#1}%  
      \csxdef{#1}{\noexpand\hbs@do\expandafter\noexpand\csname hbs@#1\endcsname}%  
    }%  
  }%  
  \itf@AfterPackage{interfaces-tocloft}{%  
    \itf@cft@newsection{#1}{toc}%  
  }%  
  \itf@AfterPackage{interfaces-hypbmsec}{%  
    \expandafter\let\csname last#1title\endcsname\relax  
    \expandafter\let\csname last#1bookmark\endcsname\relax  
    \expandafter\let\csname last#1toc\endcsname\relax  
  }%  
} % \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.

```
713 %-----  
714 %% Defining keys for each section  
715 \def\itf@ttl@newsection#1#2{  
  #1=section, #2=/* or empty  
  \ifcsundef{#1}{  
    sectionning command undefined: temporarily defined in straight class  
    \itf@ttl@createsection{#1}}  
  {}%  
  \ifblank{#2}{%  
    \pgfqkeys{/itf/titlesec/#1}{% these keys do not exist for starred/numberless sections  
      class/.code=\itf@ttl@titleclass{#1}##1[]@\nil,% with \addtoreset  
      class/.value required,% if sup-level is given  
      class*/.code=\itf@ttl@titleclass*{#1}##1[]@\nil,% without \addtoreset  
      class*/.value required,% even if sup-level is given  
      shape/.initial=% initial shapes set by titlesec.sty for not numbered sections.  
      the#1/.store in cs=the#1,%  
      the#1/.default=\pgfkeysvalueof{/itf/titlesec/#1/label},%  
    }%  
    {% starred (numberless) unless titlesec, we do not put the default shape to hang,  
     \pgfkeysifdefined{/itf/titlesec/#1/shape}{% but to the shape of the not starred  
       \pgfkeysgetvalue{/itf/titlesec/#1/shape}\itf@tp% sectionning command  
       \def\itf@tp{hang}}%  
     \pgfkeyslet{/itf/titlesec/#1#2/shape}\itf@tp%  
   }%  
  \pgfqkeys{/itf/titlesec/#1#2}{%  
    shape/.value required,%  
    shape@cs/.initial=  
      \ifdef{\itf@last@from@next\itf@last@from@next{#1}\fi % interfaces-hypbmsec  
      \csname ttlh@\pgfkeysvalueof{/itf/titlesec/#1#2/shape}\endcsname,% constant  
      shape@cs/.is private=interface-titlesec,% constant  
      font/.store as code,%  
      format/.alias*=font,%  
      label/.store as code,%
```

```

744     sep/.estore as glue,%
745         labelsep/.alias*=sep,%
746         before/.store as code,%
747         after/.store as code,%
748         indent first/.true=\pgfkeyssetvalue{/itf/titlesec/#1#2/indent first}{\@ne},%
749         indent first/.false=\pgfkeyssetvalue{/itf/titlesec/#1#2/indent first}{\z@},%
750             indentfirst/.alias=indent first,%
751                 indent first=true,%
752                 left/.estore as glue,%
753                 top/.estore as glue,%
754                 bottom/.estore as glue,%
755                 right/.estore as glue,%
756                 break/.code=\ifblank{##1}
757                     {\cslet{\#1break}{\empty}}
758                     {\protected\csdef{\#1break}{##1}},%
759                     break*/.code=\protected\apptocs{\#1break}{##1},%
760                     break+/.alias=break*,%
761                     mark/.code=\ifblank{##1}
762                         {\cslet{\#1mark}{\gobble}}
763                         {\protected\long\csdef{\#1mark}####1{##1}},%
764                         mark/.default=,
765 }% \pgfkeys
766 \ifblank{##2}{% no default marks for starred (numberless) \section*
767 \itf@AfterPackage{interfaces-marks}{% requires interfaces-marks
768     \pgfkeys{/itf/titlesec/#1/mark/.default=\marksthe{##1}}%
769 }% AfterPackage interfaces-mark
770 \itf@AfterPackage{interfaces-bookmark}{% requires interfaces-bookmark
771     \pgfqkeys{/itf/titlesec/#1}{% does not exist for starred sections
772         bookmark/.is multiple choice,
773             bookmark/level/.estore in cs=toclevel@#1,% depth of the bookmark
774                 bookmark/level/.default=,
775                 bookmark/level/.blank=\ifcvoid{toclevel@#1}{99}{\csuse{toclevel@#1}},%
776                 bookmark/toclevel/.alias=bookmark/level,
777                 bookmark/bold/.code=\ifblank{##1}
778                     {\pgfkeyslet{/itf/titlesec/#1/bookmark/bold}{\empty}}
779                     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/bold}{bold=##1}},%
780                     bookmark/bold/.default=true,
781                     bookmark/bold*/.code=\ifblank{##1}
782                         {\pgfkeyslet{/itf/titlesec/#1/bookmark/bold*}{\empty}}
783                         {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/bold*}{bold*=##1}},%
784                         bookmark/bold*/.default=true,
785                     bookmark/italic/.code=\ifblank{##1}
786                         {\pgfkeyslet{/itf/titlesec/#1/bookmark/italic}{\empty}}
787                         {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/italic}{italic=##1}},%
788                         bookmark/italic/.default=true,
789                     bookmark/italic*/.code=\ifblank{##1}
790                         {\pgfkeyslet{/itf/titlesec/#1/bookmark/italic*}{\empty}}
791                         {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/italic*}{italic*=##1}},%
792                         bookmark/italic*/.default=true,
793                     bookmark/color/.code=\ifblank{##1}
794                         {\pgfkeyslet{/itf/titlesec/#1/bookmark/color}{\empty}}
795                         {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/color}{color=##1}},%
796                         bookmark/color/.default=,
797                     bookmark/color*/.code=\ifblank{##1}
798                         {\pgfkeyslet{/itf/titlesec/#1/bookmark/color*}{\empty}}
799                         {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/color*}{color*=##1}},%
800                         bookmark/color*/.default=,
801                     bookmark/on/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on}{disable=false},%
802                         bookmark/on/.value forbidden,
803                     bookmark/off/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on}{disable=true},%
804                         bookmark/off/.value forbidden,
805                         bookmark/none/.alias=bookmark/off,
806                     bookmark/style/.code=\ifblank{##1}

```

```

807      {\pgfkeyslet{/itf/titlesec/#1/bookmark/style}\@empty}
808      {\pgfkeysetvalue{/itf/titlesec/#1/bookmark/style}{style=##1,}},%
809      bookmark/.default=,%  

810      bookmark/.unknown/.code=\pgfkeysetvalue{/itf/titlesec/#1/bookmark/other}{\pgfkeysco}
811      bookmark/bold/.initial=,%  

812      bookmark/bold*/.initial=,%  

813      bookmark/italic/.initial=,%  

814      bookmark/italic*/.initial=,%  

815      bookmark/color/.initial=,%  

816      bookmark/color*/.initial=,%  

817      bookmark/on/.initial=,%  

818      bookmark/style/.initial=,%  

819  } }% \AfterPackage*{interfaces-bookmark}  

820  }% not starred (numberless) \section  

821  {}% starred/numberless \section*: do not create the keys  

822 }% \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.

```

823 %% bookmarks styles for sectionning commands:  

824 \@makeother\#  

825 \itf@AfterPackage{interfaces-bookmark}{%  

826   \itf@patch\itf@preapptocmd\addcontentsline % #2=section name in  

827   {\begingroup\itf@ttl@BKM{/itf/titlesec/#2/bookmark}}%\addcontentsline  

828   {\endgroup}  

829   {\PackageInfo{interfaces-titlesec}  

830     {Command \string\addcontentsline\space successfully patched!}}}  

831   {\PackageWarning{interfaces-titlesec}  

832     {Patch of \string\addcontentsline\space failed  

833     \MessageBreak bookmark-style specs for sections  

834     \MessageBreak will be unavailable}}%  

835 }% AfterPackage interfaces-bookmark  

836 \catcode`\#=6  

837 \def\itf@ttl@BKM#1{%
838   \pgfkeysifdefined{#1/.@cmd}{%
839     {%
840       \edef\itf@tp{\% do nothing if the keys are not set (shortcut)
841         \pgfkeysvalueof{#1:bold}%
842         \pgfkeysvvalueof{#1:bold*}%
843         \pgfkeysvvalueof{#1/italic}%
844         \pgfkeysvvalueof{#1/italic*}%
845         \pgfkeysvvalueof{#1/color}%
846         \pgfkeysvvalueof{#1/color*}%
847         \pgfkeysvvalueof{#1/on}%
848         \pgfkeysvvalueof{#1/style}}%
849       \ifx\itf@tp\@empty
850       \else
851         \global\pgfkeyslet{#1:bold*}\@empty
852         \global\pgfkeyslet{#1/italic*}\@empty
853         \global\pgfkeyslet{#1/color*}\@empty
854         \expandafter\bookmarksetup\expandafter{\itf@tp}%
855         \pgfkeysifdefined{/itf/bookmark/*}{%
856           {%
857             \bookmarkssetup{color*=} overwrites \sectionformat...{bookmark/color*=}%
858             \pgfkeysetvalue{/itf/bookmark/*}\itf@tp
859             \expandafter\bookmarksetup\expandafter{\itf@tp}%
860           }%
861         \fi}
862       {}%
863     }%
864   }%
865 }% \itf@ttl@BKM

```

## 21.5.4 \bottomtitles

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

```

864 %-----  

865 %% bottomtitles  

866 \newrobustcmd*\bottomtitles{\@ifnextchar\bgroup  

867   \itf@setbottomtitlespace  

868   {\afterassignment\itf@set@bottomtitle@space\itf@sk}%
869 }% \bottomtitles  

870 \def\itf@setbottomtitlespace#1{%
871   \@defaultunits\itf@sk#1\textheight\relax\@nnil  

872   \itf@set@bottomtitle@space  

873 }% \itf@setbottomtitlespace  

874 \def\itf@set@bottomtitle@space{\edef\bottomtitlespace{\the\itf@sk}}  

  

875 </titlesec>
```

## 21.6 <sup>f</sup> Interface hyperref

```

876 (*hyperref)  

877 \ProvidesPackage{interfaces-hyperref}  

878   [2010/10/27 v1.2 provides raised hyperdef (FC)]  

879 \RequirePackage{interfaces-base}  

880 \itf@Check@Package{hyperref}
```

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

```

881 \newrobustcmd\raisedhyperdef[4][]{\begingroup  

882   \ifblank{#1}{  

883     {}  

884   }%  

885   \expandafter\def\expandafter\HyperRaiseLinkHook\expandafter{  

886     \expandafter\def\expandafter\HyperRaiseLinkHook  

887       \expandafter{\HyperRaiseLinkHook}% restore current definition  

888     \HyperRaiseLinkHook  

889     \advance\noexpand\HyperRaiseLinkLength by\dimexpr#1\relax\relax}%  

890   }%  

891   \ifx \\#2\\ \def \Hy@AnchorName {#3}\else  

892     \def \Hy@AnchorName {#2.#3}%  

893   \fi  

894   \ltx@leavevemode  

895   \Hy@raisedlink{\hyper@anchorstart{\Hy@AnchorName}\hyper@anchorend}%  

896   \endgroup  

897   #4%  

898 }% \raisedhyperdef  

  

899 </hyperref>
```

## 21.7 <sup>f</sup> Interface bookmark

```

900 (*bookmark)  

901 \ProvidesPackage{interfaces-bookmark}  

902   [2010/10/27 v1.2 more control on bookmarks (FC)]  

903 \RequirePackage{interfaces-base}  

904 \RequirePackage{interfaces-hyperref}  

905 \RequirePackage{refcount}  

906 \itf@Check@Package{bookmark}  

907 \itf@pgfkeys
```

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

```
908 \expandafter\itf@patch\expandafter\patchcmd\csname\string\bookmark\endcsname
```

```

909  {\immediate\write\@mainaux}{\itf@BKM@write}
910  {\PackageInfo{interfaces-bookmark}
911    {Command \expandafter\string\csname\string\bookmark\endcsname\space
912     successfully patched!}}
913  {\PackageWarning{interfaces-bookmark}
914    {Command \expandafter\string\csname\string\bookmark\endcsname\space
915     could not be patched\MessageBreak
916     Loading of interfaces-bookmark is canceled}%
917  \endinput}

```

### 21.7.1 star keys\* definitions

```

918 \global\let\itf@BKM@enable@ne % 0 = disable*, 1=enable, \@M = disable
919 \pgfqkeys{/itf/bookmark}{%
920   disable/.true=\global\let\itf@BKM@enable@\M,%
921   disable/.false=\global\let\itf@BKM@enable@ne,%
922   disable*/.true=\global\let\itf@BKM@enable\z@,%
923   disable*/.false=\global\let\itf@BKM@enable@ne,%
924   atend/.true=\ifnum\itf@BKM@enable>0\edef\itf@BKM@enable{\number-\itf@BKM@enable}\fi,%
925   atend/.false=\ifnum\itf@BKM@enable<0\edef\itf@BKM@enable{\number-\itf@BKM@enable}\fi,%
926   bold*/.true=\pgfqkeys{/itf/bookmark}{@save={bold=true}},%
927   bold*/.false=\pgfqkeys{/itf/bookmark}{@save={bold=false}},%
928   bold*/@blank/.code=\pgfqkeys{/itf/bookmark/@restore}{bold},%
929   bold*/.blank=@blank,%
930   italic*/.true=\pgfqkeys{/itf/bookmark}{@save={italic=true}},%
931   italic*/.false=\pgfkeys{/itf/bookmark}{@save={italic=false}},%
932   italic*/@blank/.code=\pgfqkeys{/itf/bookmark/@restore}{italic},%
933   italic*/.blank=@blank,%
934   color*/.code={\ifblank{#1}{%
935     {\pgfqkeys{/itf/bookmark/@restore}{color}}%
936     {\pgfqkeys{/itf/bookmark}{@save={color={#1}}}}}}%
937 },%
938 @save/.code={%
939   \pgfkeysifdefined{/itf/bookmark/@save/color}{%
940     {\pgfkeysset{/itf/bookmark/@save/color}\BKM@color}}%
941   \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{%
942     {\pgfkeysset{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}}%
943   \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}{%
944     {\pgfkeysEsetvalue{/itf/bookmark/@save/grouplevel}{\the\currentgrouplevel}}%
945     \pgfkeysaddvalue{/itf/bookmark/*}{}{, #1}}%
946   \bookmarksetup{#1}}%
947 },%
948 @restore/color/.code=%
949   \pgfkeysifdefined{/itf/bookmark/@save/color}{%
950     {\pgfkeysgetvalue{/itf/bookmark/@save/color}\BKM@color}}%
951   {}%
952 @restore/FontStyle/.code=%
953   \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{%
954     {\pgfkeysgetvalue{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}}%
955   {}%
956 @restore/.code={%
957 %   \pgfqkeys{/itf/bookmark/@restore}{color,FontStyle}% repeating the following lines
958   \pgfkeysifdefined{/itf/bookmark/@save/color}{% is quicker than setting a /.style ...
959     {\pgfkeysgetvalue{/itf/bookmark/@save/color}\BKM@color}}%
960   {}%
961   \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{%
962     {\pgfkeysgetvalue{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}}%
963   {}%
964   \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}{%
965     {% shortcut: \bookmarksetup{bold*} called inside main group (level 0)
966     \ifnum\pgfkeysvalueof{/itf/bookmark/@save/grouplevel}=\z@%
967       \global\pgfkeysset{/itf/bookmark/@save/FontStyle}\@undefined

```

```

968          \global\pgfkeyslet{/itf/bookmark/@save/color}\@undefined
969          \global\pgfkeyslet{/itf/bookmark/@save/grouplevel}\@undefined
970          \global\pgfkeyslet{/itf/bookmark/*}\@undefined
971          \global\let\BKM@color\BKM@color
972          \global\let\BS@BKM@FontStyle\BS@BKM@FontStyle
973          \else % restore after each group as needed
974          \pgfkeyslet{/itf/bookmark/@save/FontStyle}\@undefined
975          \pgfkeyslet{/itf/bookmark/@save/color}\@undefined
976          \pgfkeyslet{/itf/bookmark/*}\@undefined
977          \ifnum\currentgrouplevel>\z@
978              \aftergroup\itf@BKM@restore
979          \else
980              \pgfkeyslet{/itf/bookmark/@save/grouplevel}\@undefined
981          \fi\fi
982      }
983  {}%
984 }%
985 @restore/bold/.code={% bold is "hidden" in \BS@BKM@FontStyle
986     \begingroup
987         \edef\itf@tp{\bookmarkget{bold}}%
988         \pgfqkeys{/itf/bookmark/@restore}{FontStyle}%
989         \count@=\bookmarkget{bold}%
990         \edef\itf@tp{\endgroup
991             \ifnum\itf@tp=\count@
992                 \noexpand\bookmarksetup{%
993                     bold\ifcase\count@ =false\fi}\fi
994             }\itf@tp},%
995 @restore/italic/.code={% italic is "hidden" in \BS@BKM@FontStyle
996     \begingroup
997         \edef\itf@tp{\bookmarkget{italic}}%
998         \pgfqkeys{/itf/bookmark/@restore}{FontStyle}%
999         \count@=\bookmarkget{italic}%
1000        \edef\itf@tp{\endgroup
1001            \ifnum\itf@tp=\count@
1002                \noexpand\bookmarksetup{%
1003                    italic\ifcase\count@ =false\fi}\fi
1004            }\itf@tp},%
1005 }
1006 \def\itf@BKM@restore{%
1007     \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}
1008     {\pgfkeysalso{/itf/bookmark/@restore}}
1009     {}%
1010 }% \itf@BKM@restore
1011 %% Plug the BKM keys in the interfaces pgfkeys (not efficient but quickly written...)
1012 \define@key{BKM}{disable}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/disable={#1}}}
1013 \define@key{BKM}{disable*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/disable*={#1}}}
1014 \define@key{BKM}{bold*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/bold*={#1}}}
1015 \define@key{BKM}{italic*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/italic*={#1}}}
1016 \define@key{BKM}{color*}[]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/color*={#1}}}
1017 \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.

```

1018 \def\itf@BKM@write#1%
1019     \aftergroup\itf@BKM@restore
1020     \ifcase\itf@BKM@enable
1021         \global\let\itf@BKM@enable\@ne
1022     \or
1023         \immediate\write\@mainaux{\#1}%
1024     \else
1025         \ifcase -\itf@BKM@enable % (case -0 = 0 tested before)
1026             \or % (case -1 = enable but At End)
1027                 \let\itf@BKM@enable\@ne

```

```

1028      \edef\itf@tp{\immediate\write\@mainaux{#1}%
1029          \expandafter\BookmarkAtEnd\expandafter{\itf@tp}%
1030      \fi
1031  \fi
1032 }% \itf@BKM@write

```

## 21.7.2 ref and pageref keys

```
\itf@BKM@pageref
```

```

1033 %% pageref= key for \bookmark (using refcount : \getpagerefnumber)
1034 \def\itf@BKM@pageref[#1]{%
1035   \expandafter\itf@BKM@@pageref
1036   \csname r@#1\endcsname{#1}%
1037 }% \itf@BKM@pageref
1038 \def\itf@BKM@@pageref#1#2{%
1039   \ifx#1\relax
1040     \protect\G@refundefinedtrue
1041     \@latex@warning{%
1042       Hyper reference ‘#2’ on page \thepage \space undefined
1043       \MessageBreak for bookmark destination page
1044     }%
1045   \fi
1046   \edef\BKM@page{\getrefbykeydefault{#2}{page}{1}}%
1047 }% \itf@BKM@pageref
1048 \define@key{BKM}{pageref}{%
1049   \Hy@babelnormalise\itf@BKM@pageref[{"#1"}]}

```

```
\itf@BKM@ref
```

```

1050 %% ref= key for \bookmark
1051 \def\itf@BKM@ref[#1]{%
1052   \expandafter\itf@BKM@@ref
1053   \csname r@#1\endcsname{#1}%
1054 }% \itf@BKM@ref
1055 \def\itf@BKM@@ref#1#2{%
1056   \ifx#1\relax
1057     \protect\G@refundefinedtrue
1058     \@latex@warning{%
1059       Hyper reference ‘#2’ on page \thepage \space undefined
1060       \MessageBreak for bookmark destination
1061     }%
1062     \def\BKM@dest{??}%
1063   \else
1064     \edef\BKM@dest{\expandafter\@fourthoffive#1\@empty\@empty}%
1065   \fi
1066 }% \BKM@@ref
1067 \define@key{BKM}{ref}{%
1068   \Hy@babelnormalise\itf@BKM@ref[{"#1"}]}

```

1069 </bookmark>

## 21.8 † Interface **hypbmsec**

### 21.8.1 **interfaces-hypbmsec**

```

1070 <*hypbmsec>
1071 \ProvidesPackage{interfaces-hypbmsec}
1072     [2010/10/27 v1.2 lastsection etc. (FC)]
1073 \RequirePackage{interfaces-base}
1074 \itf@Check@Package{hypbmsec}
1075 \AfterPackage*{interfaces-titlesec}{% requires interface-titlesec
1076     \input{interfaces-hypbmsec.code.tex}%
1077 }%
1078 </hypbmsec>
```

### 21.8.2 **interfaces-hypbmsec.code.tex**

```

1079 <*hypbmsec.code.tex>
1080 \ProvidesFile{interfaces-hypbmsec.code.tex}
1081     [2010/10/27 v1.2 code for interfaces-hypbmsec (FC)]
```

\hbs@do (patch)

```

1082 \itf@patch\pretocmd\hbs@do{%
1083     \edef\itf@tp{\expandafter\itf@strip@at\string#1}%
1084 }%
1085 {\PackageInfo{interfaces-hypbmsec}{%
1086     {Command \string\hbs@do\space successfully patched}}%
1087 {\PackageWarning{interfaces-hypbmsec}{%
1088     {Command \string\hbs@do\space could not be patched
1089     \MessageBreak loading of ‘interfaces-hypbmsec’ aborted
1090     \MessageBreak \string\lastsection\space will not work}%
1091 \endinput}}
```

\hbs@process (patch)

```

1092 \itf@patch\pretocmd\hbs@process{%
1093     \itf@hbs@next{\#1}%
1094 }%
1095 {\PackageInfo{interfaces-hypbmsec}{%
1096     {Command \string\hbs@process\space successfully patched}}%
1097 {\PackageWarning{interfaces-hypbmsec}{%
1098     {Command \string\hbs@process\space could not be patched
1099     \MessageBreak loading of ‘interfaces-hypbmsec’ aborted
1100     \MessageBreak \string\lastsection\space will not work}%
1101 \endinput}}}
```

\itf@hbs@next

```

1102 \long\def\itf@hbs@next#1{%
1103     \csdef{itf@next\itf@tp title}{\#1}%
1104     \expandafter\let\csname itf@next\itf@tp toc\expandafter\endcsname
1105         \csname
1106             \ifx\hbs@tocstring\relax itf@next\itf@tp title\else
1107                 hbs@tocstring\fi
1108         \endcsname
1109     \expandafter\let\csname itf@next\itf@tp bookmark\expandafter\endcsname
1110         \csname
1111             \ifx\hbs@bmstring\relax itf@next\itf@tp toc\else
1112                 hbs@bmstring\fi
1113         \endcsname
1114     \expandafter\expandafter\expandafter
1115     \expandafter\expandafter\expandafter}
```

```

1116  \expandafter\itf@hbs@chkclass
1117      \expandafter\expandafter\expandafter\string
1118          \expandafter\hbs@seccmd\string\ttl@nil
1119      {\expandafter\itf@last@from@next\expandafter{\itf@tp}}%
1120 }% \itf@hbs@next
1121 \long\expandafter\def\expandafter\itf@hbs@chkclass
1122     \expandafter#\expandafter\string\ttl@#2@nil{%
1123 \ifblank{#1}{\if@nobreak\itf@hbs@nobreakfix#2@nnil\fi}%
1124 }% \itf@hbs@chkclass
1125 \def\itf@hbs@nobreakfix#1{\itf@hbs@n@breakfix}
1126 \def\itf@hbs@n@breakfix#1#2@nnil{%
1127     \unless\ifx\lastsectionname\relax
1128     \unless\ifnum\csname\ttl@#1\endcsname > \csname\ttl@lastsectionname\endcsname\relax
1129         @nobreakfalse
1130     \fi\fi
1131 }% \itf@hbs@nobreakfix

\itf@last@from@next

1132 \def\itf@last@from@next#1{%
1133     \gdef\lastsectionname{#1}%
1134     \global\expandafter\let\csname last#1title\expandafter\endcsname
1135         \csname itf@next#1title\endcsname
1136     \global\expandafter\let\csname last#1toc\expandafter\endcsname
1137         \csname itf@next#1toc\endcsname
1138     \global\expandafter\let\csname last#1bookmark\expandafter\endcsname
1139         \csname itf@next#1bookmark\endcsname
1140     \let\@elt\itf@resetlastsections
1141     \csname cl@#1\endcsname
1142 }% \itf@last@from@next
1143 \def\itf@resetlastsections#1{%
1144     \expandafter\let\csname last#1title\endcsname\relax
1145     \expandafter\let\csname last#1toc\endcsname\relax
1146     \expandafter\let\csname last#1bookmark\endcsname\relax
1147     \csname cl@#1\endcsname
1148 }% \itf@resetlastsections

1149 \let\lastsectionname\relax
1150 @for\itf@tp:=part,section,subsection,subsubsection,paragraph,subparagraph%
1151   \do{\expandafter\let\csname last\itf@tp title\endcsname\relax
1152     \expandafter\let\csname last\itf@tp toc\endcsname\relax
1153     \expandafter\let\csname last\itf@tp bookmark\endcsname\relax
1154   }%
1155 \AtBeginDocument{\ifundef\chapter{}%
1156   {\let\lastchaptertitle\relax
1157   \let\lastchaptertos\relax
1158   \let\lastchapterbookmark\relax}%
1159 }%
1160 
```

## 21.9 † Interface ☞ fancyhdr : \pagesetup

```

1161 {*fancyhdr}
1162 \ProvidesPackage{interfaces-fancyhdr}
1163   [2010/10/27 v1.2 pgfkeys macros for fancyhdr (FC)]
1164 \RequirePackage{interfaces-base}
1165 \itf@Check@Package{fancyhdr}
1166 \itf@pgfkeys
1167 \robustify\fancyhf
1168 \robustify\fancyhfoffset

```

### 21.9.1 \pagesetup

\pagesetup

```

1169 \newrobustcmd*\pagesetup{%
1170   \let\fancy@gbl\relax % Just in case...
1171   \@ifstar
1172     {\itf@pagesetup{}}
1173     {\itf@pagesetup{reset,}{}}
1174 }% \pagesetup
1175 \def\itf@pagesetup#1{\@testopt
1176   {\itf@p@gesetup{#1}}{}%
1177 }% \itf@pagesetup
1178 \def\itf@p@gesetup#1[#2]#3{\ifblank{#2}
1179   {% modifies the current page style
1180     \itf@pgfkeys\pgfqkeys{/itf/fancyhdr}{#1#3}\relax\ps@@fancy}%
1181   {% defines or modifies a given page style
1182     \ifblank{#1}
1183       \csappto% star form: keep existing value (if not overwritten)
1184       \csdef% no star form: overwrite (+ reset at \pagestyle time)
1185       {ps@#2}{\itf@pgfkeys\pgfqkeys{/itf/fancyhdr}{#1#3}\relax\ps@@fancy}%
1186   }%
1187 }% \itf@p@gesetup
1188 \pgfqkeys{/itf/pagesetup}{%
1189   @current/.is private=interfaces-fancyhdr,%
1190   @style/.is private=interfaces-fancyhdr,%
1191 }

```

\pagestyle

```

1192 \renewrobustcmd*\pagestyle[1]{\ifcsundef{ps@#1}
1193   {\itf@undefinedpagestyle{#1}}
1194   {\csname ps@#1\endcsname}%
1195 }% \pagestyle

```

\thispagestyle

```

1196 \renewrobustcmd*\thispagestyle[1]{\ifcsundef{ps@#1}
1197   {\itf@undefinedpagestyle{#1}}
1198   {\global\@specialpagetrue\gdef\@specialstyle{#1}}%
1199 }% \thispagestyle
1200 \newcommand*\itf@undefinedpagestyle[1]{%
1201   \@latex@error{Undefined page style: #1
1202     \MessageBreak your command will be ignored}@\eha
1203 }% \itf@undefinedpagestyle

```

## 21.9.2 Keys for \pagestyle

```

1204 \pgfkeys{%
1205   /handlers/.fancy@hf/.code=%
1206   \pgfkeysalso{\pgfkeyscurrentpath/.store as code=%}
1207   \csedef{fancy#1}{%
1208     \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath/font}%
1209     \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\strut},%
1210 }% handler for fancyhdr interface
1211 \pgfqkeys{/itf/fancyhdr}{% keys for \pagestyle
1212   code/.code={#1},% execution at expansion of \ps@... => \pgfqkeys{...,code={...}},...
1213   pagestyle/.style={code=\pagestyle{#1}},%
1214   page style/.alias=pagestyle,%
1215   load style/.alias=pagestyle,%
1216   head/rule/.store as dimen in cs=headrulewidth,%
1217   head/rule/.default=.4pt,%
1218   head/rule/.blank=\z@,%
1219   head/rulewidth/.alias*=head/rule,%
1220   headrulewidth/.alias*=head/rule,%
1221   head/rule/width/.alias*=head/rule,%
1222   head/rule/font/.store as code,%
1223   head/rule/color/.style={head/rule/font+=\color{#1}},%
1224   headruler/.code=%
1225   \let\headrule\pgfkeyscurrentvalue
1226   \ifx\headrule\@empty\else
1227     \preto\headrule{\bgroup\pgfkeysvalueof{/itf/fancyhdr/head/rule/font}}%
1228     \appto\headrule{\egroup}%
1229   \fi,%
1230   headruler/.default=%
1231   \if@fancyplain\let\headrulewidth\plainheadrulewidth\fi
1232   \hrule@height\headrulewidth \@width\headwidth
1233   \vskip-\headrulewidth\relax,%
1234   headrule/.alias=headruler,%
1235   headrule,% sets default
1236   foot/rule/.store as dimen in cs=footrulewidth,%
1237   foot/rule/.default=.4pt,%
1238   foot/rule/.blank=\z@,%
1239   foot/rulewidth/.alias*=foot/rule,%
1240   footrulewidth/.alias*=foot/rule,%
1241   foot/rule/width/.alias*=foot/rule,%
1242   foot/rule/font/.store as code,%
1243   foot/rule/color/.style={foot/rule/font+=\color{#1}},%
1244   footruler/.code=%
1245   \let\footrule\pgfkeyscurrentvalue
1246   \ifx\footrule\@empty\else
1247     \preto\footrule{\bgroup\pgfkeysvalueof{/itf/fancyhdr/head/rule/font}}%
1248     \appto\footrule{\egroup}%
1249   \fi,%
1250   footruler/.default=%
1251   \if@fancyplain\let\footrulewidth\plainfootrulewidth\fi
1252   \vskip-2\footrulewidth
1253   \hrule@height\footrulewidth \@width\headwidth\vskip\footrulewidth,%
1254   footrule/.alias=footruler,%
1255   footrule,% sets default
1256   rule/.style={head/rule={#1},foot/rule={#1}},%
1257   rule+/.style={head/rule+={#1},foot/rule+={#1}},%
1258   rule-/.style={head/rule-={#1},foot/rule-={#1}},%
1259   rule*/.style={head/rule*={#1},foot/rule*={#1}},%
1260   rules/.alias*=rule,rulewidth/.alias*=rule,%
1261   rule/width/.alias*=rule,rules/width/.alias*=rule,%
1262   norule/.style={rule=\z@},%
1263   norule/.value forbidden,%
1264   norules/.alias=norule,%

```

```

1265 rule/font/.style={head/rule/font={#1},foot/rule/font={#1}},%
1266 rule/font+/.style={head/rule/font+={#1},foot/rule/font+={#1}},%
1267     rule/font*/.alias=rule/font+,%
1268     rules/font/.alias=rule/font,
1269     rules/font+/.alias=rule/font+,%
1270     rules/font*/.alias=rule/font+,%
1271 rule/color/.style={head/rule/color={#1},foot/rule/color={#1}},%
1272     rules/color/.alias=rule/color,%
1273 foot/rule/skip/.store as glue in cs=footruleskip,%
1274     foot/rule/skip/.default=.3\normalbaselineskip,%
1275     foot/rule/skip/.blank=\z@,%
1276     foot/ruleskip/.alias*=foot/rule/skip,%           => puis les alias&style: /.alias 2
1277     footruleskip/.alias*=foot/rule/skip,%           crée aussi
1278 font/.style={head/font={#1},foot/font={#1}},%
1279     font+/.style={head/font+={#1},foot/font+={#1}},%
1280     font*/.alias=font+,%
1281 head/.style={head/left={#1},head/center={#1},head/right={#1}},%
1282     head/font/.style={head/left/font={#1},head/center/font={#1},head/right/font={#1}},%
1283     head/font+/.style={head/left/font+={#1},head/center/font+={#1},head/right/font+={#1}},%
1284     head/font*/.alias=head/font+,%
1285 nohead/.style={head=,head/font=,head/rule=},
1286 foot/.style={foot/left={#1},foot/center={#1},foot/right={#1}},%
1287     foot/font/.style={foot/left/font={#1},foot/center/font={#1},foot/right/font={#1}},%
1288     foot/font+/.style={foot/left/font+={#1},foot/center/font+={#1},foot/right/font+={#1}},%
1289     foot/font*/.alias=foot/font+,%
1290 nofoot/.style={foot=,foot/font=,foot/rule=},
1291 left/.style={head/left={#1},foot/left={#1}},%
1292     left/font/.style={head/left/font={#1},foot/left/font={#1}},%
1293     left/font+/.style={head/left/font+={#1},foot/left/font+={#1}},%
1294     left/font*/.alias=left/font+,%
1295 center/.style={head/center={#1},foot/center={#1}},%
1296     center/font/.style={head/center/font={#1},foot/center/font={#1}},%
1297     center/font+/.style={head/center/font+={#1},foot/center/font+={#1}},%
1298     center/font*/.alias=center/font+,%
1299 right/.style={head/right={#1},foot/right={#1}},%
1300     right/font/.style={head/right/font={#1},foot/right/font={#1}},%
1301     right/font+/.style={head/right/font+={#1},foot/right/font+={#1}},%
1302     right/font*/.alias=right/font+,%
1303 head/left/.style={head/left/even={#1},head/left/odd={#1}},%
1304     head/left/font/.style={head/left/even/font={#1},head/left/odd/font={#1}},%
1305     head/left/font+/.style={head/left/even/font+={#1},head/left/odd/font+={#1}},%
1306     head/left/font*/.alias=head/left/font+,%
1307 head/center/.style={head/center/even={#1},head/center/odd={#1}},%
1308     head/center/font/.style={head/center/even/font={#1},head/center/odd/font={#1}},%
1309     head/center/font+/.style={head/center/even/font+={#1},head/center/odd/font+={#1}},%
1310     head/center/font*/.alias=head/center/font+,%
1311 head/right/.style={head/right/even={#1},head/right/odd={#1}},%
1312     head/right/font/.style={head/right/even/font={#1},head/right/odd/font={#1}},%
1313     head/right/font+/.style={head/right/even/font+={#1},head/right/odd/font+={#1}},%
1314     head/right/font*/.alias=head/right/font+,%
1315 foot/left/.style={foot/left/even={#1},foot/left/odd={#1}},%
1316     foot/left/font/.style={foot/left/even/font={#1},foot/left/odd/font={#1}},%
1317     foot/left/font+/.style={foot/left/even/font+={#1},foot/left/odd/font+={#1}},%
1318     foot/left/font*/.alias=foot/left/font+,%
1319 foot/center/.style={foot/center/even={#1},foot/center/odd={#1}},%
1320     foot/center/font/.style={foot/center/even/font={#1},foot/center/odd/font={#1}},%
1321     foot/center/font+/.style={foot/center/even/font+={#1},foot/center/odd/font+={#1}},%
1322     foot/center/font*/.alias=foot/center/font+,%
1323 foot/right/.style={foot/right/even={#1},foot/right/odd={#1}},%
1324     foot/right/font/.style={foot/right/even/font={#1},foot/right/odd/font={#1}},%
1325     foot/right/font+/.style={foot/right/even/font+={#1},foot/right/odd/font+={#1}},%
1326     foot/right/font*/.alias=foot/right/font+,%
1327 head/left/odd/.fancy@hf=olh,%

```

```

1328   head/left/odd/font/.store as code,%
1329     odd/head/left/.alias=head/left/odd,%
1330     odd/head/left/font/.alias*=head/left/odd/font,%
1331     head/left/.alias=head/left/odd,%
1332     head/left/font/.alias*=head/left/odd/font,%
1333   head/center/odd/.fancy@hf=och,%
1334   head/center/odd/font/.store as code,%
1335     odd/head/center/.alias=head/center/odd,%
1336     odd/head/center/font/.alias*=head/center/odd/font,%
1337     head/center/.alias=head/center/odd,%
1338     head/center/font/.alias*=head/center/odd/font,%
1339   head/right/odd/.fancy@hf=orh,%
1340   head/right/odd/font/.store as code,%
1341     odd/head/right/.alias=head/right/odd,%
1342     odd/head/right/font/.alias*=head/right/odd/font,%
1343     head/right/.alias=head/right/odd,%
1344     head/right/font/.alias*=head/right/odd/font,%
1345   head/left/even/.fancy@hf=elh,%
1346   head/left/even/font/.store as code,%
1347     even/head/left/.alias=head/left/even,%
1348     even/head/left/font/.alias*=head/left/even/font,%
1349     head/even/left/.alias=head/left/even,%
1350     head/even/left/font/.alias*=head/left/even/font,%
1351   head/center/even/.fancy@hf=ech,%
1352   head/center/even/font/.store as code,%
1353     even/head/center/.alias=head/center/even,%
1354     even/head/center/font/.alias*=head/center/even/font,%
1355     head/even/center/.alias=head/center/even,%
1356     head/even/center/font/.alias*=head/center/even/font,%
1357   head/right/even/.fancy@hf=erh,%
1358   head/right/even/font/.store as code,%
1359     even/head/right/.alias=head/right/even,%
1360     even/head/right/font/.alias*=head/right/even/font,%
1361     head/even/right/.alias=head/right/even,%
1362     even/head/right/font/.alias*=head/right/even/font,%
1363   foot/left/odd/.fancy@hf=olf,%
1364   foot/left/odd/font/.store as code,%
1365     odd/foot/left/.alias=foot/left/odd,%
1366     odd/foot/left/font/.alias*=foot/left/odd/font,%
1367     foot/odd/left/.alias=foot/left/odd,%
1368     foot/odd/left/font/.alias*=foot/left/odd/font,%
1369   foot/center/odd/.fancy@hf=ocf,%
1370   foot/center/odd/font/.store as code,%
1371     odd/foot/center/.alias=foot/center/odd,%
1372     odd/foot/center/font/.alias*=foot/center/odd/font,%
1373     foot/odd/center/.alias=foot/center/odd,%
1374     foot/odd/center/font/.alias*=foot/center/odd/font,%
1375   foot/right/odd/.fancy@hf=orf,%
1376   foot/right/odd/font/.store as code,%
1377     odd/foot/right/.alias=foot/right/odd,%
1378     odd/foot/right/font/.alias*=foot/right/odd/font,%
1379     odd/foot/right/.alias=foot/right/odd,%
1380     odd/foot/right/font/.alias*=foot/right/odd/font,%
1381   foot/left/even/.fancy@hf=elf,%
1382   foot/left/even/font/.store as code,%
1383     even/foot/left/.alias=foot/left/even,%
1384     even/foot/left/font/.alias*=foot/left/even/font,%
1385     foot/even/left/.alias=foot/left/even,%
1386     foot/even/left/font/.alias*=foot/left/even/font,%
1387   foot/center/even/.fancy@hf=ecf,%
1388   foot/center/even/font/.store as code,%
1389     even/foot/center/.alias=foot/center/even,%
1390     even/foot/center/font/.alias*=foot/center/even/font,%

```

```

1391         foot/even/center/.alias=foot/center/even,%
1392         foot/even/center/font/.alias*=foot/center/even/font,%
1393         foot/right/even/.fancy@hf=erf,%
1394         foot/right/even/font/.store as code,%
1395             even/foot/right/.alias=foot/right/even,%
1396             even/foot/right/font/.alias*=foot/right/even/font,%
1397             foot/even/right/.alias=foot/right/even,%
1398                 foot/even/right/font/.alias*=foot/right/even/font,%
1399             offset/.style={head/offset={#1},foot/offset={#1}},%
1400                 offset+.style={head/offset+={#1},foot/offset+={#1}},%
1401                 offset-.style={head/offset-={#1},foot/offset-={#1}},%
1402                 offset*/.style={head/offset*={#1},foot/offset*={#1}},%
1403             head/offset/.style={head/left/offset={#1},head/right/offset={#1}},%
1404                 head/offset+/.style={head/left/offset+={#1},head/right/offset+={#1}},%
1405                 head/offset-/.style={head/left/offset-={#1},head/right/offset-={#1}},%
1406                 head/offset*/.style={head/left/offset*={#1},head/right/offset*={#1}},%
1407                     offset/head/.alias*=head/offset,%
1408             foot/offset/.style={foot/left/offset={#1},foot/right/offset={#1}},%
1409                 foot/offset+/.style={foot/left/offset+={#1},foot/right/offset+={#1}},%
1410                 foot/offset-/.style={foot/left/offset-={#1},foot/right/offset-={#1}},%
1411                 foot/offset*/.style={foot/left/offset*={#1},foot/right/offset*={#1}},%
1412                     offset/foot/.alias*=foot/offset,%
1413             left/offset/.style={foot/left/offset={#1},head/left/offset={#1}},%
1414                 left/offset+/.style={foot/left/offset+={#1},head/left/offset+={#1}},%
1415                 left/offset-/.style={foot/left/offset-={#1},head/left/offset-={#1}},%
1416                 left/offset*/.style={foot/left/offset*={#1},head/left/offset*={#1}},%
1417                     offset/left/.alias*=left/offset,%
1418             right/offset/.style={foot/right/offset={#1},head/right/offset={#1}},%
1419                 right/offset+/.style={foot/right/offset+={#1},head/right/offset+={#1}},%
1420                 right/offset-/.style={foot/right/offset-={#1},head/right/offset-={#1}},%
1421                 right/offset*/.style={foot/right/offset*={#1},head/right/offset*={#1}},%
1422                     offset/right/.alias*=right/offset,%
1423             head/left/offset/.style={head/left/even/offset={#1},head/left/odd/offset={#1}},%
1424                 head/left/offset+/.style={head/left/even/offset+={#1},head/left/odd/offset+={#1}},%
1425                 head/left/offset-/.style={head/left/even/offset-={#1},head/left/odd/offset-={#1}},%
1426                 head/left/offset*/.style={head/left/even/offset*={#1},head/left/odd/offset*={#1}},%
1427                     offset/head/left/.alias*=head/left/offset,%
1428                     offset/left/head/.alias*=head/left/offset,%
1429             head/right/offset/.style={head/right/even/offset={#1},head/right/odd/offset={#1}},%
1430                 head/right/offset+/.style={head/right/even/offset+={#1},head/right/odd/offset+={#1}},%
1431                 head/right/offset-/.style={head/right/even/offset-={#1},head/right/odd/offset-={#1}},%
1432                 head/right/offset*/.style={head/right/even/offset*={#1},head/right/odd/offset*={#1}},%
1433                     offset/head/right/.alias*=head/right/offset,%
1434                     offset/right/head/.alias*=head/right/offset,%
1435             foot/left/offset/.style={foot/left/even/offset={#1},foot/left/odd/offset={#1}},%
1436                 foot/left/offset+/.style={foot/left/even/offset+={#1},foot/left/odd/offset+={#1}},%
1437                 foot/left/offset-/.style={foot/left/even/offset-={#1},foot/left/odd/offset-={#1}},%
1438                 foot/left/offset*/.style={foot/left/even/offset*={#1},foot/left/odd/offset*={#1}},%
1439                     offset/foot/left/.alias*=foot/left/offset,%
1440                     offset/left/foot/.alias*=foot/left/offset,%
1441             foot/right/offset/.style={foot/right/even/offset={#1},foot/right/odd/offset={#1}},%
1442                 foot/right/offset+/.style={foot/right/even/offset+={#1},foot/right/odd/offset+={#1}},%
1443                 foot/right/offset-/.style={foot/right/even/offset-={#1},foot/right/odd/offset-={#1}},%
1444                 foot/right/offset*/.style={foot/right/even/offset*={#1},foot/right/odd/offset*={#1}},%
1445                     offset/foot/right/.alias*=foot/right/offset,%
1446                     offset/right/foot/.alias*=foot/right/offset,%
1447             head/left/even/offset/.store in skip cs=f@ncy0@elh,%
1448                 head/left/even/offset/.append code=\fancy@setoffs,%
1449                     offset/head/left/even/.alias*=head/left/even/offset,%
1450             head/left/odd/offset/.store in skip cs=f@ncy0@olh,%
1451                 head/left/odd/offset/.append code=\fancy@setoffs,%
1452                     offset/head/left/odd/.alias*=head/left/odd/offset,%
1453                     offset/head/left/odd/.alias*=head/left/odd/offset,%

```

```

1454     head/right/even/offset/.store in skip cs=f@ncy0@erh,%
1455         head/right/even/offset/.append code=\fancy@setoffs,%
1456             offset/head/right/even/.alias*=head/right/even/offset,%
1457     head/right/odd/offset/.store in skip cs=f@ncy0@orh,%
1458         head/right/odd/offset/.append code=\fancy@setoffs,%
1459             offset/head/right/odd/.alias*=head/right/odd/offset,%
1460     foot/left/even/offset/.store in skip cs=f@ncy0@elf,%
1461         foot/left/even/offset/.append code=\fancy@setoffs,%
1462             offset/foot/left/even/.alias*=foot/left/even/offset,%
1463     foot/left/odd/offset/.store in skip cs=f@ncy0@olf,%
1464         foot/left/odd/offset/.append code=\fancy@setoffs,%
1465             offset/foot/left/odd/.alias*=foot/left/odd/offset,%
1466     foot/right/even/offset/.store in skip cs=f@ncy0@erf,%
1467         foot/right/even/offset/.append code=\fancy@setoffs,%
1468             offset/foot/right/even/.alias*=foot/right/even/offset,%
1469     foot/right/odd/offset/.store in skip cs=f@ncy0@orf,%
1470         foot/right/odd/offset/.append code=\fancy@setoffs,%
1471             offset/foot/right/odd/.alias*=foot/right/odd/offset,%
1472     reset/.style={norule,head=,foot=,offset=\z@,foot/rule/skip,rule/font=,font=},%
1473         reset/.value forbidden,%
1474 }% keys for \pagesetup (stored into token \itf@tk)
1475 \AtBeginDocument{%
1476 \ifdim\headwidth<\z@
1477 %% This catches the case that \headwidth hasn't been initialized and the
1478 %% case that the user added something to \headwidth in the expectation that
1479 %% it was initialized to \textwidth. We compensate this now. This loses if
1480 %% the user intended to multiply it by a factor. But that case is more
1481 %% likely done by saying something like \headwidth=1.2\textwidth.
1482 %% The doc says you have to change \headwidth after the first call to
1483 %% \pagestyle{fancy}. This code is just to catch the most common cases were
1484 %% that requirement is violated.
1485     \global\advance\headwidth123456789sp\global\advance\headwidth\textwidth
1486 \fi}
1487 </fancyhdr>

```

## 21.10 † marks

```

1488 <*marks>
1489 \ProvidesPackage{interfaces-marks}
1490     [2010/10/27 v1.2 easier use of e-TeX marks (FC)]
1491 \RequirePackage{interfaces-base}

```

\marksthe

```

1492 \newrobustcmd\marksthe[2]{%
1493     \ifcsname marks@\#1\endcsname\else
1494         \itf@marks@new{\#1}%
1495     \fi
1496     \expandafter\marks\csname marks@\#1\endcsname{\unexpanded{\#2}}%
1497 }% \marksthe

```

\marksthecs

```

1498 \newrobustcmd*\marksthecs[2]{%
1499     \ifcsname marks@\#1\endcsname\else
1500         \itf@marks@new{\#1}%
1501     \fi
1502     \ifcsundef{\#2}%
1503         {}
1504         {\expandafter\marks\csname marks@\#1\endcsname{%
1505             \unexpanded\expandafter\expandafter\expandafter\expandafter{%
1506                 \csname\#2\endcsname}}}%
1507 }% \marksthecs

```

\itf@marks@new

```

1508 \def\itf@marks@new#1{%
1509   \expandafter\newmarks\csname marks##1\endcsname
1510   \PackageInfo{interfaces-marks}{%
1511     {New marks '#1' defined}}
1512 }% \itf@marks@warning

```

\thefirstmarks

\thebotmarks

\thetopmarks

```

1513 \newcommand*\thefirstmarks[1]{%
1514   \ifcsname marks##1\endcsname
1515     \expandafter\firstmarks\csname marks##1\endcsname
1516   \fi
1517 }% \thefirstmarks
1518 \newcommand*\thebotmarks[1]{%
1519   \ifcsname marks##1\endcsname
1520     \expandafter\botmarks\csname marks##1\endcsname
1521   \fi
1522 }% \thebotmarks
1523 \newcommand*\thetopmarks[1]{%
1524   \ifcsname marks##1\endcsname
1525     \expandafter\topmarks\csname marks##1\endcsname
1526   \fi
1527 }% \thetopmarks

```

\getthemarks

```

1528 %% \getthemarks\firstmarks{section}\macro
1529 \newrobustcmd*\getthemarks[3]{%
1530   \ifcsname marks##2\endcsname
1531     \expandafter\itf@getthemarks\expandafter##1\csname marks##2\endcsname{##3}%
1532   \else \let##3@\undefined
1533   \fi
1534 }% \getthemarks
1535 \def\itf@getthemarks#1#2#3{%
1536   \expandafter\def\expandafter##3\expandafter{##1##2}%
1537 }% \itf@getthemarks

```

\getthefirstmarks

\getthebotmarks

\getthetopmarks

```

1538 \newrobustcmd*\getthefirstmarks[2]{\getthemarks\firstmarks{##1}{##2}}
1539 \newrobustcmd*\getthebotmarks[2]{\getthemarks\botmarks{##1}{##2}}
1540 \newrobustcmd*\getthetopmarks[2]{\getthemarks\topmarks{##1}{##2}}

```

\ifmarksequal

```

1541 %% \ifmarksequal\firstmarks\botmarks{subsection}{ true }{ false }
1542 \newrobustcmd*\ifmarksequal[3]{%
1543   \begin{group}
1544     \getthemarks{##1}{##3}@tempa
1545     \getthemarks{##2}{##3}@tempb
1546   \expandafter\endgroup
1547   \csname ltx@ifdefined@tempa\ifdefined@tempb
1548     \ifx@tempa@tempb first\else
1549       second\fi\else
1550       second\fi\else
1551       second\fi
1552   oftwo\endcsname
1553 }% \ifmarksequal

```

```
1554 </marks>
```

## 21.11 † Interface ☞ `tocloft`: \tocsetup, \lofsetup and \lotsetup

```
1555 <*tocloft>
1556 \ProvidesPackage{interfaces-tocloft}
1557   [2010/10/27 v1.2 pgfkeys macros for tocloft (FC)]
1558 \RequirePackage{interfaces-base}
1559 \itf@Check@Package{tocloft}
1560 \itf@pgfkeys
```

\tocsetup The user commands.

```
\lofsetup
\lotsetup
\listofsetup
1561 \newrobustcmd\tocsetup[1]{%
1562   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/toc}{#1}}%
1563 }% \tocsetup
1564 \newrobustcmd\lofsetup[1]{%
1565   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/lof}{#1}}%
1566 }% \lofsetup
1567 \newrobustcmd\lotsetup[1]{%
1568   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/lot}{#1}}%
1569 }% \lotsetup
1570 \newrobustcmd\listofsetup[2]{%
1571   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/#1}{#2}}%
1572 }% \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`.

```
1573 \def\itf@cft@newlistof#1{%
1574   \pgfqkeys{/itf/tocloft/#1}{%
1575     pagestyle/.code=\tocloftpagestyle{##1},%
1576     parskip/.store in skip cs=cftparskip,%
1577     parsep/.alias=parskip,%
1578     mark/.store in cs=cftmark#1,%
1579     marks/.alias=mark,%
1580     label/.store as code,%
1581     ref/.store as code,%
1582     ref/.initial=\csname cft#1title\endcsname,% overwritten (toc,lof,lot)
1583     ref/.default=\csname cft#1title\endcsname,% overwritten (toc,lof,lot)
1584     ref/.blank=\pgfkeysnovalue,%
1585     title/.store as code in cs=cft#1title,% overwritten below for toc, lof and lot
1586     title/top/.store in skip cs=cftbefore#1titleskip,%
1587     beforetitleskip/.alias*=title/top,%
1588     title/bottom/.store in skip cs=cftafter#1titleskip,%
1589     aftertitleskip/.alias*=title/bottom,%
1590     title/font/.store as code in cs=cft#1titlefont,%
1591     title/format/.alias*=title/font,%
1592     title/after/.store as code in cs=cftafter#1title,%
1593     dotsep/.code=\itf@units@mu\itf@musk{##1}% converts to mu
1594     \edef\cftdotsep{\itf@strip@mu\itf@musk},%
1595     dotsep/.default=4.5mu,%
1596     dotsep/.blank=\cftnodots,%
1597     dot/.store in=\cftdot,%
1598     before/.store as code,%
1599     after/.store as code,%
1600     columns/before@hook/.is private=interfaces-tocloft,%
1601     columns/after@hook/.is private=interfaces-tocloft,%
1602     columns/before@hook/.initial=,%
1603     columns/after@hook/.initial=,%
1604     columns/.code={\ltx@ifpackageloaded{multicol}{%
1605       \pgfkeysEsetvalue\pgfkeyscurrentkey{\number##1}%
1606       \ifnum\number##1>\@ne
```

```

1607          \pgfkeysgetvalue{/itf/tocloft/#1/columns/before@hook}\itf@tp
1608          \ifx\itf@tp@\empty
1609              \pgfkeyssetvalue{/itf/tocloft/#1/columns/before@hook}{%
1610                  \begingroup
1611                      \multicols\pgfkeysvalueof{/itf/tocloft/#1/multicols}\relax
1612                      \begin{multicols}{\pgfkeysvalueof{/itf/tocloft/#1/columns}}%
1613                          \columnseprule\pgfkeysvalueof{/itf/tocloft/#1/columns/rule}\relax
1614                          \columnsep\pgfkeysvalueof{/itf/tocloft/#1/columns/sep}\relax
1615                          \pgfkeysgetvalue{/itf/tocloft/#1/columns/rule/color}\columnseprulecolor
1616                          \pgfkeysvalueof{/itf/tocloft/#1/multicols/begin}%
1617                  }%
1618                  \pgfkeyssetvalue{/itf/tocloft/#1/columns/after@hook}{%
1619                      \pgfkeysvalueof{/itf/tocloft/#1/multicols/end}%
1620                      \end{multicols}
1621                      \endgroup
1622                  }%
1623          \fi
1624      \else
1625          \pgfkeyslet\pgfkeyscurrentkey\@ne
1626          \pgfkeyslet{/itf/tocloft/#1/columns/before@hook}\@empty
1627          \pgfkeyslet{/itf/tocloft/#1/columns/after@hook}\@empty
1628      \fi}
1629      {\PackageError{interfaces-tocloft}
1630          {Key ‘columns’: package multicol is required}\@eha}%
1631  },%
1632  columns/.default=2,columns/.blank=1,%
1633      multicols/.alias=columns,%
1634      multicols/.alias=columns,%
1635  twocolumns/.true=\pgfkeysalso{columns=2},%
1636  twocolumns/.false=\pgfkeysalso{columns=1},%
1637  columns/rule/.store as dimen,%
1638  columns/rule/.default=\columnseprule,%
1639  columns/rule,% assign default
1640      columnseprule/.alias*=columns/rule,%
1641      columns/rule/width/.alias*=columns/rule,%
1642  columns/sep/.store as dimen,%
1643  columns/sep/.default=\columnsep,%
1644  columns/sep/.blank=10\p@,%
1645  columns/sep,% assign default
1646      columnsep/.alias*=columns/sep,%
1647  columns/rule/color/.store as color command,%
1648  columns/rule/color,% assign default (\normalcolor)
1649      columnseprulecolor/.alias*=columns/rule/color,%
1650      columns/rulecolor/.alias*=columns/rule/color,%
1651  multicols/.store as glue,%
1652      multicols/.default=0pt plus4pt minus3pt,%
1653      multicols,% sets default
1654  multicols/begin/.initial=,multicols/begin/.default=,% 
1655      multicols/before/.alias=multicols/begin,%
1656  multicols/end/.initial=,multicols/end/.default=,% 
1657      multicols/after/.alias=multicols/end,%
1658  }% \pgfkeys
1659  \itf@AfterPackage{interfaces-bookmark}{%
1660  \pgfqkeys{/itf/tocloft/#1}{%
1661      bookmark/.is multiple choice,%
1662          bookmark/bold/.true=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{\bold=true},%
1663          bookmark/bold/.false=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{\bold=false},%
1664          bookmark/italic/.true=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{\italic=true},%
1665          bookmark/italic/.false=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{\italic=false},%
1666          bookmark/color/.code=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{}{\color={##1}},%
1667          bookmark/level/.code=\pgfkeysEaddvalue{/itf/tocloft/#1/bookmark}{}{\level={%
1668              \ifcsname toclevel@\#\#1\endcsname
1669                  \csname toclevel@\#\#1\endcsname

```

```

1670           \else##1\fi}}},%
1671     bookmark/text/.store as code,%
1672       bookmark/text/.default=\contentsname,%
1673       bookmark/.unknown/.code=\pgfkeysEaddvalue{/itf/tocloft/#1/bookmark}{}{\pgfkeyscurren
1674     bookmark/.initial=,%
1675 }% \pgfkeys
1676 }% AfterPackage interfaces-bookmark
1677 %% while making a format \jobname is expanded to the name of the format... Therefore, the
1678 \AtEndPreamble{%
1679   \BeforeFile{\jobname.#1}{%
1680     \pgfkeysvalueof{/itf/tocloft/#1/before}%
1681     \pgfkeysvalueof{/itf/tocloft/#1/columns/before@hook}%
1682   }%
1683   \AfterFile{\jobname.#1}{%
1684     \pgfkeysvalueof{/itf/tocloft/#1/columns/after@hook}%
1685     \pgfkeysvalueof{/itf/tocloft/#1/after}%
1686   }%
1687 }%
1688 \ifcsname #1setup\endcsname\else
1689   \pgfqkeys{/itf/tocloft/#1}{%
1690     pagenumbers/.true=@cftpnumon{#1},%
1691     pagenumbers/.false=@cftpnumoff{#1},%
1692       pagename/.alias=pagenumbers
1693   }%
1694   \expandafter\pretocmd\csname@cftmake#1title\expandafter\endcsname
1695     \expandafter{\expandafter\itf@tocloft\csname cft#1title\endcsname{#1}{listof#1}}
1696   {\PackageInfo{interfaces-tocloft}
1697     {Command \expandafter\string\csname@cftmake#1title\endcsname
1698       successfully augmented}}
1699   {\PackageWarning{interfaces-tocloft}
1700     {Command \expandafter\string\csname@cftmake#1title\endcsname
1701       space could not be patched
1702       \MessageBreak Keys label and bookmark won't work for \string\listofsetup}%
1703   \fi
1704 }% \itf@cft@newlistof

```

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

```

1705 \itf@cft@newlistof{toc}
1706 \itf@cft@newlistof{lof}
1707 \itf@cft@newlistof{lot}

1708 %% Specific to toc only -----
1709 \pgfqkeys{/itf/tocloft/toc}{%
1710   depth/.defcounter=tocdepth,%
1711     depth/.value required,%
1712     tocdepth/.alias=depth,%
1713   title/.store as code in cs=contentsname,%
1714     contentsname/.alias=title,%
1715   ref/.initial=\contentsname,%
1716     ref/.default=\contentsname,%
1717   pagenumbers/.style={% boolean key
1718     part/pagenumbers={#1}, chapter/pagenumbers={#1}, section/pagenumbers={#1}, %
1719     subsubsection/pagenumbers={#1}, subsection/pagenumbers={#1}, %
1720     paragraph/pagenumbers={#1}, subparagraph/pagenumbers={#1}, %
1721   pagename/.alias=pagenumbers,%
1722 }%
1723 %% Specific to lof and lot -----
1724 \pgfqkeys{/itf/tocloft/lof}{%
1725   title/.store as code in cs=listfigurename,%
1726     listfigurename/.alias=title,%
1727   ref/.initial=\listfigurename,%
1728     ref/.default=\listfigurename,%
1729 }%

```

```

1730 \pgfqkeys{/itf/tocloft/lot}{%
1731   title/.store as code in cs=listtablename,%
1732   listtablename/.alias=title,%
1733   ref/.initial=\listtablename,%
1734   ref/.default=\listtablename,%
1735 }%
1736 \if@cftsubfigopt
1737 \pgfqkeys{/itf/tocloft}{%
1738   lof/pagenumbers/.style={figure/pagenumbers={#1},subfigure/pagenumbers={#1}},%
1739   lof/pagenumber/.alias=lof/pagenumbers,%
1740   lot/pagenumbers/.style={figure/pagenumbers={#1},subfigure/pagenumbers={#1}},%
1741   lot/pagenumber/.alias=lot/pagenumbers,%
1742 }%
1743 \else
1744 \pgfqkeys{/itf/tocloft}{%
1745   lof/pagenumbers/.style={figure/pagenumbers={#1}},%
1746   lof/pagenumber/.alias=lof/pagenumbers,%
1747   lot/pagenumbers/.style={figure/pagenumbers={#1}},%
1748   lot/pagenumber/.alias=lot/pagenumbers,%
1749 }%
1750 \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`.

```

1751 \def\itf@cft@name#1.#2.#3@nil{cft#1\itf@cft@secname{#2}#3}
1752 \def\itf@cft@secname#1{%
1753   \expandafter\itf@cft@sec@name\csname#1\endcsname{#1}%
1754 }% \itf@cft@secname
1755 \def\itf@cft@sec@name#1#2{%
1756   \ifx#1\chapter chap\else
1757     \ifx#1\section sec\else
1758       \ifx#1\subsection subsec\else
1759         \ifx#1\subsubsection subsubsec\else
1760           \ifx#1\paragraph para\else
1761             \ifx#1\subparagraph subpara\else
1762               \ifx#1\figure fig\else
1763                 \ifx#1\subsubfigure subfig\else
1764                   \ifx#1\table tab\else
1765                     \ifx#1\subtable subtab\else
1766                       #2\fi
1767                     \fi\fi\fi\fi\fi\fi
1768 }% \itf@cft@section@name

```

### 21.11.1 Specific handlers for tocloft

```

1769 %% tocloft setup for tables sections :
1770 \pgfkeys{%
1771   /handlers/.cft@def/.style={\pgfkeyscurrentpath/.store in cs=\itf@cft@name #1@nil},%
1772   /handlers/.cft@code/.style={\pgfkeyscurrentpath/.store as code in cs=\itf@cft@name #1@nil}
1773   /handlers/.cft@length/.style={\pgfkeyscurrentpath/.store in skip cs=\itf@cft@name #1@nil}
1774 }

```

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

```

1775 \def\itf@cft@newsection#1#2{%
#1=part / chap / sec / subsec etc., #2 = #1 = toc | lof | lot
1776   \pgfqkeys{/itf/tocloft/#2/#1}{%
1777     skip/.cft@length=before.#1.skip,%  

1778     beforeskip/.alias*=skip,%  

1779     top/.alias*=skip,%

```

```

1780     indent/.cft@length=.#1.indent,%
1781     font/.cft@code=.#1.font,%
1782     font/.append code={\pgfkeysalso{/itf/tocloft/#2/#1/pagenumber/font={##1}}},%
1783         format/.alias*=font,%
1784         number/width/.cft@length=.#1.numwidth,%
1785             numwidth/.alias*=number/width,%
1786             presnum/.cft@code=.#1.presnum,%
1787                 number/before/.alias*=presnum,%
1788                 aftersnum/.cft@code=.#1.aftersnum,%
1789                     number/after/.alias*=aftersnum,%
1790                     aftersnumb/.cft@code=.#1.aftersnumb,%
1791                         title/before/.alias*=aftersnumb,%
1792                         leader/.cft@def=.#1.leader,%
1793                             leader/.default=\cftdotfill{\csname\itf@cft@name.#1.dotsep@nil\endcsname},%
1794                             leaders/.alias=leader,%
1795                             dotsep/.code={\itf@units@mu\itf@musk{##1}%
1796                                 \csedef{\itf@cft@name.#1.dotsep@\nil}{\itf@strip@mu\itf@musk}},%
1797                             dotsep/.default=\cftdotsep,%
1798                             dotsep/.blank=\cftnodots,%
1799                             nodots/.code=\csdef{\itf@cft@name.#1.dotsep@\nil}{\cftnodots},%
1800                             nodots/.value forbidden,%
1801                             pagefont/.cft@code=.#1.pagefont,%
1802                                 page/font/.alias*=pagefont,%
1803                                 pagename/font/.alias*=pagefont,%
1804                                 afterpnum/.cft@code=.#1.afterpnum,%
1805                                     page/after/.alias*=afterpnum,%
1806                                     pagename/after/.alias*=afterpnum,%
1807                                     pagenumbers/.on=@cftpnumon{\itf@cft@secname{#1}},%
1808                                     pagenumbers/.off=@cftpnumoff{\itf@cft@secname{#1}},%
1809                                         pagename/.alias=pagenumbers,%
1810 }% \pgfqkeys
1811 }% \itf@cft@newsection

1812 \@for\itf@tp:=part,chap,chapter,sec,section,subsec,subsection,%
1813     subsubsec,subsubsection,para,paragraph,subpara,subparagraph%
1814     \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{toc}}%
1815 \if@cftsubfigopt % @ifpackagewith{tocloft}{subfig} ??
1816   \@for\itf@tp:=fig,figure,subfig,subfigure%
1817     \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lof}}%
1818   \@for\itf@tp:=tab,table,subtab,subtable%
1819     \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lot}}%
1820 \else
1821   \@for\itf@tp:=fig,figure%
1822     \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lof}}%
1823   \@for\itf@tp:=tab,table%
1824     \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lot}}%
1825 \fi

@cftmaketocitle (patch)
@cftmakeloftitle (patch)
@cftmakelottitle (patch)
1826 \itf@patch\pretocmd@cftmaketocitle
1827   {\itf@tocloft\contentsname{toc}{tableofcontents}}
newlistentry (patch)
1828   {\PackageInfo{interfaces-tocloft}%
1829     {Command \string@\cftmaketocitle\space successfully augmented}%
1830     {\PackageWarning{interfaces-tocloft}%
1831       {Command \string@\cftmaketocitle\space could not be patched%
1832        \MessageBreak label and bookmark keys for \string@\cftmaketocitle\space won't work}}%
1833 \itf@patch\pretocmd@cftmakeloftitle
1834   {\itf@tocloft\listfigurename{lof}{listoffigures}}
1835   {\PackageInfo{interfaces-tocloft}%
1836     {Command \string@\cftmakeloftitle\space successfully augmented}%
1837     {\PackageWarning{interfaces-tocloft}%
1838       {Command \string@\cftmakeloftitle\space could not be patched}}}

```

```

1839      \MessageBreak label and bookmark keys for \string\lofsetup\space won't work}%
1840 \itf@patch\pretocmd@\cftmakelottitle
1841   {\itf@tocloft\listtablename{lot}{listoftables}}
1842   {\PackageInfo{interfaces-tocloft}
1843     {Command \string@\cftmakelottitle\space successfully augmented}%
1844   {\PackageWarning{interfaces-tocloft}
1845     {Command \string@\cftmakelottitle\space could not be patched
1846      \MessageBreak label and bookmark keys for \string\lotsetup\space won't work}%
1847 \itf@patch\apptocmd\newlistentry
1848   {\itf@cft@newlistof{\#2}}
1849   {\PackageInfo{interfaces-tocloft}
1850     {Command \string\newlistentry\space successfully augmented}%
1851   {\PackageWarning{interfaces-tocloft}
1852     {Command \string\newlistentry\space could not be patched
1853      \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.

```

1854 \def\itf@tocloft#1#2#3{%
1855   \expandafter\def\expandafter#1\expandafter{%
1856     \expandafter\itf@cft@label
1857       \expandafter#1\expandafter{\#1}{\#2}{\#3}}%
1858 }% \itf@tocloft
1859 \long\def\itf@cft@label#1#2#3#4{%
1860   \def#1{\#2}% restore original
1861   \begingroup
1862   \pgfkeysifdefined{/itf/tocloft/#3/bookmark}
1863   {
1864     \raisedhyperdef{interfaces}{\#4}{\#2}%
1865     \pgfkeysgetvalue{/itf/tocloft/#3/bookmark/text}\itf@tp
1866     \ifx\itf@tp\empty\else
1867       \pgfkeysgetvalue{/itf/tocloft/#3/bookmark}@tempa
1868       \expandafter\bookmark\expandafter[%%
1869         \expandafter{\@tempa,dest=interfaces.\#4}]%
1870         \pgfkeysvalueof{/itf/tocloft/#3/bookmark/text}}%
1871       \global\BKM@startatroottrue
1872     \fi
1873   {\#2}%
1874   \pgfkeysgetvalue{/itf/tocloft/#3/label}\itf@tp
1875   \ifx\itf@tp\empty\else
1876     \def\@currentHref{interfaces.\#4}%
1877     \pgfkeysgetvalue{/itf/tocloft/#3/ref}@currentlabel
1878     \def\@currentlabelname{\#2}%
1879     \label{\pgfkeysvalueof{/itf/tocloft/#3/label}}%
1880   \fi
1881   \endgroup
1882   \gdef\itf@tp{\def#1{\#2}\let\itf@tp\undefined}% restore original
1883   \aftergroup\itf@tp % simple group introduced by tocloft to print \contentsname
1884 }% \itf@cft@label

1885 </tocloft>

```

```

1886 /*makecell*/
1887 \ProvidesPackage{interfaces-makecell}
1888   [2010/10/27 v1.2 more control on tabular cells (FC)]
1889 \RequirePackage{interfaces-base}
1890 \itf@Check@Package{makecell}
1891 \edef\itf@AtEnd{\catcode126 \the\catcode126}% 126 = ~
1892 \AtEndOfPackage{\itf@AtEnd}
1893 \catcode126 \active % 126 = ~
1894 \itf@pgfkeys

```

## 21.12.1 `countvalues`

`\countvalues`

```

1895 \def\countvalues#1{\number\numexpr
1896   \expandafter\itf@countvalues\expandafter#1,\countvalues@nnil
1897 }% \countvalues
1898 \def\itf@countvalues#1#2,#3@nnil{%
1899   \ifx\countvalues#3%
1900     +#1\expandafter\remove@to@nnil
1901   \else
1902     +1\expandafter\itf@countvalues
1903   \fi#3@nnil
1904 }% \itf@countvalues

```

`\itf@getvalue`

```

1905 \def\itf@getvalue#1#2{%
1906   \romannumeral-`q\expandafter\itf@getv@lue\expandafter#1,\itf@getvalue@nil{#2-1}@nnil
1907 }% \itf@getvalue
1908 \def\itf@getv@lue#1#2,#3@nil#4@nnil{%
1909   \ifnum\numexpr#1=\numexpr#4\relax
1910     #2\expandafter\remove@to@nnil
1911   \else\ifx\itf@getvalue#3%
1912     \expandafter\expandafter\expandafter\remove@to@nnil
1913   \else
1914     \expandafter\expandafter\expandafter\itf@getv@lue
1915   \fi\fi{#1+1}#3@nil#4@nnil
1916 }% \itf@getv@lue

```

## 21.12.2 `\repeatcell`

`\repeatcell`

```

1917 \newif\ifitf@celltranspose
1918 \def\itf@ampactive{%
1919   \def\tab{&}{%
1920     \lccode`\~=\38\lowercase{\def~{\&}}{%
1921   }% \itf@ampactive
1922 \newrobustcmd*\repeatcell{\begingroup
1923   \let\tab\relax
1924   \catcode38\active
1925   \itf@repeatcell
1926 }% \repeatcell
1927 \newrobustcmd*\itf@repeatcell[3][]{%
1928   \ifblank{#1}{% alignment option ?
1929     {\let\itf@makecell@\firstofone{%
1930       {\def\itf@makecell##1{\noexpand\makecell[{#1}]{{##1}}}{%
1931         \countdef\itf@col=100% local

```

```

1932  \countdef\itf@row=101% local
1933  \countdef\itf@cnt=102% local
1934  \toksdef\itf@cell=1% local
1935  \itf@pgfkeys
1936  \pgfqkeys{/itf/makecell/repeatcell}{@reset={#2},#3}%
1937  \toks@{}% local
1938  \ifnum\itf@row>\z@ % number
1939    \edef\itf@rowloop{\noexpand\@whilenum\itf@row<\the\itf@row}%
1940    \def\itf@cell@init{\def\r{\row}\let\R@empty}%
1941  \else % list
1942    \def\itf@rowloop{\@for\r:=\itf@rows}%
1943    \expandafter\def\expandafter\itf@cell@init\expandafter{%
1944      \expandafter\def\expandafter\itf@cell@init\R\expandafter{%
1945        \expandafter\itf@getvalue\expandafter{\expandafter\@empty\itf@rows}}}%
1946  \fi
1947  \ifnum\itf@col>\z@ % number
1948    \edef\itf@colloop{\noexpand\@whilenum\itf@col<\the\itf@col}%
1949    \expandafter\def\expandafter\itf@cell@init\expandafter{%
1950      \itf@cell@init\def\c{\column}\let\C@empty}%
1951  \else % list
1952    \def\itf@colloop{\@for\c:=\itf@cols}%
1953    \expandafter\expandafter\expandafter\def
1954      \expandafter\expandafter\expandafter\itf@cell@init
1955      \expandafter\expandafter\expandafter{%
1956        \expandafter\itf@cell@init
1957        \expandafter\def\expandafter\itf@cell@init\C\expandafter{%
1958          \expandafter\itf@getvalue\expandafter{\expandafter\@empty\itf@cols}}}%
1959  \fi
1960  \itf@cell@init % in case they are used in formulas
1961  \ifitf@celltranspose
1962    \itf@col\z@
1963    \itf@colloop\do{%
1964      \ifnum\itf@col>\z@
1965        \pgfkeysifdefined{/itf/makecell/repeatcell/cr/row\the\itf@col}%
1966        {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr/row\the\itf@col}\itf@thecr}%
1967        {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr}\itf@thecr}%
1968        \toks@\expandafter\expandafter\expandafter{%
1969          \expandafter\the\expandafter\toks@\itf@thecr}%
1970    \fi
1971    \advance\itf@col\@ne
1972    \itf@row\z@
1973    \itf@rowloop\do{%
1974      \advance\itf@row\@ne
1975      \ifnum\itf@row>\@ne
1976        \toks@\expandafter\expandafter\expandafter{%
1977          \expandafter\the\expandafter\toks@&}%
1978        \fi
1979        \toks@\expandafter\expandafter\expandafter{%
1980          \expandafter\the\expandafter\toks@%
1981          \itf@cell@init\itf@ampactive}%
1982        \itf@cell@insert}%
1983      }%
1984    \else
1985      \itf@row\z@
1986      \itf@rowloop\do{%
1987        \ifnum\itf@row>\z@
1988          \pgfkeysifdefined{/itf/makecell/repeatcell/cr/row\the\itf@row}%
1989          {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr/row\the\itf@row}\itf@thecr}%
1990          {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr}\itf@thecr}%
1991          \toks@\expandafter\expandafter\expandafter{%
1992            \expandafter\the\expandafter\toks@\itf@thecr}%
1993          \fi
1994          \advance\itf@row\@ne
1995          \itf@col\z@

```

```

1995   \itf@colloop\do{%
1996     \advance\itf@col\@ne
1997     \ifnum\itf@col>\@ne
1998       \toks@\expandafter{\the\toks@@}%
1999     \fi
2000     \toks@\expandafter\expandafter\expandafter{%
2001       \expandafter\the\expandafter\toks@
2002       \itf@cell@init\itf@ampactive}%
2003     \itf@cell@insert}%
2004   }%
2005 \fi
2006 \itf@message{Code introduced by repeatcell in the \@currenvir:\the\toks@}%
2007 \expandafter\endgroup\the\toks@
2008 \ignorespaces
2009 }% \itf@repeatcell

```

\itf@message

```
2010 \let\itf@message@gobble
```

Now the keys for \repeatcell:

```

2011 \def\itf@cell@rowcolkey{%
2012   \removefromstart\pgfkeyscurrentname{row}%
2013   \ifin@\else
2014     \removefromstart\pgfkeyscurrentname{col}%
2015   \fi
2016   \ifin@
2017     \itf@number\itf@cnt\pgfkeyscurrentname\relax\@nnil
2018     {\pgfkeyslet\pgfkeyscurrentkey\pgfkeyscurrentvalue}%
2019     \in@false
2020   \fi
2021   \ifin@\else
2022     \pgfkeysvalueof{/errors/unknown key/.@cmd}\pgfkeyscurrentkey
2023                               \pgfkeyscurrentvalue
2024                               \pgfeov
2025   \fi
2026 }% \itf@cell@rowcolkey
2027 \pgfqkeys{/itf/makecell/repeatcell}{%
2028   font/.store in=\itf@cellfont,%
2029   font/.unknown/.code=\itf@cell@rowcolkey,%
2030   text/.store in=\itf@celltext,%
2031     text/.default=\ifdefined\C\column\fi,%
2032   text/.unknown/.code=\itf@cell@rowcolkey,%
2033   cr/.initial=\,,%
2034     cr/.default=\crcr,cr/.blank=\,,end/.alias=cr,%
2035   cr/.unknown/.code=\itf@cell@rowcolkey,%
2036   rows/.code=\itf@number\itf@row#1\relax\@nnil
2037     {\ifnum\itf@row<\z@\itf@row\@ne\fi}
2038     {\itf@row\z@
2039     \def\itf@rows{\#1}\itf@expandonceifcs\itf@rows},%
2040   rows/.default=1,rows/.blank=1,%
2041   @cols/.is private=interfaces-makecell,%
2042   column/.is private=interfaces-makecell,%
2043   row/.is private=interfaces-makecell,%
2044   transpose/.is if=itf@celltranspose,%
2045   .unknown/.code=%
2046     \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text \itf@cell@formulaerror{}\z@\fi
2047     \ifx\pgfkeyscurrentvalue\empty \itf@cell@formulaerror{}\@ne\fi
2048     \pgfkeysifdefined{\pgfkeyscurrentkey/@calc}{%
2049       {\itf@cell@formulaerror{}\tw@}
2050       {\pgfkeyslet{\pgfkeyscurrentkey/@calc}\relax}%
2051     \itf@cell@getformula{\#1}%

```

```

2052     \pgfkeysEaddvalue{/itf/makecell/repeatcell/@calc}{}
2053         {\the\toks@\expandafter\noexpand\csname\pgfkeyscurrentname\endcsname}%
2054         \expandafter\let\csname\pgfkeyscurrentname\endcsname\@undefined,% (order of evaluation
2055 @reset/.code=% #1 are the columns
2056     \itf@celltransposefalse
2057     \let\itf@celltext\@empty
2058     \let\itf@cellfont\@empty
2059     \pgfkeyssetvalue{/itf/makecell/repeatcell/cr}{\\}%
2060     \itf@row\@ne
2061     \pgfkeyssetvalue{/itf/makecell/repeatcell/@calc}{%
2062         \itf@cell@numcalc\itf@col\column
2063         \itf@cell@numcalc\itf@row\row}%
2064     \itf@number\itf@col#1\relax\@nnil
2065         {\ifnum\itf@col<\z@\itf@col\@ne\fi}
2066         {\itf@col\z@
2067             \def\itf@cols{\#1}\itf@expandonceifcs\itf@cols},%
2068 }% keys for \repeatcell

```

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

```

2069 \def\itf@cell@getformula#1{%
2070     \itf@cell@getf@rmula #1$$$ \itf@cell@getformula$\relax\@nnil
2071     \expandafter\ifx\expandafter\toks@\the\toks@
2072         \itf@cell@formulaerror{#1}\thr@@
2073     \fi
2074 }% \itf@cell@getformula
2075 \def\itf@cell@getf@rmula #1##2##3##4##5##6{%
2076     \toks@\{\toks@{%
2077         \ifx\itf@cell@getformula #4%
2078             \toks@\{\itf@cell@numcalc{#1}\}%
2079         \else\ifx $#4%
2080             \ifx\itf@cell@getformula #6%
2081                 \ifdefined\pgfmathparse\else\itf@cell@formulaerror{#2}4\fi
2082                 \toks@\{\itf@cell@pgfcalc{#2}\}%
2083             \else\ifx $#6%
2084                 \ifdefined\FPeval\else\itf@cell@formulaerror{#3}5\fi
2085                 \toks@\{\itf@cell@FPcalc{#3}\}%
2086             \fi\fi
2087             \fi\fi\remove@to@nnil
2088 }% \itf@cell@getf@rmula
2089 \def\itf@cell@formulaerror#1#2{%
2090     \def\itf@tp{#1}\@onelvel@sanitize\itf@tp
2091     \ifcase#2\@nil
2092         \PackageError{interfaces-makecell}{%
2093             {Missing formula for key \pgfkeyscurrentname!}\@ehd
2094         \or %1
2095         \PackageError{interfaces-makecell}{%
2096             {Computed key \pgfkeyscurrentname\space was left blank!}\@ehd
2097         \or %2
2098         \PackageError{interfaces-makecell}{%
2099             {Duplicate definition for computed key \pgfkeyscurrentname
2100             \MessageBreak\string\repeatcell\space will redefine the key}\@ehd
2101         \or %3
2102         \PackageError{interfaces-makecell}{%
2103             {Incorrect formula in \string\repeatcell
2104             \MessageBreak Check: \itf@tp
2105             \MessageBreak Take care of commas and equal signs...}\@ehd
2106         \or %4
2107         \PackageError{interfaces-makecell}{%
2108             {Formula: \itf@tp
2109             \MessageBreak requires \string\pgfmathparse:\space\string\usetikz{pgfmath}}\@ehd
2110         \or %5
2111         \PackageError{interfaces-makecell}{%

```

```
2112 {Formula: \itf@tp
2113   \MessageBreak requires \string\FPeval:\space\string\usepackage{fp}\}@ehd
2114 \else
2115 \PackageError{interfaces-makecell}
2116   {Impossible value!}\@ehd
2117 \fi
2118 }% \itf@cell@formulaerror
```

```
\itf@cell@calc
\itf@cell@numcalc
\itf@cell@pgfcalc
\itf@cell@FPcalc
2119 \protected\def\itf@cell@calc#1#2{%
2120   \edef#1{\def\noexpand#1{#2}}%
2121   \toks@\expandafter\expandafter\expandafter{%
2122     \expandafter\the\expandafter\toks@#1}%
2123   #1%
2124 }% \itf@cell@calc
2125 \protected\def\itf@cell@numcalc#1#2{%
2126   \itf@cell@calc#2{\number\numexpr#1}%
2127 }% \itf@cell@numcalc
2128 \protected\def\itf@cell@pgfcalc#1#2{%
2129   \pgfmathparse{#1}%
2130   \itf@cell@calc#2\pgfmathresult
2131 }% \itf@cell@pgfcalc
2132 \protected\def\itf@cell@FPcalc#1#2{%
2133   \begingroup
2134   \setbox\z@\vbox{\%FPeval introduces some weird code, hence the calculation in a \vbox
2135     \FPeval\FPresult{#1}%
2136     \global\let\FPresult\FPresult}%
2137   \endgroup
2138   \itf@cell@calc#2\FPresult
2139 }% \itf@cell@FPcalc
```

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

```
2140 \def\itf@cell@getifdefined#1#2{%
2141   \pgfkeysifdefined{/itf/makecell/repeatcell/#1}%
2142     {\pgfkeysgetvalue{/itf/makecell/repeatcell/#1}#2}%
2143     {\let#2\empty}%
2144 }%
2145 \def\itf@cell@insert{%
2146   \pgfkeysvalueof{/itf/makecell/repeatcell/@calc}%
2147   \ifx\R\empty\else
2148     \toks@\expandafter\expandafter\expandafter{%
2149       \expandafter\the\expandafter\toks@
2150       \expandafter\def\expandafter\r\expandafter{\r}}%
2151   \fi
2152   \ifx\C\empty\else
2153     \toks@\expandafter\expandafter\expandafter{%
2154       \expandafter\the\expandafter\toks@
2155       \expandafter\def\expandafter\c\expandafter{\c}}%
2156   \fi
2157   \itf@cell@getifdefined{font/row\the\itf@row}\itf@cellfontrow
2158   \itf@cell@getifdefined{font/col\the\itf@col}\itf@cellfontcol
2159   \itf@cell@getifdefined{text/row\the\itf@row}\itf@cellrow
2160   \itf@cell@getifdefined{text/col\the\itf@col}\itf@cellcol
2161   \toks@\expandafter\expandafter\expandafter{%
2162     \expandafter\the\expandafter\toks@\itf@cellfont}%
2163   \ifx\itf@cellrow\empty
2164     \ifx\itf@cellcol\empty
2165       \itf@cell\expandafter\expandafter\expandafter{%
2166         \expandafter\itf@cellfontcol\itf@cellfontrow}%
2167       \let\itf@tp\itf@celltext
2168     \else
2169       \itf@cell\expandafter{\itf@cellfontcol}%
2170       \let\itf@tp\itf@cellcol
2171     \fi
2172   \else
2173     \ifx\itf@cellcol\empty
2174       \itf@cell\expandafter{\itf@cellfontrow}%
2175       \let\itf@tp\itf@cellrow
2176     \else
```

```

2177      \itf@cell{}%
2178      \ifitf@celltranspose
2179      \edef\itf@tp{\noexpand\itf@diaghead(-2,1)
2180          {\expandonce\itf@cellfontrow\expandonce\itf@cellrow}
2181          {\expandonce\itf@cellfontcol\expandonce\itf@cellcol}}%
2182      \else
2183      \edef\itf@tp{\noexpand\itf@diaghead(-2,1)
2184          {\expandonce\itf@cellfontcol\expandonce\itf@cellcol}
2185          {\expandonce\itf@cellfontrow\expandonce\itf@cellrow}}%
2186      \fi
2187      \fi
2188      \fi
2189      \itf@cell\expandafter\expandafter\expandafter{%
2190          \expandafter\the\expandafter\itf@cell\itf@tp}%
2191      \edef\itf@tp{\itf@makecell{\the\itf@cell}}%
2192      \toks@\expandafter\expandafter\expandafter{%
2193          \expandafter\the\expandafter\toks@\itf@tp}%
2194 }% \itf@cell@insert

```

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

```

2195 \def\itf@diaghead(#1)#2#3{%
2196     \begingroup
2197     \edef\itf@tp{%
2198         \noexpand\diaghead({#1}){\unexpanded{#2 #3}}%
2199             {\unexpanded{#2}}%
2200             {\unexpanded{#3}}}}%
2201     \expandafter\endgroup\itf@tp
2202 }% \itf@diaghead

```

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

```

\CT@cellc

2203 \AfterPackage*{colortbl}{%
2204 \def\CT@cellc[#1][#2]#3{%
2205     #1 is either \@firstofone or \@gobble, [#2]#3 = color spec
2206     \xdef\CT@cell@color{\noexpand\CT@color#1[{{#2}}]}{#3}%
2207     \global\let\noexpand\CT@cell@color\relax}%
2208 }
```

2208

## 21.13 † Interface ☞ enumitem : \setlist

```

2209 {*enumitem}
2210 \ProvidesPackage{interfaces-enumitem}
2211 [2010/10/27 v1.2 more control on lists (FC)]
2212 \RequirePackage{interfaces-base}
2213 \itf@Check@Package{enumitem}
2214 \itf@pgfkeys

```

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

```

2215 \providerobustcmd*\interitem{%
2216     \item[]\hskip-\leftmargin
2217 }% \interitem

```

```
\setlist (extension, star form)
```

```

2218 \renewrobustcmd*\setlist{\begingroup \@ifstar
2219   {\let\do\csgappto \itf@setlist}
2220   {\let\do\csgdef \itf@setlist}%
2221 }% \setlist
2222 \newrobustcmd*\itf@setlist[2][0]{%
2223   \let\enit@a@\empty
2224   \let\enit@b@\empty
2225   @for\x:=#1\do{%
2226     \ifx\x@\empty\else
2227       \itf@number\count@\x\relax@nnil
2228         {\edef\enit@b{\enit@b,\the\count@}}
2229         {\edef\enit@a{\enit@a,\x}}%
2230       \fi}%
2231   \ifx\enit@a@\empty\def\enit@a{list}\fi
2232   \ifx\enit@b@\empty\def\enit@b{0}\fi
2233   @for\aa:=\enit@a\do{%
2234     \ifx\aa@\empty\else
2235       @for\b:=\enit@b\do{%
2236         \ifx\b@\empty\else
2237           \ifcsundef{end\aa}
2238             {\expandafter\itf@setlist@error\expandafter{\aa}}
2239             {\do{\enit@{\aa\romannumeral\b}{#2}}}%\aa\romannumeral\b}{#2}}%
2240         \fi}%
2241       \fi}
2242   \endgroup
2243 }% \itf@setlist
2244 \def\itf@setlist@error#1{%
2245   \PackageError{interfaces-enumitem}{Environment '#1' is undefined
2246   \MessageBreak Use \string\newlist\space before \string\setlist
2247   \MessageBreak Your command will be ignored}@\eha
2248 }% \itf@setlist@error
2249 }% \itf@setlist@error

2250 </enumitem>
```

## 21.14 † Interface ↗ truncate

```

2251 {*truncate}
2252 \ProvidesPackage{interfaces-truncate}
```

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

```

2253 \newrobustcmd\trunc{text}[3][\TruncateMarker]{% do nothing if width of #3 is less than #2
2254   \begingroup
2255     \hbadness\maxdimen\hfuzz\vsiz
2256     \setbox\z@=\hbox to\dimexpr#2\relax{\ignorespaces#3\unskip}%
2257     \ifnum\badness>@\M
2258       \def\next{\truncate[#1]{#2}{#3}}%
2259     \else \def\next{#3}%
2260     \fi
2261     \expandafter\endgroup\next
2262 }% \trunc{text}
```

```
\htruncadjust
```

```

2263 \newrobustcmd\htruncadjust[1]{\@testopt
2264   {\itf@htruncadjust[#1]}{}}
2265 }% \htruncadjust
2266 \long\def\itf@htruncadjust#1[#2]{% #1=total width, #2=[x_1,...,x_n], #3=text_1,...,text_n
2267   \begingroup
2268     \let\itf@trunc@width\empty
```

```

2269 \let\itf@istruncated@\gobble % to remove the first comma
2270 \let\the@dimen\relax
2271 \count@\z@ \dimen@\z@
2272 \ifblank{#2}{% no optional [x_1,...,x_n] parameters
2273   {%
2274     \@for\x:=#3\do{%
2275       \advance \count@ \ne
2276       \setbox\z@=\hbox{\ignorespaces\x\unskip}%
2277       \dimen\count@ = \wd\z@
2278       \toks\count@\expandafter{\the\dimen\count@}%
2279       \advance\dimen@ \dimen\count@}%
2280   }
2281   {%
2282     \@for\x:=#2\do{%
2283       \advance \count@ \ne
2284       \toks\count@\expandafter{\x\p@}%
2285       \advance\dimen@ \x\p@
2286       \dimen\count@ -\ne\p@}%
2287   }%
2288 \advance\count@\ne \dimendef\itf@freespace=\count@
2289 \advance\count@\ne \dimendef\itf@tempdim=\count@
2290 \advance\count@\ne \dimendef\itf@sum=\count@
2291 \count@\z@ %%dimen0=\the\dimen@\quad\number\dimen@\par
2292 \itf@freespace\z@ % free space left when w_i < x_i/sum(x_i) * total
2293 \@for\x:=#3\do{%
2294   \advance \count@ \ne
2295   \ifdim \dimen\count@<\z@ % not measured before
2296     \setbox\z@=\hbox{\ignorespaces\x\unskip}%
2297     \dimen\count@ = \wd\z@
2298   \fi
2299   \itf@tempdim = \itf@scaledimen{#1}*\the\toks\count@\dimen@\relax
2300   \ifdim \dimen\count@ > \itf@tempdim
2301     \dimen\count@ = \itf@tempdim
2302     \edef\itf@istruncated{\itf@istruncated,\the\count@}%
2303     \advance\itf@sum by\the\toks\count@
2304   \else
2305     \advance\itf@freespace \itf@tempdim
2306     \advance\itf@freespace -\dimen\count@
2307   \fi
2308   \edef\itf@trunc@width{\itf@trunc@width\the\count@\{\the\dimen\the\count@\}}%
2309 }%
2310 %% Free space=\the\itf@freespace\par
2311 \ifdim\itf@freespace > \z@ % reinsert free space proportionally
2312   \ifx\itf@istruncated\gobble\else
2313     \@for\x:=\itf@istruncated\do{%
2314       \advance\dimen\x by\itf@scaledimen\itf@freespace*\the\toks\x\itf@sum\relax
2315     }%
2316   \fi\fi
2317 \def\the@dimen{\the\dimen}%
2318 \edef\itf@trunc@width{\itf@trunc@width}%
2319 \expandafter\endgroup
2320   \expandafter\def
2321     \expandafter\itf@trunc@width
2322       \expandafter{\itf@trunc@width}%
2323 %% total=\the\dimexpr#1\relax
2324 }% \itf@htruncadjust
2325 \def\itf@scalefactor#1#2{\strip@pt\dimexpr#1*65536/#2\relax}
2326 \def\itf@scaledimen#1#2/#3\relax{%
2327   \dimexpr\itf@scalefactor{#2}{#3}\dimexpr#1\relax\relax
2328 }% \itf@scaledimen

```

\htruncwidth

```

2329 \newcommand*\htruncwidth[1]{% purely expandable
2330   \expandafter\itf@htruncwidth\itf@trunc@width\relax\relax@nnil{#1}%
2331 }% \htruncwidth
2332 \def\itf@htruncwidth#1#2#3@nnil#4{%
2333   \ifx\relax#1\itf@trunc@nosuchwidth{#4}%
2334   \else\ifnum#1=#4
2335     #2%
2336   \else
2337     \itf@htruncwidth#3@nnil{#4}%
2338   \fi\fi
2339 }% \itf@htruncwidth
2340 \def\itf@trunc@nosuchwidth#1{%
2341   \PackageError{interfaces-truncate}{%
2342     You requested the width for text nr#1
2343     \MessageBreak but there is no such text
2344     \MessageBreak Therefore no such width!}\@ehd
2345 }% \itf@trunc@nosuchwidth

2346 </truncate>

```

## 21.15 † interface fancyvrb

```

2347 <*fancyvrb>
2348 \ProvidesPackage{interfaces-fancyvrb}
2349   [2010/10/27 v1.2 Custom verbatim environments (FC)]
2350 \RequirePackage{interfaces-base}
2351 \itf@Check@Package{fancyvrb}
2352 \itf@pgfkeys
2353 \def\itf@tp#1{%
2354 \DefineVerbatimEnvironment{\itf:VerbLines}{Verbatim}{%
2355   gobble=1,frame=lines,framesep=6pt,fontseries=m}
2356 \DefineVerbatimEnvironment{\itf:Verb}{Verbatim}{%
2357   gobble=1,fontseries=m}
2358 \DefineVerbatimEnvironment{\itf:Verb*}{Verbatim}{%
2359   gobble=1,fontseries=m,commandchars=$()}

2360 }
2361 \ltx@ifpackageloaded{doc}{%
2362   {\itf@tp{1}}
2363   {\itf@tp{0}}}
2364 \AtBeginDocument{%
2365 \ifdef{\VerbLines}{%
2366 \ifdef{\endVerbLines}{%
2367   \letcs{\VerbLines}{\itf:VerbLines}%
2368   \letcs{\endVerbLines}{\enditf:VerbLines}%
2369 \fi\fi
2370 \ifdef{\Verb}{%
2371   \letcs{\Verb}{\itf:Verb}%
2372   \letcs{\endVerb}{\enditf:Verb}%
2373 \fi\fi
2374 \ifcsname Verb*\endcsname{%
2375 \ifcsname endVerb*\endcsname{%
2376   \csletcs{\Verb*}{\itf:Verb*}%
2377   \csletcs{\endVerb*}{\enditf:Verb*}%
2378 \fi\fi
2379 }
2380 </fancyvrb>

```

## 21.16 † Interface ☞ appendix and the appendices environment

```

2381 <*appendix>
2382 \ProvidesPackage{interfaces-appendix}
2383   [2010/10/27 v1.2 easy control of appendices environment (FC)]
2384 \RequirePackage{interfaces-base}

```

```

2385 \itf@Check@Package{appendix}
2386 \itf@pgfkeys

\appendices

2387 \let\itf@appendices@pp\appendices
2388 \let\itf@endappendices@pp\endappendices
2389 \let\itf@appendices@pp\appendices
2390 \let\itf@endappendices@pp\endappendices
2391 \renewenvironment{appendices}[1][]{%
2392   \pgfqkeys{/itf/appendix/appendices}{#1}%
2393   \itf@appendices@pp
2394 }
2395 {\itf@endappendices@pp}%

2396 \pgfqkeys{/itf/appendix/appendices}{%
2397   title/.store as code in cs=appendixpagename,%
2398   title/.append code=\ifx\appendixpagename\empty\@dopage@ppfalse
2399   \else\@dopage@pptrue\fi,%
2400   title/.value required,%
2401   intoc/.is if=@dotoc@pp,%
2402   intoc/title/.store as code in cs=appendixtocname,%
2403   intoc/title/.append code=\ifx\appendixtocname\empty\@dotoc@ppfalse
2404   \else\@dotoc@pptrue\fi,%
2405   intoc/title/.default=\pgfkeysvalueof{/itf/appendix/appendices/title},%
2406   section/.is enum={section,chapter},%
2407   section/.append code=%
2408   \ifcase\pgfkeysvalueof{/itf/appendix/appendices/section/.number}\relax
2409   \@chapter@ppfalse\or \@chapter@pptrue\fi
2410   \if@chapter@pp\ifundefined{chapter}{%
2411     \PackageError{interfaces-appendix}{%
2412       You asked chapter-style appendices but your
2413       \MessageBreak current class does not provide chapters!}\@eha
2414     \pgfkeysalso{/itf/appendix/appendices/section}{}{}}%
2415   \fi},%
2416   chapter/.style={section=chapter},%
2417   chapter/.value forbidden,%
2418 }% keys for appendices environment

2419 </appendix>

```

## 21.17 † Interface ↗ embedfile : \embedthisfile

```

2420 {*embedfile}
2421 \ProvidesPackage{interfaces-embedfile}
2422   [2010/10/27 v1.2 easy macros to embed files into pdf (FC)]
2423 \RequirePackage{interfaces-base}
2424 \itf@Check@Package{embedfile}
2425 \newrobustcmd*\embedthisfile{\ifstar
2426   \embedthisfile@main
2427   \embedthisfile@all
2428 }% \embedthisfile
2429 \onlypreamble\embedthisfile
2430 \def\embedthisfile@main{\embedfile{\jobname.tex}}
2431 \def\embedthisfile@all{%
2432   \embedthisfile@main
2433   \AfterPreamble{%
2434     \ltx@ifundefined{@listfiles}{%
2435       \let@listfiles\empty% to avoid: \global\let@filelist\relax in \document
2436       {}%
2437     }%
2438     \gpreto@enddocumenthook{%
2439       \let\do\itf@EmbedOne

```

```

2440      \expandafter\docslist\expandafter{@filelist}%
2441  }%
2442 }% \embedthisfile@all
2443 \def\itf@EmbedOne#1{\itf@Embed@ne#1.tex\@nil}
2444 \def\itf@Embed@ne#1.tex#2\@nil{%
2445   \ltx@ifundefined{cp@#1}%
2446   {}%
2447   {\lifblank{#2}%
2448   {}%
2449   {\embedfile{#1.tex}}}}%
2450 }% \itf@Embed@ne
2451 </embedfile>

```

## 21.18 † Interface ☞ environ

```

2452 <*environ>
2453 \ProvidesPackage{interfaces-environ}
2454   [2010/07/23 v1.0 ReNewEnviron = RenewEnviron (FC)]
2455 \RequirePackage{interfaces-base}
2456 \AfterPackage{environ}{%
2457   \ifundef\ReNewEnviron
2458   {\let\ReNewEnviron\RenewEnviron}
2459   {}%
2460 }%
2461 </environ>

```

## 21.19 † Interface umrand

```

2462 <*umrand>
2463 \ProvidesPackage{interfaces-umrand}
2464   [2010/07/23 v1.0 patch for umrand.sty ifnotempty (FC)]
2465 \RequirePackage{interfaces-base}
2466 \BeforePackage{umrand}{%
2467   \let\itf@newif\newif
2468   \def\itf@tp{\ifnotempty}%
2469   \def\newif#1{\def\x{#1}\ifx\x\itf@tp \expandafter@gobble
2470   \else\expandafter\ltx@newif\fi{#1}}%
2471 }
2472 \AfterPackage{umrand}{%
2473   \let\newif\itf@newif
2474   \let\itf@newif@\undefined
2475 }
2476 </umrand>

```

## 21.20 † Interface ☞ scrlfile

```

2477 <*scrlfile>
2478 \ProvidesPackage{interfaces-scrlfile}
2479   [2010/10/27 v1.2 scrlfile hooks improved (FC)]
2480 \RequirePackage{interfaces-base}

```

\@pkgextension In L<sup>A</sup>T<sub>E</sub>X they are defined \onlypreamble. So we use the ltxcmds equivalents:

\@clsextension

```

2481 \AfterEndPreamble{%
2482   \let@\pkgextension\ltx@pkgextension
2483   \let@\clsextension\ltx@clsextension
2484   \let@ifpackageloaded\ltx@ifpackageloaded
2485   \let@ifclassloaded\ltx@ifclassloaded
2486 }

```

\scr@load@hook

```

2487 \renewrobustcmd*\scr@load@hook[2]{%
2488   \csuse{#2-@#1}%

```

```
2489 \ifcsname#2-@#1-\string\keep\endcsname\else\global\csundef{#2-@#1}\fi
2490 }% \scr@load@hook
```

\scr@set@load@hook (optimisation)

```
2491 \renewrobustcmd*\scr@set@load@hook[2]{\csgappto{#2-@#1}}
```

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

```
\AfterFile
2492 \renewrobustcmd*\BeforeFile{@ifstar
2493   {\itf@s@LoadFile{before}}
2494   {\itf@LoadFile{before}}%
2495 }% \BeforeFile
2496 \renewrobustcmd*\AfterFile{@ifstar
2497   {\itf@s@LoadFile{after}}
2498   {\itf@LoadFile{after}}%
2499 }% \AfterFile
```

```
2500 \def\itf@s@LoadFile#1#2{%
2501   \global\cslet{#2-@#1-\string\keep}@\empty
2502   \scr@set@load@hook{#1}{#2}%
2503 }% \itf@s@LoadFile
2504 \def\itf@LoadFile#1#2{%
2505   \global\csundef{#2-@#1-\string\keep}%
2506   \scr@set@load@hook{#1}{#2}%
2507 }% \itf@LoadFile
```

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

```
2508 \renewcommand*\@input[1]{\IfFileExists{#1}
2509   {\scr@load@hook{before}{#1}%
2510    @@\input\@filef@und
2511    \scr@load@hook{after}{#1}}%
2512   {\typeout{No file #1.}}%
2513 }% \@input
```

```
2514 </scrfile>
```

## 21.21 † Interface pgfkeys ([pgfmanual](#))

### 21.21.1 interfaces-pgfkeys

```
2515 <*pgfkeys>
2516 \ProvidesPackage{interfaces-pgfkeys}
2517   [2010/10/27 v1.2 pgfkeys handlers definitions (FC)]
2518 \RequirePackage{interfaces-base}
2519 \AtEndOfPackage{%
2520   \let\itf@pgfkeys@unpack\pgfkeys@unpack
2521   \let\itf@pgfkeys@case@three\pgfkeys@case@three
2522   \let\itf@pgfkeys@exp@call\pgfkeys@exp@call
2523   \let\itf@pgfkeysaddvalue\pgfkeysaddvalue
2524 }% AtEndOfPackage
2525 \def\itf@pgfkeys{%
2526   \global\let\pgfkeys@unpack\itf@pgfkeys@unpack
2527   \global\let\pgfkeys@case@three\itf@pgfkeys@case@three
2528   \global\let\pgfkeys@exp@call\itf@pgfkeys@exp@call
2529   \global\let\pgfkeysaddvalue\itf@pgfkeysaddvalue
2530 }% \itf@pgfkeys
2531 \AtBeginDocument{%
2532   \itf@pgfkeys
```

```
2533   \let\itf@pgfkeys\relax
2534 }% AtBeginDocument
```

## Some new pgfkeys macros

\pgfkeysmeaning

```
2535 \providedeclaration*\pgfkeysmeaning[1]{%
2536   \pgfkeysifdefined{#1}%
2537   {\expandafter\meaning\csname pgfk@#1\endcsname}%
2538   {\meaning\@undefined}%
2539 }%
```

\pgfkeysvalueof@unexpanded

```
2540 \def\pgfkeysvalueof@unexpanded#1{%
2541   \unexpanded\expandafter\expandafter\expandafter{%
2542     \csname pgfk@#1\endcsname}%
2543 }% \pgfkeysvalueof@unexpanded
```

## Some pgfkeys macros redefinition/corrections

\pgfkeysEsetvalue

```
\pgfkeysaddvalue (bug correction)
\pgfkeysEaddvalue
2544 \newrobustcmd*\pgfkeysEsetvalue[2]{%
2545   \expandafter\edef\csname pgfk@#1\endcsname{#2}%
2546 }% \pgfkeysEsetvalue
2547 \renewrobustcmd*\pgfkeysaddvalue[3]{% \cspreappto{pgfk@#1}{#2}{#3} (bug in pgfkeys.code.tex)
2548   \edef\pgfkeys@global@temp{%
2549     \unexpanded{#2}%
2550     \ifcsname pgfk@#1\endcsname
2551       \unexpanded\expandafter\expandafter\expandafter{%
2552         \csname pgfk@#1\endcsname}%
2553     \fi
2554     \unexpanded{#3}%
2555   \pgfkeyslet{#1}\pgfkeys@global@temp
2556 }% \pgfkeysaddvalue
2557 \newrobustcmd*\pgfkeysEaddvalue[3]{% \csepappto{pgfk@#1}{#2}{#3}
2558   \edef\pgfkeys@global@temp{%
2559     #2%
2560     \ifcsname pgfk@#1\endcsname
2561       \unexpanded\expandafter\expandafter\expandafter{%
2562         \csname pgfk@#1\endcsname}%
2563     \fi
2564     #3}%
2565   \pgfkeyslet{#1}\pgfkeys@global@temp
2566 }% \pgfkeysaddvalue
```

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

```
2567 %% missed a pair of braces around the argument in pgfkeys.code.tex ( code could break if it c
2568 \def\pgfkeys@exp@call#1{\pgfkeysalso{\pgfkeyscurrentpath={#1}}}
```

## Handlers /blank

```

2569 %% The value is now checked by \itf@pgfkeys@default@blank@required:
2570 \def\pgfkeys@unpack#1=#2=#3\pgfkeys@stop{%
2571   \pgfkeys@spdef\pgfkeyscurrentkey{\#1}%
2572   \edef\pgfkeyscurrentkey{\pgfkeyscurrentkey}%
2573   \ifx\pgfkeyscurrentkey\pgfkeys@empty
2574     % Skip
2575   \else
2576     \pgfkeys@add@path@as@needed
2577     \pgfkeys@spdef\pgfkeyscurrentvalue{\#2}%
2578     \itf@pgfkeys@default@blank@required % <FC>
2579             \pgfkeyscurrentkey % <FC>
2580             \pgfkeys@case@one % <FC>
2581   \fi
2582 }% \pgfkeys@unpack

```

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

```

2583 \def\itf@pgfkeys@default@blank@required#1#2{%
2584   \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text% Hmm... no value
2585     \pgfkeysifdefined{\#1/.@def}{%
2586       {\pgfkeysgetvalue{\#1/.@def}\pgfkeyscurrentvalue}%
2587       {}% no default, so leave it
2588     \fi
2589     \ifx\pgfkeyscurrentvalue\empty
2590       \pgfkeysifdefined{\#1/.@blank}{%
2591         {\pgfkeysgetvalue{\#1/.@blank}\pgfkeyscurrentvalue}%
2592         {}% no /.blank specified
2593       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text% in case /.blank -> \pgfkeysnovalue
2594         \pgfkeysifdefined{\#1/.@def}{%
2595           {\pgfkeysgetvalue{\#1/.@def}\pgfkeyscurrentvalue}%
2596           {}% no default, so leave it
2597         \fi\fi
2598         \ifx\pgfkeyscurrentvalue\pgfkeysvaluerequired
2599           \pgfkeysvalueof{/errors/value required/.@cmd}\pgfkeyscurrentkey
2600                         \pgfkeyscurrentvalue
2601                         \pgfeov
2602       \else
2603         #2%
2604       \fi
2605 }% \itf@pgfkeys@default@blank@required

```

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

```

2606 %% To handle the cases of: /handler/.default= /handler/.blank= /handler/.value required,
2607 %% it is necessary to redefine \pgfkeys@case@three
2608 \renewcommand*\pgfkeys@case@three{%
2609   \pgfkeys@split@path
2610   \pgfkeysifdefined{/handlers/\pgfkeyscurrentname/.@cmd}{%
2611     {%
2612       \itf@pgfkeys@default@blank@required{/handlers/\pgfkeyscurrentname}%
2613       {\pgfkeysgetvalue{/handlers/\pgfkeyscurrentname/.@cmd}\pgfkeys@code
2614         \expandafter\pgfkeys@code\pgfkeyscurrentvalue\pgfeov}%
2615     }
2616     \pgfkeys@unknown % no handler, thus the key is unknown
2617 }% \pgfkeys@case@three

```

**Handlers definition: `/alias`, `/blank`**

```

2618 \pgfkeys{%
2619   /handlers/.alias/.code=\itf@pgfkeys@ifaliasexists{#1}{%
2620     \pgfkeysedef\pgfkeyscurrentpath{\noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}},%
2621     /handlers/.alias/.value required,%}
2622   /handlers/.blank/.code=\pgfkeyssetvalue{\pgfkeyscurrentpath/.@blank}{#1},%
2623   /handlers/.blank/.default=\pgfkeysnovalue,%}
2624 }% /.alias and /.blank
2625 \def\itf@pgfkeys@ifaliasexists#1#2{%
2626   \pgfkeysifdefined{\pgfkeysdefaultpath#1/.@cmd}{#2}%
2627   {\pgfkeysifdefined{\pgfkeysdefaultpath#1}%
2628    {#2}%
2629    {\itf@pgfkeys@aliaserror{#1}}}%
2630 }% \itf@pgfkeys@ifaliasexists
2631 \def\itf@pgfkeys@aliaserror#1{%
2632   \PackageError{interfaces-pgfkeys}{%
2633     You want the key \pgfkeyscurrentpath\space%
2634     \MessageBreak to be an alias for \pgfkeysdefaultpath#1%
2635     \MessageBreak but the key \pgfkeysdefaultpath#1%
2636     \MessageBreak is not defined!}\@eha\@gobble
2637 }% \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).

```

2638 \pgfkeys{%
2639   /handlers/.alias*/.code=\itf@pgfkeys@ifaliasexists{#1}{%
2640     \pgfkeysedef\pgfkeyscurrentpath{\noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}}%}
2641     @tfor\x:=+-*\do{%
2642       \pgfkeysifdefined{\pgfkeysdefaultpath#1\x/.@cmd}%
2643       {\pgfkeysedef{\pgfkeyscurrentpath\x}{%
2644         \noexpand\pgfkeysalso{\pgfkeysdefaultpath#1\x={##1}}}}%
2645     }%
2646   },%
2647   /handlers/.alias*/.value required,%}
2648 }% /.alias*

```

**Handlers definition: `/is private`**

```

2649 \pgfkeys{%
2650   /handlers/.is private/.code=\pgfkeysdef\pgfkeyscurrentpath{%
2651     \PackageError{#1}{Key \pgfkeyscurrentkey\space is private%
2652     \MessageBreak and should not be modified}\@ehd},%
2653   /handlers/.is private to class/.code=\pgfkeysdef\pgfkeyscurrentpath{%
2654     \ClassError{#1}{Key \pgfkeyscurrentkey\space is private%
2655     \MessageBreak and should not be modified}\@ehd}%
2656 }% /.is private

```

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

```

2657 \def\itf@pgfkeys@choice[#1]#2#3{%
2658   \pgfkeys{\pgfkeyscurrentpath/.cd,%
2659     .is choice,%
2660     #2/.code={#3},%
2661     .default/.expanded=\@firstoftwo#1}%
2662 }% \itf@pgfkeys@choice
2663 \pgfkeys{%
2664   /handlers/.true/.code=\itf@pgfkeys@choice[{true}{false}]{true}{#1},%
2665   /handlers/.false/.code=\itf@pgfkeys@choice[{true}{false}]{false}{#1},%
2666   /handlers/.on/.code=\itf@pgfkeys@choice[{on}{off}]{on}{#1},%
2667   /handlers/.off/.code=\itf@pgfkeys@choice[{on}{off}]{off}{#1},%
2668 }% /.true /.false

```

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

```

2669 \pgfkeys{%
2670   /handlers/.store in cs/.code=\pgfkeysdef\pgfkeyscurrentpath{\csdef{#1}{##1}},%
2671   /handlers/.estore in cs/.code=\pgfkeysdef\pgfkeyscurrentpath{\csedef{#1}{##1}},%
2672 }% /.store in cs

```

**Handler definition: ./store as code**

```

2673% /.store as code handlers
2674 \pgfkeys{%
2675   /handlers/.store as code/.code={%
2676     \pgfkeysedef\pgfkeyscurrentpath{%
2677       \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
2678     \pgfkeyslet\pgfkeyscurrentpath\empty
2679     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\empty
2680     @tfor\x:=#1\do{%
2681       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2682         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
2683           \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
2684           \noexpand\unexpanded{##1}}}}%
2685       \pgfkeyslet{\pgfkeyscurrentpath\x/.@def}\empty
2686     },%
2687   /handlers/.store as code/.default=*,%
2688 }% /.store as code

```

**Handlers definition: ./estore as code**

```

2689 \pgfkeys{%
2690   /handlers/.estore as code/.code={%
2691     \pgfkeysedef\pgfkeyscurrentpath{%
2692       \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{##1}}%
2693     \pgfkeyslet\pgfkeyscurrentpath\empty
2694     @tfor\x:=#1\do{%
2695       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2696         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
2697           \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
2698           \noexpand\unexpanded{##1}}}}%
2699     },%
2700   /handlers/.estore as code/.default=*,%
2701 }% /.estore as code

```

**Handlers definition: `./store as code in cs=<cs-name>`**

```

2702 \pgfkeys{%
2703   /handlers/.store as code in cs/.code={%
2704     \pgfkeysdef\pgfkeyscurrentpath{%
2705       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2706       \else\csdef{\#1}{##1}\fi}%
2707     \@tfor\x:=*+\do{%
2708       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2709         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
2710           \noexpand\ifcsname#1\endcsname
2711             \noexpand\expandafter\noexpand\expandonce
2712             \noexpand\csname#1\endcsname
2713           \noexpand\fi
2714             \noexpand\unexpanded{##1}}}}%
2715       \pgfkeyslet{\pgfkeyscurrentpath\x/.@def}\@empty
2716     }},%
2717   /handlers/.store as code in cs/.value required,%
2718 }% /.store as code in cs
2719 \pgfkeys{%
2720   /handlers/.estore as code in cs/.code={%
2721     \pgfkeysdef\pgfkeyscurrentpath{%
2722       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2723       \else\csedef{\#1}{##1}\fi}%
2724     \@tfor\x:=*+\do{%
2725       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2726         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
2727           \noexpand\ifcsname#1\endcsname
2728             \noexpand\expandafter\noexpand\expandonce
2729             \noexpand\csname#1\endcsname
2730           \noexpand\fi
2731             \noexpand\unexpanded{##1}}}}%
2732     },% default if \pgfkeysnovalue which expands to \@empty
2733   /handlers/.estore as code in cs/.value required,%
2734 }% /.estore as code in cs

```

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

```

2735 \protected\def\itf@pgfk@color{\#1\#2\@nil}%
2736   \ifx\@let@token\color
2737     \def\pgfkeyscurrentvalue{\#2}%
2738   \else\ifx\@let@token\normalcolor
2739     \def\pgfkeyscurrentvalue{\default}%
2740   \else\ifx\@let@token\pgfkeysnovalue
2741     \def\pgfkeyscurrentvalue{\default}%
2742   \fi\fi\fi
2743 }% \itf@pgfk@color
2744 \pgfkeys{%
2745   /handlers/.store as color/.code=%
2746     \pgfkeysedef\pgfkeyscurrentpath{%
2747       \futurelet\noexpand\@let@token\itf@pgfk@color
2748       ##1\noexpand\pgfkeysnovalue\noexpand\@nil
2749       \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
2750 }% /.store as color

```

**Handlers definition: `./store as color in cs=<cs-name>`**

```

2751 \pgfkeys{%
2752   /handlers/.store as color in cs/.code=%
2753   \pgfkeysedef\pgfkeyscurrentpath{%
2754     \futurelet\noexpand\let@token\itf@pgfk@color
2755     ##1\noexpand\pgfkeysnovalue\noexpand\@nil
2756     \cslet{\#1}\pgfkeyscurrentvalue}%
2757 }% /.store as color in cs

```

**Handlers definition: `./store as color command`**

```

2758 \protected\def\itf@pgfk@colorcmd#1\@nil{%
2759   \ifcat @\noexpand\let@token
2760     \def\pgfkeyscurrentvalue{\color{\#1}}%
2761   \else\ifcat 0\@expand\let@token
2762     \def\pgfkeyscurrentvalue{\color{\#1}}%
2763   \else\ifx\@let@token\@nil
2764     \let\pgfkeyscurrentvalue\normalcolor
2765   \fi\fi\fi
2766 }% \itf@pgfk@colorcmd
2767 \pgfkeys{%
2768   /handlers/.store as color command/.code=%
2769   \pgfkeysedef\pgfkeyscurrentpath{%
2770     \futurelet\noexpand\let@token\itf@pgfk@colorcmd
2771     ##1\noexpand\@nil
2772     \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
2773     \pgfkeyssetvalue{\pgfkeyscurrentpath/.@def}\normalcolor
2774 }% /.store as color command

```

**Handlers definition: `./store as color command in cs=<cs-name>`**

```

2775 \pgfkeys{%
2776   /handlers/.store as color command in cs/.code=%
2777   \pgfkeysedef\pgfkeyscurrentpath{%
2778     \futurelet\noexpand\let@token\itf@pgfk@colorcmd
2779     ##1\noexpand\@nil
2780     \cslet{\#1}\noexpand\pgfkeyscurrentvalue}%
2781     \pgfkeyssetvalue{\pgfkeyscurrentpath/.@def}\normalcolor
2782 }% /.store as color command in cs

```

**Handlers definition `./store as number`**

```

2783 \pgfkeys{%
2784   /handlers/.store as number/.code=%
2785   \pgfkeysedef\pgfkeyscurrentpath{%
2786     \normalcolor\pgfkeyssetvalue{\pgfkeyscurrentpath}{\numexpr##1\relax}%
2787     \pgfkeyslet\pgfkeyscurrentpath\z@
2788     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
2789     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
2790     \@tfor\x:=#1\do{%
2791       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2792         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
2793           \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
2794           \x
2795           \numexpr\unexpanded{\#1}\relax}%
2796         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@\fi}%
2797         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@\fi}%
2798       }%,%
2799     /handlers/.store as number/.default=+-*,%
2800   }% /.store as number

```

## Handlers definition: `./estore as number`

```

2801 \pgfkeys{%
2802   /handlers/.estore as number/.code={%
2803     \pgfkeysedef\pgfkeyscurrentpath{%
2804       \noexpand\pgfkeysetvalue{\pgfkeyscurrentpath}{\number\numexpr##1\relax}%
2805     \pgfkeyslet\pgfkeyscurrentpath\z@%
2806     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@%
2807     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@%
2808     @tfor\x:=#1\do{%
2809       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2810         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2811           \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}%
2812           \x
2813           \numexpr##1\relax}%
2814       }%,%
2815     /handlers/.estore as number/.default=+-*,%
2816   }% /.estore as number

```

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

```

2817 \pgfkeys{%
2818   /handlers/.store as number in cs/.code={%
2819     \pgfkeysedef\pgfkeyscurrentpath{%
2820       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2821       \else\csdef{#1}{\numexpr##1\relax}\fi}%
2822     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@%
2823     @tfor\x:=+-*\do{%
2824       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2825         \letcs\noexpand\pgfkeys@temp{%
2826           \noexpand\ifcsname#1\endcsname
2827             #1\noexpand\else z@\noexpand\fi}%
2828         \edef\noexpand\pgfkeys@temp{%
2829           \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2830             \noexpand\expandonce\noexpand\pgfkeys@temp
2831             \x
2832             \numexpr\noexpand\unexpanded{##1}\relax}%
2833           }\noexpand\pgfkeys@temp}%
2834         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@\fi}%
2835         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@\fi}%
2836     }%,%
2837   /handlers/.store as number in cs/.value required,%
2838 }% /.store as number in cs

```

## Handlers definition: `./estore as number in cs`

```

2839 \pgfkeys{%
2840   /handlers/.estore as number in cs/.code={%
2841     \pgfkeysedef\pgfkeyscurrentpath{%
2842       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2843       \else\csedef{#1}{\number\numexpr##1\relax}\fi}%
2844     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@%
2845     @tfor\x:=+-*\do{%
2846       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2847         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2848           \noexpand\csname\noexpand\ifcsname#1\endcsname
2849             #1\noexpand\else z@\noexpand\fi\endcsname
2850             \x
2851             \numexpr##1\relax}%
2852         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@\fi}%
2853         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@\fi}%
2854     }%,%

```

```
2855   /handlers/.estore as number in cs/.value required,%
2856 }% /.estore as number in cs
```

### Handlers definition: `/defcounter`

```
2857 \pgfkeys{%
2858   /handlers/.defcounter/.code=\pgfkeysdef\pgfkeyscurrentpath{\defcounter{\#1}{##1}}%
2859 }% /.defcounter
```

### Handlers definition: `/store as glue`

```
2860 \pgfkeys{%
2861   /handlers/.store as glue/.code={%
2862     \pgfkeysedef\pgfkeyscurrentpath{%
2863       \noexpand\pgfkeysetvalue{\pgfkeyscurrentpath}{\glueexpr##1\relax}}%
2864     \pgfkeyslet\pgfkeyscurrentpath\z@skip
2865     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@skip
2866     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
2867     \@tfor\x:=#1\do{%
2868       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2869         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
2870           \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
2871           \x
2872           \if*\x\noexpand\itf@units@scale##1!
2873             \else\glueexpr\noexpand\unexpanded{##1}\relax\fi}}%
2874         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@skip\fi}%
2875         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@skip\fi}%
2876       }},%
2877     /handlers/.store as glue/.default=+-*,%
2878 }% /.store as glue
```

### Handlers definition: `/estore as glue`

```
2879 \pgfkeys{%
2880   /handlers/.estore as glue/.code={%
2881     \pgfkeysedef{\pgfkeyscurrentpath}{%
2882       \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{%
2883         \noexpand\the\glueexpr##1\relax}}%
2884     \pgfkeyslet\pgfkeyscurrentpath\z@skip
2885     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@skip
2886     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
2887     \@tfor\x:=#1\do{%
2888       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2889         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2890           \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\x
2891           \if*\x \noexpand\itf@units@scale##1!
2892             \else\glueexpr##1\relax\fi}}%
2893         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@skip\fi}%
2894         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@skip\fi}%
2895       }},%
2896     /handlers/.estore as glue/.default=+-*,%
2897 }% /.estore as glue
```

**Handlers definition: `./store as dimen`**

```

2898 \pgfkeys{%
2899   /handlers/.store as dimen/.code={%
2900     \pgfkeysedef\pgfkeyscurrentpath{%
2901       \noexpand\pgfkeysetvalue{\pgfkeyscurrentpath}{\dimexpr##1\relax}}%
2902     \pgfkeyslet\pgfkeyscurrentpath\z@%
2903     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@%
2904     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@%
2905     @tfor\x:=#1\do{%
2906       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2907         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
2908           \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
2909           \x%
2910           \if*\x \noexpand\itf@units@scale##1!
2911           \else\dimexpr\noexpand\unexpanded{##1}\relax\fi}}}%
2912     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
2913     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
2914   },%
2915   /handlers/.store as dimen/.default=+-*,%
2916 }% /.store as dimen

```

**Handlers definition: `./estore as dimen`**

```

2917 \pgfkeys{%
2918   /handlers/.estore as dimen/.code={%
2919     \pgfkeysedef\pgfkeyscurrentpath{%
2920       \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{%
2921         \noexpand\the\dimexpr##1\relax}}%
2922     \pgfkeyslet\pgfkeyscurrentpath\z@%
2923     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@%
2924     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@%
2925     @tfor\x:=#1\do{%
2926       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2927         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2928           \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\x%
2929           \if*\x \noexpand\itf@units@scale##1!
2930           \else \dimexpr##1\relax\fi}}%
2931     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
2932     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
2933   },%
2934   /handlers/.estore as dimen/.default=+-*,%
2935 }% /.estore as dimen

```

**Handlers definition: `./store as glue in cs=<cs-name>`**

```

2936 \pgfkeys{%
2937   /handlers/.store as glue in cs/.code={%
2938     \pgfkeysedef\pgfkeyscurrentpath{%
2939       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text%
2940       \else\csdef{#1}{\glueexpr##1\relax}\fi}%
2941     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip%
2942     @tfor\x:=+-*\do{%
2943       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2944         \letcs\noexpand\pgfkeys@temp{%
2945           \noexpand\ifcsname#1\endcsname%
2946           #1\noexpand\else z@skip\noexpand\fi}%
2947           \edef\noexpand\pgfkeys@temp{%
2948             \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2949               \noexpand\expandonce\noexpand\pgfkeys@temp%
2950               \x%
2951               \if*\x\noexpand\itf@units@scale##1!
```

```

2952          \else\glueexpr\noexpand\unexpanded{##1}\relax\fi}}%
2953          }\noexpand\pgfkeys@temp}%
2954          \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
2955          \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
2956      },%
2957      /handlers/.store as glue in cs/.value required,%
2958 }% /.store as glue in cs

```

### Handlers definition: `./estore as glue in cs=<cs-name>`

```

2959 \pgfkeys{%
2960     /handlers/.estore as glue in cs/.code={%
2961         \pgfkeysdef\pgfkeyscurrentpath{%
2962             \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2963             \else\csedef{#1}{\the\glueexpr##1\relax}\fi}%
2964             \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
2965             \atfor\x:=+-*\do{%
2966                 \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2967                     \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2968                         \noexpand\csname\noexpand\ifcsname#1\endcsname
2969                             #1\noexpand\else z@skip\noexpand\fi\endcsname
2970                             \x
2971                             \if*\x\noexpand\itf@units@scale##1!
2972                             \else\glueexpr##1\relax\fi}}%
2973             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
2974             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
2975         },%
2976         /handlers/.etore as glue in cs/.value required,%
2977     }% /.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` (*i.e.* a length). The skip is given by its name, not as a control sequence.

```

2978 \pgfkeys{%
2979     /handlers/.store in skip cs/.code={%
2980         \pgfkeysdef\pgfkeyscurrentpath{%
2981             \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2982             \else\csname#1\endcsname\glueexpr##1\relax\fi}%
2983             \pgfkeyslet{/pgfkeyscurrentpath/.@blank}\z@skip
2984             \atfor\x:=+-*\do{%
2985                 \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2986                     \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2987                         \csname#1\endcsname % should have been defined as a skip (not expandable)
2988                         \x
2989                         \if*\x\noexpand\itf@units@scale##1!
2990                         \else\glueexpr##1\relax\fi}}%
2991             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
2992             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
2993         },%
2994         /handlers/.store in skip cs/.value required,%
2995     }% /.store in skip cs

```

**Handlers definition: `./store as dimen in cs=<cs-name>`**

```

2996 \pgfkeys{%
2997   /handlers/.store as dimen in cs/.code={%
2998     \pgfkeysdef\pgfkeyscurrentpath{%
2999       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3000       \else\csdef{\#1}{\dimexpr##1\relax}\fi}%
3001     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3002     \@tfor\x:=+-*\do{%
3003       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3004         \letcs\noexpand\pgfkeys@temp{%
3005           \ifcsname#1\endcsname#1\else z@\fi}%
3006           \edef\noexpand\pgfkeys@temp{%
3007             \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3008               \noexpand\expandonce\noexpand\pgfkeys@temp
3009               \x
3010               \if*\x\noexpand\itf@units@scale##1!
3011               \else\dimexpr\noexpand\unexpanded{##1}\fi}%
3012             }\noexpand\pgfkeys@temp}%
3013           \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@\fi}%
3014           \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@\fi}%
3015     }},%
3016   /handlers/.store as dimen in cs/.value required,%
3017 }% /.store as dimen in cs {csname}{+-}

```

**Handlers definition: `./estore as dimen in cs=<cs-name>`**

```

3018 \pgfkeys{%
3019   /handlers/.estore as dimen in cs/.code={%
3020     \pgfkeysdef\pgfkeyscurrentpath{%
3021       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3022       \else @defaultunits\itf@sk ##1pt\relax@nnil
3023       \csedef{\#1}{\the\dimexpr\itf@sk\relax}\fi}%
3024     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3025     \@tfor\x:=+-*\do{%
3026       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3027         \noexpand\ifcsname#1\endcsname
3028           \noexpand@defaultunits\itf@sk
3029           \noexpand\csname#1\endcsname pt\relax\noexpand@nnil
3030           \noexpand\else\itf@sk\z@\noexpand\fi
3031           \if*\x
3032             \itf@sk\dimexpr\itf@sk*\noexpand\itf@units@scale##1!\relax
3033           \else
3034             \noexpand@defaultunits@tempdima ##1pt\relax\noexpand@nnil
3035             \advance\itf@sk\x@tempdima
3036           \fi
3037           \noexpand\pgfkeysalso{\pgfkeyscurrentpath=\itf@sk}}%
3038         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@\fi}%
3039         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@\fi}%
3040     }},%
3041   /handlers/.etore as dimen in cs/.value required,%
3042 }% /.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.

```

3043 \pgfkeys{%
3044   /handlers/.store in dimen cs/.code={%
3045     \pgfkeysdef\pgfkeyscurrentpath{%
3046       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text%
3047       \else\csname#1\endcsname\dimexpr##1\relax\fi}%
3048     \pgfkeyslet{/pgfkeyscurrentpath/.@blank}\z@
3049     @tfor\x:=+-*\do{%
3050       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3051         \if*\x \else%
3052           \noexpand@\defaultunits\itf@sk ##1pt\relax\noexpand@nnil%
3053           \fi%
3054           \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3055             \csname#1\endcsname % should have been defined as a dimen (not expandable)
3056             \x%
3057             \if*\x \noexpand\itf@units@scale##1!
3058             \else\itf@sk\fi}}%
3059     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@\fi}%
3060     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@\fi}%
3061   },%
3062   /handlers/.store in dimen cs/.value required,%
```

3063 }% /.store in dimen cs

**Handlers definition: `./is multiple choice`**

```

3064 \pgfkeys{%
3065   /handlers/.is multiple choice/.code=%
3066     \pgfkeys{\pgfkeyscurrentpath/.cd,%
3067       .unknown/.style={/errors/unknown choice value=\pgfkeyscurrentkey\pgfkeyscurrentvalue%
3068       .code=\expandafter\pgfkeys\expandafter{\pgfkeyscurrentkey/.cd,####1}%
3069     },%
3070   /handlers/.is multiple choice/.value forbidden,%
```

3071 }% /.is multiple choice

**Handlers definition: `./is enum`**

```

3072 \pgfkeys{%
3073   /handlers/.is enum/.code={%
3074     \itf@cnt\z@
3075     @for\x:=\pgfkeyscurrentvalue\do{%
3076       \pgfkeysedef{\pgfkeyscurrentpath/\x}{%
3077         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}\{\x\}%
3078         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath/.number}\{\the\itf@cnt\}%
3079       \ifnum\itf@cnt=\z@
3080         \pgfkeyslet{\pgfkeyscurrentpath/.@def}\x%
3081         \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\x%
3082         \fi%
3083         \advance\itf@cnt\@ne%
3084       }%
3085       \pgfkeysalso{\pgfkeyscurrentpath/.is choice},%
3086   /handlers/.is enum/.value required,%
```

3087 }% /.is enum

3088 </pgfkeys>

## 21.22 † Interface etoolbox - etoolbox-like commands

```

3089 {*etoolbox}
3090 \ProvidesPackage{interfaces-etoolbox}
3091   [2010/10/27 v1.2 some more etoolbox-like commands (FC)]
3092 \RequirePackage{interfaces-base}
3093 \begingroup\catcode`\&=3
3094   \long\gdef\etb@doitem#1&{\do{\#1}}
3095 \endgroup

\preappto
\epreappto
\gpreappto 3096 \providerobustcmd\preappto[3]{%
3097   \edef#1{\unexpanded{\#2}\ifdefined#1\expandonce#1\fi\unexpanded{\#3}}%
\xpreappto 3098 }% \preappto
            \providerobustcmd\epreappto[3]{%
3100   \edef#1{\#2\ifdefined#1\expandonce#1\fi\#3}%
3101 }% \epreappto
            \providerobustcmd*\gpreappto{\global\preappto}
3103 \providerobustcmd*\xpreappto{\global\epreappto}

\cspreappto
\csepreatto
\csgpreappto 3104 \providerobustcmd*\cspreappto[1]{%
3105   \csname itf@\ifcsname#1\endcsname
3106     preappto\else defpreapp\fi\expandafter\endcsname
3107           \csname#1\endcsname
3108 }% \cspreappto
3109 \long\def\itf@preappto#1#2#3{\edef#1{\#2\expandonce{\#1\#3}}}
3110 \long\def\itf@defpreapp#1#2#3{\edef#1{\#2\#3}}
3111 \providerobustcmd*\csepreatto[1]{%
3112   \csname itf@\ifcsname#1\endcsname
3113     preappto\else defpreapp\fi\expandafter\endcsname
3114           \csname#1\endcsname
3115 }% \csepreatto
3116 \providerobustcmd*\csgpreappto{\global\cspreappto}
3117 \providerobustcmd*\csxpreappto{\global\csepreatto}

```

---

```

\removefrom
\eremovefrom
\gremovefrom
\xremovefrom 3118 \providerobustcmd\removefrom[2]{\in@false
3119   \long\def\itf@tp##1\itf@tp{%
3120     \in@true
3121     \edef#1{\unexpanded{\##1}}%
3122   }\expandafter\itf@removefrom\expandafter{\#1}{\#2}%
3123 }% \removefrom
3124 \providerobustcmd\eremovefrom[2]{\begin{group}
3125   \edef\x{\endgroup\removefrom\noexpand#1{\#2}%
3126   }\x
3127 }% \eremovefrom
3128 \providerobustcmd\gremovefrom[2]{%
3129   \removefrom{\#1}{\#2}%
3130   \global\let#1=#1%
3131 }% \gremovefrom
3132 \providerobustcmd\xremovefrom[2]{%
3133   \eremovefrom{\#1}{\#2}%
3134   \global\let#1=#1%
3135 }% \xremovefrom

```

\itf@removefrom

```

3136 \long\def\itf@removefrom#1#2{%
  removes #2 (code) from #1 (hook)
3137   \def\in@@##1##2\in@@{\itf@tp##1##2\in@@}%
3138   \def\in@@##1\in@@##2\in@@{%
3139     \ifx\in@@##2\else##1\fi
3140   }\expandafter\in@@\in@@#1\itf@tp\in@@#2\in@\in@@
3141 }% \itf@removefrom

```

\removefromstart

\eremovefromstart

\gremovefromstart

\xremovefromstart

```

3142 \newrobustcmd*\removefromstart[2]{\in@false
3143   \def\itf@tp##1\itf@tp{%
3144     \in@true
3145     \edef#1{\unexpanded{##1}}%
3146   }\expandafter\itf@removefromstart\expandafter{#1}{#2}%
3147 }% \removefromstart
3148 \providerobustcmd*\eremovefromstart[2]{\begingroup
3149   \edef\x{\endgroup\removefromstart\noexpand#1{#2}%
3150   }\x
3151 }% \eremovefromstart
3152 \providerobustcmd*\gremovefromstart[2]{%
3153   \removefromstart{#1}{#2}%
3154   \global\let#1=#1
3155 }% \gremovefromstart
3156 \providerobustcmd*\xremovefromstart[2]{%
3157   \eremovefromstart{#1}{#2}%
3158   \global\let#1=#1
3159 }% \xremovefromstart

```

\itf@removefromstart

```

3160 \def\itf@removefromstart#1#2{%
  removes #2 (code) from #1 (hook)
3161   \def\in@@##1##2\in@@{\##1\in@\itf@tp##2\in@@}%
3162   \def\in@@##1\in@@##2\in@@{%
3163     \ifx\in@@##2\fi
3164   }\expandafter\in@@\in@@#1\itf@tp\in@@#2\in@@
3165 }% \itf@removefromstart

```

3166 &lt;/etoolbox&gt;

## 22 History

**[2010/10/27 v1.2]**

- **interfaces-hypbmsec** needed some corrections.
- The **offset** keys for fancyhdr didn't work. Fixed.
- Correction of conversion **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 L<sup>A</sup>T<sub>E</sub>X. **interfaces** is organised as different modules which are loaded after the master package is loaded. Each module can be loaded separately.

\sectionformat: interface for package `titlesec` to modifies the titles produced by the sectionning commands (and their bookmarks).

\pagesetup: interface for package `fancyhdr` to modify the current page style or create new page styles.

\tocsetup: interface for package `tocloft` to fine tune the aspect of the table of contents, list of figures / of tables. \tocsetup also applies to \shorttableofcontents provided by the `shorttoc` package.

\repeatcell: to create ‘dynamic’ tabulars, possibly with formula computation (\numexpr, \pgfmathparse or \FPeval formulas can be plugged in the tabular or array).

\rowfont: to modify the font and/or the alignment of each cell in a tabular/array row.

+ some macros to make easier the use of ε-T<sub>E</sub>X \marks registers (useful with \sectionformat and \pagesetup).

**interfaces** is mainly based on `scrfile.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, 824, <u>836</u>                                                                                              |
| \& . . . . .                         | 3093                                                                                                                        |
| \@@input . . . . .                   | 2510                                                                                                                        |
| \@addtoreset . . . . .               | 675, 683, 721, <u>723</u>                                                                                                   |
| \@break@tfor . . . . .               | 193                                                                                                                         |
| \@cftmakeloftitle_(patch) . . . . .  | <u>1826</u>                                                                                                                 |
| \@cftmakelottitle_(patch) . . . . .  | <u>1826</u>                                                                                                                 |
| \@cftmaketoctitle_(patch) . . . . .  | <u>1826</u>                                                                                                                 |
| \@cftpnumoff . . . . .               | 1691, 1808                                                                                                                  |
| \@cftpnumon . . . . .                | 1690, 1807                                                                                                                  |
| \@clsextension . . . . .             | <u>2481</u>                                                                                                                 |
| \@defaultunits . . . . .             | .. 147, 148, 199, 214, <u>871</u> , 3022, 3028, 3034, 3052                                                                  |
| \@elt . . . . .                      | 1140                                                                                                                        |
| \@expandtwoargs . . . . .            | 193, 208, 209                                                                                                               |
| \@filef@und . . . . .                | 2510                                                                                                                        |
| \@filelist . . . . .                 | 2435, 2440                                                                                                                  |
| \@firstofone . . . . .               | 1929, 2204                                                                                                                  |
| \@flushglue . . . . .                | 377                                                                                                                         |
| \@fourthoffive . . . . .             | 1064                                                                                                                        |
| \@ifclassloaded . . . . .            | 2485                                                                                                                        |
| \@ifpackageloaded . . . . .          | 2484                                                                                                                        |
| \@input_(patch) . . . . .            | <u>2508</u>                                                                                                                 |
| \@mainaux . . . . .                  | 909, 1023, 1028                                                                                                             |
| \@nobreakfalse . . . . .             | 1129                                                                                                                        |
| \@onelevel@sanitize . . . . .        | 2090                                                                                                                        |
| \@onlypreamble . . . . .             | 52, <u>2429</u>                                                                                                             |
| \@pkgextension . . . . .             | <u>2481</u>                                                                                                                 |
| \@tfor . . . . .                     | 191, 2641, 2680,<br>2694, 2707, 2724, 2790, 2808, 2823, 2845, 2867,<br>2887, 2905, 2925, 2942, 2965, 2984, 3002, 3025, 3049 |
| \@whilenum . . . . .                 | 1939, 1948                                                                                                                  |
| \bookmark_(patch) . . . . .          | <u>908</u>                                                                                                                  |
| \~ . . . . .                         | 1920                                                                                                                        |
| <b>A</b>                             |                                                                                                                             |
| \active . . . . .                    | 1893, 1924                                                                                                                  |
| \addcontentsline_(patch) . . . . .   | <u>823</u>                                                                                                                  |
| \AddLineBeginMainAux . . . . .       | 555                                                                                                                         |
| \adl@arrayrestore . . . . .          | 76, 83                                                                                                                      |
| \adl@endarray . . . . .              | 76, 83                                                                                                                      |
| \AfterFile . . . . .                 | 562, 1683, <u>2492</u>                                                                                                      |
| \aftergroup . . . . .                | 450, 978, 1019, 1883                                                                                                        |
| \AfterPackage . . . . .              | 55, 68, 72, 89, 246, 334,<br>337, 344, 379, 443, 459, 819, 1075, 2203, 2456, 2472                                           |
| \allpgfmodules . . . . .             | <u>468</u> , 521, 538                                                                                                       |
| \alltikzlibraries . . . . .          | 469, <u>522</u>                                                                                                             |
| \alltikzpackages . . . . .           | <u>465</u> , 517, 520, <u>535</u>                                                                                           |
| \appendices . . . . .                | <u>2387</u>                                                                                                                 |
| \apptocmd . . . . .                  | 98, 118, <u>1847</u>                                                                                                        |
| <b>B</b>                             |                                                                                                                             |
| \BeforeFile . . . . .                | 558, 1679, <u>2492</u>                                                                                                      |
| \BeforePackage . . . . .             | 38, <u>2466</u>                                                                                                             |
| \BKM@ref . . . . .                   | 1066                                                                                                                        |
| \BKM@color . . . . .                 | 940, 950, 959, <u>971</u>                                                                                                   |
| \BKM@dest . . . . .                  | 1062, 1064                                                                                                                  |
| <b>C</b>                             |                                                                                                                             |
| \BKM@page . . . . .                  | 1046                                                                                                                        |
| \bookmark . . . . .                  | 8, 908, 911, 914, 1033, 1050, 1868                                                                                          |
| \BookmarkAtEnd . . . . .             | 1029                                                                                                                        |
| \bookmarkget . . . . .               | 987, 989, 997, 999                                                                                                          |
| \bookmarksetup . . . . .             | 7, 854, 856, 858, 946, 965, 992, 1002                                                                                       |
| \botmarks . . . . .                  | 1520, 1539, 1541                                                                                                            |
| \bottomtitles . . . . .              | 7, <u>864</u>                                                                                                               |
| \bottomtitlespace . . . . .          | 874                                                                                                                         |
| \BS@BKM@FontStyle . . . . .          | 942, 954, 962, 972, 985, 995                                                                                                |
| <b>D</b>                             |                                                                                                                             |
| \C . . . . .                         | 187, 1950, 1957, 2031, 2152                                                                                                 |
| \c . . . . .                         | 1950, 1952, 2155                                                                                                            |
| \Centering . . . . .                 | 390, 393                                                                                                                    |
| \centering . . . . .                 | 355, 358                                                                                                                    |
| \cftdot . . . . .                    | 1597                                                                                                                        |
| \cftdotfill . . . . .                | 1793                                                                                                                        |
| \cftdotsep . . . . .                 | 1594, 1797                                                                                                                  |
| \cftnodots . . . . .                 | 1596, 1798, 1799                                                                                                            |
| \changefont . . . . .                | 3, <u>243</u> , 271                                                                                                         |
| \chapter . . . . .                   | 1155, 1756, 2410                                                                                                            |
| \color . . . . .                     | 232, 233, 274, 278, 1223, 1243, 2736, 2760, 2762                                                                            |
| \colorlet . . . . .                  | 229                                                                                                                         |
| \column . . . . .                    | 1950, 2031, 2062                                                                                                            |
| \columnsep . . . . .                 | 1614, 1643                                                                                                                  |
| \columnseprule . . . . .             | 1613, 1638                                                                                                                  |
| \columnseprulecolor . . . . .        | 1615                                                                                                                        |
| \contentsname . . . . .              | 1672, 1715, 1716, 1827, 1883                                                                                                |
| \count . . . . .                     | 147                                                                                                                         |
| \countdef . . . . .                  | 1931, 1932, 1933                                                                                                            |
| \countvalues . . . . .               | 12, <u>1895</u>                                                                                                             |
| \csepreatpto . . . . .               | 2557, <u>3104</u>                                                                                                           |
| \csgappto . . . . .                  | 2219, 2491                                                                                                                  |
| \csgpreappto . . . . .               | <u>3104</u>                                                                                                                 |
| \cspreatpto . . . . .                | 2547, <u>3104</u>                                                                                                           |
| \csxpreappto . . . . .               | <u>3104</u>                                                                                                                 |
| \CT@cell@color . . . . .             | 2205, 2206                                                                                                                  |
| \CT@cellc . . . . .                  | 2203                                                                                                                        |
| \CT@color . . . . .                  | 2205                                                                                                                        |
| \CT@end . . . . .                    | 70, 76                                                                                                                      |
| \currentgrouplevel . . . . .         | 944, 977                                                                                                                    |
| \currentgroupype . . . . .           | 413, 418                                                                                                                    |
| <b>D</b>                             |                                                                                                                             |
| \default@color . . . . .             | 233                                                                                                                         |
| \defcounter . . . . .                | 2858                                                                                                                        |
| \DefineVerbatimEnvironment . . . . . | 2354, 2356, 2358                                                                                                            |
| \diaghead . . . . .                  | 2198                                                                                                                        |
| \dimen . . . . .                     | 2277, 2278,<br>2279, 2286, 2295, 2297, 2300, 2301, 2306, 2314, 2317                                                         |
| \dimendef . . . . .                  | 2288, 2289, 2290                                                                                                            |
| \dump . . . . .                      | 115                                                                                                                         |
| <b>E</b>                             |                                                                                                                             |
| \embedfile . . . . .                 | 2430, 2449                                                                                                                  |
| \embedthisfile . . . . .             | 18, 2425, 2428, 2429                                                                                                        |
| \endarray . . . . .                  | 68                                                                                                                          |
| \endlinechar . . . . .               | 560, 564                                                                                                                    |

|                                 |                                                 |                                                      |                                                  |
|---------------------------------|-------------------------------------------------|------------------------------------------------------|--------------------------------------------------|
| \eprerappto . . . . .           | 3096                                            | \interitem . . . . .                                 | 16, 2215                                         |
| \eremovefrom . . . . .          | 3118                                            | \itf@AfterPackage . . . . .                          | 65                                               |
| \eremovefromstart . . . . .     | 3142                                            | \itf@ampactive . . . . .                             | 1918, 1921, 1980, 2002                           |
| \everycr . . . . .              | 280, 283, 296, 297, 307                         | \itf@BKM@enable . . . . .                            | 918,                                             |
|                                 |                                                 | 920, 921, 922, 923, 924, 925, 1020, 1021, 1025, 1027 |                                                  |
| <b>F</b>                        |                                                 | \itf@BKM@pageref . . . . .                           | 1033                                             |
| \f@baselineskip . . . . .       | 258, 261, 266                                   | \itf@BKM@ref . . . . .                               | 1050                                             |
| \f@linespread . . . . .         | 258, 261, 264, 265, 266, 267, 268, 269          | \itf@BKM@restore . . . . .                           | 978, 1006, 1010, 1019                            |
| \f@size . . . . .               | 258, 261, 266                                   | \itf@BKM@write . . . . .                             | 909, 1018                                        |
| \fancy@setoffs . . . . .        | 1448, 1451, 1455, 1458, 1461, 1464, 1467, 1470  | \itf@cell@c . . . . .                                | 388, 395                                         |
| \fancyhf . . . . .              | 1167                                            | \itf@cell@c . . . . .                                | 343                                              |
| \fancyhoffset . . . . .         | 1168                                            | \itf@cell@calc . . . . .                             | 2119                                             |
| \figure . . . . .               | 1762                                            | \itf@cell@formulaerror . . . . .                     |                                                  |
| \firstmarks . . . . .           | 1515, 1528, 1538, 1541                          | 2046, 2047, 2049, 2072, 2081, 2084, 2089, 2118       |                                                  |
| \fontdimen . . . . .            | 175, 181                                        | \itf@cell@FPcalc . . . . .                           | 2085, 2119                                       |
| \fontencoding . . . . .         | 249                                             | \itf@cell@getformula . . . . .                       | 2051, 2069                                       |
| \fontfamily . . . . .           | 251                                             | \itf@cell@init . . . . .                             | 1940,                                            |
| \fontseries . . . . .           | 253                                             | 1943, 1949, 1950, 1954, 1956, 1960, 1980, 2002       |                                                  |
| \fontshape . . . . .            | 255                                             | \itf@cell@insert . . . . .                           | 1981, 2003, 2140                                 |
| \footrule . . . . .             | 1245, 1246, 1247, 1248                          | \itf@cell@j . . . . .                                | 404, 409                                         |
| \footrulewidth . . . . .        | 1251, 1252, 1253                                | \itf@cell@j . . . . .                                | 343                                              |
| \FPeval . . . . .               | 2084, 2113, 2134, 2135                          | \itf@cell@L . . . . .                                | 380, 387                                         |
| \FPresult . . . . .             | 2135, 2136, 2138                                | \itf@cell@l . . . . .                                | 343                                              |
| \futurelet . . . . .            | 103, 158, 2747, 2754, 2770, 2778                | \itf@cell@numcalc . . . . .                          | 2062, 2063, 2078, 2119                           |
|                                 |                                                 | \itf@cell@pgfcalc . . . . .                          | 2082, 2119                                       |
| <b>G</b>                        |                                                 | \itf@cell@R . . . . .                                | 396, 403                                         |
| \getrefbykeydefault . . . . .   | 1046                                            | \itf@cell@r . . . . .                                | 343                                              |
| \getthebotmarks . . . . .       | 1538                                            | \itf@cellfont . . . . .                              | 338                                              |
| \getthefirstmarks . . . . .     | 1538                                            | \itf@celllalign . . . . .                            |                                                  |
| \gettemarks . . . . .           | 11, 1528, 1538, 1539, 1540, 1544, 1545          | 305, 318, 338, 346, 354, 362, 370, 381, 389, 397     |                                                  |
| \getthetopmarks . . . . .       | 1538                                            | \itf@celllallign . . . . .                           | 405                                              |
| \gluestretchorder . . . . .     | 430, 435                                        | \itf@cellleft . . . . .                              | 284,                                             |
| \gluetomu . . . . .             | 172                                             | 294, 295, 302, 320, 338, 348, 350, 351, 356, 358,    |                                                  |
| \gpreappto . . . . .            | 3096                                            | 359, 364, 366, 367, 370, 372, 373, 383, 385,         |                                                  |
| \gremovefrom . . . . .          | 3118                                            | 386, 391, 393, 394, 399, 401, 402, 405, 407, 408     |                                                  |
| \gremovefromstart . . . . .     | 3142                                            | \itf@cellralign . . . . .                            | 306,                                             |
|                                 |                                                 | 327, 338, 349, 357, 365, 371, 384, 392, 400, 406     |                                                  |
| <b>H</b>                        |                                                 | \itf@cellright . . . . .                             | 285, 303, 326, 338                               |
| \hbs@do_(patch) . . . . .       | 1082                                            | \itf@cellrightfalse . . . . .                        | 323, 330                                         |
| \hbs@process_(patch) . . . . .  | 1092                                            | \itf@cellrighttrue . . . . .                         | 316                                              |
| \headrule . . . . .             | 1225, 1226, 1227, 1228                          | \itf@celltransposefalse . . . . .                    | 2056                                             |
| \headrulewidth . . . . .        | 1231, 1232, 1233                                | \itf@cft@label . . . . .                             | 1854                                             |
| \headwidth . . . . .            | 1232, 1253, 1476, 1477, 1478, 1481, 1482, 1485  | \itf@cft@newlistof . . . . .                         | 1573, 1705, 1706, 1707, 1848                     |
| \hfilneg . . . . .              | 436                                             | \itf@cft@newsection . . . . .                        |                                                  |
| \htruncadjust . . . . .         | 17, 2263                                        | 705, 1775, 1814, 1817, 1819, 1822, 1824              |                                                  |
| \htruncwidth . . . . .          | 2329                                            | \itf@cft@secname . . . . .                           | 1751, 1752, 1754, 1807, 1808                     |
| \HyperRaiseLinkHook . . . . .   | 885, 886, 887, 888                              | \itf@cft@sectionname . . . . .                       | 1751                                             |
| \HyperRaiseLinkLength . . . . . | 889                                             | \itf@countvalues . . . . .                           | 1896, 1898, 1902, 1904                           |
|                                 |                                                 | \itf@diaghead . . . . .                              | 2179, 2183, 2195                                 |
| <b>I</b>                        |                                                 | \itf@everycr@tok . . . . .                           | 280, 283, 307, 308                               |
| \if@nobreak . . . . .           | 1123                                            | \itf@expandonceifcs . . . . .                        | 137, 2039, 2067                                  |
| \ifcsundef . . . . .            | 595,                                            | \itf@flush . . . . .                                 | 349, 356, 357, 364, 384, 391, 392, 399, 411, 422 |
|                                 | 598, 634, 681, 688, 716, 1192, 1196, 1502, 2237 | \itf@getthemarks . . . . .                           | 1531, 1535, 1537                                 |
| \ifcsvoid . . . . .             | 775                                             | \itf@getvalue . . . . .                              | 1905, 1945, 1958                                 |
| \ifitf@cellright . . . . .      | 311, 322                                        | \itf@hbs@chkclass . . . . .                          | 1116, 1121, 1124                                 |
| \ifitf@celltranspose . . . . .  | 1917, 1961, 2178                                | \itf@hbs@n@breakfix . . . . .                        | 1125, 1126                                       |
| \ifitf@colortbl . . . . .       | 343, 344, 412, 428                              | \itf@hbs@next . . . . .                              | 1093, 1102                                       |
| \ifmarksequal . . . . .         | 11, 1541                                        | \itf@hbs@nobreakfix . . . . .                        | 1123, 1125, 1131                                 |
| \ifnotempty . . . . .           | 2468                                            | \itf@hfil . . . . .                                  | 424, 441, 445                                    |
| \ifttl@explicit . . . . .       | 643                                             | \itf@hfill . . . . .                                 | 425, 446                                         |
| \ifundef . . . . .              | 1155, 2410, 2457                                | \itf@hskip . . . . .                                 | 426, 447                                         |
| \ignorespaces . . . . .         | 2008, 2256, 2276, 2296                          | \itf@htruncadjust . . . . .                          | 2264, 2266, 2324                                 |
| \in@ . . . . .                  | 193, 208, 209, 535, 538, 3139, 3140, 3161, 3163 | \itf@htruncwidth . . . . .                           | 2330, 2332, 2337, 2339                           |

|                                               |                                                                          |                                        |                                                                 |
|-----------------------------------------------|--------------------------------------------------------------------------|----------------------------------------|-----------------------------------------------------------------|
| \itf@ignorehfil . . . . .                     | 349, 357, 365, 384, 392, 400, 423                                        | \itf@units@scale . . . . .             | 217, 2872, 2891, 2910, 2929, 2951, 2971, 2989, 3010, 3032, 3057 |
| \itf@justify . . . . .                        | 370, 372, 375, 378                                                       | \itf@units@normalize . . . . .         | 219, 220                                                        |
| \itf@last@from@next . . . . .                 | 738, 1119, 1132                                                          | \itf@usetikzcode . . . . .             | 524, 533                                                        |
| \itf@makecell . . . . .                       | 1929, 1930, 2191                                                         |                                        |                                                                 |
| \itf@marks@new . . . . .                      | 1494, 1500, 1508                                                         |                                        |                                                                 |
| \itf@message . . . . .                        | 2006, 2010                                                               | <b>J</b>                               |                                                                 |
| \itf@musk . . . . .                           | 129, 1593, 1594, 1795, 1796                                              | \justifying . . . . .                  | 405, 407                                                        |
| \itf@nohfil . . . . .                         | 440, 442, 444, 450                                                       |                                        |                                                                 |
| \itf@number . . . . .                         | 146, 2017, 2036, 2064, 2227                                              |                                        |                                                                 |
| \itf@pagesetup . . . . .                      | 1172, 1173, 1175, 1177                                                   |                                        |                                                                 |
| \itf@patch . . . . .                          | . 103, 826, 908, 1082, 1092, 1826, 1833, 1840, 1847                      |                                        |                                                                 |
| \itf@pgfk@color . . . . .                     | 2735, 2743, 2747, 2754                                                   |                                        |                                                                 |
| \itf@pgfk@colorcmd . . . . .                  | 2758, 2766, 2770, 2778                                                   |                                        |                                                                 |
| \itf@pgfkeys@aliaserror . . . . .             | 2629, 2631, 2637                                                         |                                        |                                                                 |
| \itf@pgfkeys@case@three . . . . .             | 2521, 2527                                                               |                                        |                                                                 |
| \itf@pgfkeys@default@blank@required . . . . . | 2569, 2578, 2583, 2612                                                   |                                        |                                                                 |
| \itf@pgfkeys@ifaliasexists . . . . .          | 2619, 2625, 2630, 2639                                                   |                                        |                                                                 |
| \itf@preappto . . . . .                       | 3109                                                                     |                                        |                                                                 |
| \itf@preapptocmd . . . . .                    | 96, 106, 826                                                             |                                        |                                                                 |
| \itf@prepnext@tok . . . . .                   | 311                                                                      |                                        |                                                                 |
| \itf@rem@mu . . . . .                         | 165, 169                                                                 |                                        |                                                                 |
| \itf@removefrom . . . . .                     | 3122, 3136                                                               |                                        |                                                                 |
| \itf@removefromstart . . . . .                | 3146, 3160                                                               |                                        |                                                                 |
| \itf@removehfil . . . . .                     | 346, 354, 362, 381, 389, 397, 423                                        |                                        |                                                                 |
| \itf@repeatcell . . . . .                     | 1925, 1927, 2009                                                         |                                        |                                                                 |
| \itf@resetlastsections . . . . .              | 1140, 1143, 1148                                                         |                                        |                                                                 |
| \itf@rowfont . . . . .                        | 281, 282                                                                 | <b>M</b>                               |                                                                 |
| \itf@rowfont@reset . . . . .                  | 297, 300                                                                 | \makecell . . . . .                    | 1930                                                            |
| \itf@scaledimen . . . . .                     | 2299, 2314, 2326, 2328                                                   | \marks . . . . .                       | 43, 1496, 1504                                                  |
| \itf@scalefactor . . . . .                    | 2325, 2327                                                               | \marksthe . . . . .                    | 11, 768, 1492                                                   |
| \itf@sectionformat . . . . .                  | 572, 573, 575                                                            | \marksthecs . . . . .                  | 1498                                                            |
| \itf@setbottomtitlespace . . . . .            | 867, 870, 873                                                            | \maxdimen . . . . .                    | 2255                                                            |
| \itf@setlist . . . . .                        | 2219, 2220, 2222, 2243                                                   | \muexpr . . . . .                      | 179                                                             |
| \itf@setlist@error . . . . .                  | 2238, 2244, 2249                                                         | \multicolssep . . . . .                | 1611                                                            |
| \itf@sk . . . . .                             | 128, 868, 871, 874, 3022, 3023, 3028, 3030, 3032, 3035, 3037, 3052, 3058 | \muskip . . . . .                      | 171, 184, 203                                                   |
| \itf@strip@at . . . . .                       | 132, 621, 1083                                                           | \mutoglu . . . . .                     | 179                                                             |
| \itf@strip@dot . . . . .                      | 132, 174, 180                                                            |                                        |                                                                 |
| \itf@strip@mu . . . . .                       | 164, 1594, 1796                                                          | <b>N</b>                               |                                                                 |
| \itf@tk . . . . .                             | 130, 1474                                                                | \newcounter . . . . .                  | 682                                                             |
| \itf@tocloft . . . . .                        | 1695, 1827, 1834, 1841, 1854, 1858                                       | \newlist . . . . .                     | 2247                                                            |
| \itf@ttl@BKM . . . . .                        | 827, 837, 863                                                            | \newlistentry . . . . .                | 1826                                                            |
| \itf@ttl@createsection . . . . .              | 694, 717                                                                 | \newmarks . . . . .                    | 1509                                                            |
| \itf@ttl@getformat . . . . .                  | 586                                                                      | \newmuskip . . . . .                   | 129                                                             |
| \itf@ttl@getspacing . . . . .                 | 586                                                                      | \newskip . . . . .                     | 128, 423                                                        |
| \itf@ttl@newsection . . . . .                 | 594, 713                                                                 | \newtoks . . . . .                     | 130, 280                                                        |
| \itf@ttl@sectionformat . . . . .              | 585                                                                      | \noalign . . . . .                     | 281, 294, 301                                                   |
| \itf@ttl@setkey . . . . .                     | 602, 606, 609, 610, 611, 612, 623, 624, 625, 626, 627                    | \normalbaselineskip . . . . .          | 1274                                                            |
| \itf@ttl@setup . . . . .                      | 586                                                                      | \normalcolor . . . . .                 | 1648, 2738, 2764, 2773, 2781, 2786                              |
| \itf@ttl@titleclass . . . . .                 | 661, 721, 723                                                            |                                        |                                                                 |
| \itf@ttl@titleclass@i . . . . .               | 664, 665, 667, 679                                                       | <b>P</b>                               |                                                                 |
| \itf@ttl@titleclass@ii . . . . .              | 675, 680, 693                                                            | \pagesetup . . . . .                   | 10, 1169, 1211, 1474                                            |
| \itf@ttl@update . . . . .                     | 589, 629                                                                 | \pagestyle . . . . .                   | 1184, 1192, 1213, 1483                                          |
| \itf@undefinedpagestyle . . . . .             | 1193, 1197, 1200, 1203                                                   | \paragraph . . . . .                   | 1760                                                            |
| \itf@units@mu . . . . .                       | 184, 1593, 1795                                                          | \patchcmd . . . . .                    | 105, 127, 908                                                   |
| \itf@units@mutopt . . . . .                   | 171, 212                                                                 | \pdfstringdefDisableCommands . . . . . | 247                                                             |
| \itf@units@normalize . . . . .                | 225                                                                      | \pgfkeys@case@three . . . . .          | 2521, 2527, 2607, 2608, 2617                                    |
| \itf@units@pt . . . . .                       | 203                                                                      | \pgfkeys@exp@call . . . . .            | 2522, 2528, 2567                                                |
| \itf@units@pttому . . . . .                   | 171, 197                                                                 | \pgfkeys@global@temp . . . . .         | 2548, 2555, 2558, 2565                                          |
|                                               |                                                                          | \pgfkeys@unpack . . . . .              | 2520, 2526, 2570, 2582                                          |
|                                               |                                                                          | \pgfkeys@addvalue . . . . .            | 2544                                                            |
|                                               |                                                                          | \pgfkeysEaddvalue . . . . .            | 1667, 1673, 2052, 2544                                          |

|                                                |                                             |                                |                       |
|------------------------------------------------|---------------------------------------------|--------------------------------|-----------------------|
| <code>\pgfkeysEsetvalue</code>                 | 2544                                        | T                              | 1919, 1923            |
| <code>\pgfkeysmeaning</code>                   | 2535                                        | <code>\table</code>            | 1764                  |
| <code>\pgfkeysnovalue</code>                   | 1584, 2593, 2623, 2732, 2740, 2748, 2755    | <code>\textfont</code>         | 175, 181              |
| <code>\pgfkeysvalueof@unexpanded</code>        | 2540, 2683, 2697, 2793, 2870, 2908          | <code>\textheight</code>       | 871                   |
| <code>\pgfkeysvaluerequired</code>             | 2598                                        | <code>\the@dimen</code>        | 2270, 2308, 2317      |
| <code>\pgfmathparse</code>                     | 2081, 2109, 2129                            | <code>\thebotmarks</code>      | 1513                  |
| <code>\pgfmathresult</code>                    | 2130                                        | <code>\thefirstmarks</code>    | 11, 1513              |
| <code>\preappto</code>                         | 3096                                        | <code>\thepage</code>          | 1042, 1059            |
| <code>\prepnext@tok</code>                     | 332, 335, 336                               | <code>\thetopmarks</code>      | 1513                  |
| <code>\pretocmd</code>                         | 97, 118, 1082, 1092, 1694, 1826, 1833, 1840 | <code>\thispagestyle</code>    | 1196                  |
| <b>R</b>                                       |                                             |                                |                       |
| <code>\R</code>                                | 1940, 1944, 2147                            | <code>\tikzset</code>          | 461                   |
| <code>\r</code>                                | 1940, 1942, 2150                            | <code>\titleclass</code>       | 554, 669, 672, 697    |
| <code>\RaggedLeft</code>                       | 398, 401                                    | <code>\titleformat</code>      | 554                   |
| <code>\raggedleft</code>                       | 363, 366                                    | <code>\titlespacing</code>     | 554                   |
| <code>\RaggedRight</code>                      | 382, 385                                    | <code>\tocloftpagestyle</code> | 1575                  |
| <code>\raggedright</code>                      | 347, 350                                    | <code>\tocsetup</code>         | 9, 9, 1561, 1832      |
| <code>\raisedhyperdef</code>                   | 7, 881, 1864                                | <code>\toksdef</code>          | 1934                  |
| <code>\rem@pt</code>                           | 265                                         | <code>\topmarks</code>         | 1525, 1540            |
| <code>\remove@to@nnil</code>                   | 142, 1900, 1910, 1912, 2087                 | <code>\truncate</code>         | 2258                  |
| <code>\removefrom</code>                       | 23, 3118                                    | <code>\TruncateMarker</code>   | 2253                  |
| <code>\removefromstart</code>                  | 2012, 2014, 3142                            | <code>\truncText</code>        | 17, 2253              |
| <code>\RenewEnviron</code>                     | 2457, 2458                                  | <code>\ttl@getkeys</code>      | 631, 632              |
| <code>\RenewEnviron</code>                     | 2458                                        | <code>\ttl@spacing@ii</code>   | 654                   |
| <code>\repeatcell</code>                       | 12, 1917, 2068, 2100, 2103                  | <code>\tlh@runinifempty</code> | 566                   |
| <code>\rightskip</code>                        | 376                                         | <code>\tlp@append</code>       | 550, 556              |
| <code>\row</code>                              | 1940, 2063                                  | <b>U</b>                       |                       |
| <code>\rowfont</code>                          | 3, 280                                      | <code>\unkern</code>           | 429                   |
| <b>S</b>                                       |                                             |                                |                       |
| <code>\scr@load@hook</code>                    | 2487, 2509, 2511                            | <code>\unless</code>           | 1127, 1128            |
| <code>\scr@set@load@hook_(optimisation)</code> | 2491                                        | <code>\unskip</code>           | 434, 2256, 2276, 2296 |
| <code>\sectionformat</code>                    | 5, 571, 856                                 | <code>\usepgfmodule</code>     | 521, 539              |
| <code>\selectfont</code>                       | 244                                         | <code>\usetikz</code>          | 5, 525, 526, 2109     |
| <code>\set@fontsize</code>                     | 258, 261, 266                               | <code>\usetikzlibrary</code>   | 511, 522, 540         |
| <code>\setlist</code>                          | 16                                          | <b>V</b>                       |                       |
| <code>\setlist_(extension,_star_form)</code>   | 2218                                        | <code>\vsize</code>            | 2255                  |
| <code>\skip</code>                             | 171, 184, 203                               | <code>\vskip</code>            | 1233, 1252, 1253      |
| <code>\stretch</code>                          | 413, 415                                    | <b>W</b>                       |                       |
| <code>\strip@pt</code>                         | 2325                                        | <code>\write</code>            | 909, 1023, 1028       |
| <code>\strut</code>                            | 1209                                        | <b>X</b>                       |                       |
| <code>\ subparagraph</code>                    | 1761                                        | <code>\xpreappto</code>        | 3096                  |
| <code>\subsubfigure</code>                     | 1763                                        | <code>\xremovefrom</code>      | 3118                  |
| <code>\subsubsection</code>                    | 1759                                        | <code>\xremovefromstart</code> | 3142                  |
| <code>\subtable</code>                         | 1765                                        |                                |                       |