

The **interfaces*** package

interfaces with keys for other packages (FC)

<florent.chervet at free.fr>

2010/10/20 – version 1.0

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

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

~~~~~ Example ~~~~~

```
\pagesetup[document]{%
  rules=.5pt,
  foot/left/odd=\arabic{page},
  foot/right/even=\arabic{page}/\pageref{LastPage},
  head/font=\color{gray}{.85}\bfseries\small,
  head/left/even=\leftmark,
  head/left=\thefirstmarks{MiniSection},% interfaces-marks
}
\pagesetup[plain]{% for chapters...
  pagestyle=document,% preloaded settings
  head=,
  foot/left/odd = \roman{page},
  norules, head/rule=.4pt
  head/center=This is the title,
}
\sectionformat\section[hang]{
  label=\arabic{section},
  labelsep=,
  font=\bfseries\Large,
  bookmark={bold,color=blue},% interfaces-bookmark
  top=6pt plus3pt minus2pt,
  bottom=3pt plus2pt,
  break=\goodbreak,
  mark=\markboth{#1}{#1},%
}
\sectionformat\subsection[hang]{
  label=, labelsep=,
  font=\bfseries,
  bookmark=italic,
  top*=1.2,% a 120% scaling
  bottom+=0pt plus4pt,% add to previous value
  mark=\markright{#1},
}

\sectionformat\MiniSection{% new sectionning command
  class=straight[\subparagraph],
  shape=block,
  label=,
  top=3pt plus2pt, bottom=1pt,
  font=\bfseries\sffamily,
  bookmark=off,% or false or none
  break=\allowbreak,
  mark=\markthe{MiniSection}{#1},% see interfaces-marks
}
\bottomtitles=.2\textheight

\tocsetup{
  depth=3, title=Contents, pagenumbers=on,
  pagestyle=plain,% first page style: tocloftpagestyle
  title/top=1.5ex plus.5ex,
  title/bottom=0pt,
  section/indent=1em,
  section/number/width=1.5em,
  section/font+=\larger\bfseries,% append to previous setting
  section/leaders,section/dotsep,
  subsection/indent=2em,
  subsection/number/width=2.5em,
  subsection/font=\scriptsize,
  subsection/leaders,subsection/dotsep,
  subsection/pagenumbers=off,% no page numbers
  subsection/dotsep,% no leader
  columns=2,% same as twocolumns
  columns/sep=.5em,
  columns/rule=1pt,columns/rule/color=blue,
}
\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 ^A T _E X	32
10.3	Empty rows	14	21.3.1 \change font	
10.4	color palette	14	21.3.2 \rowfont	
10.5	tabulargcalc-like tables (with dcolumn package)		21.4 Interface TikZ	37
	and pgfmth	14	21.4.1 Catcode 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	39
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	18	21.5.4 \bottomtitles	
17	Interface pgfkeys	19	21.6 Interface hyperref	46
17.1	/.blank	19	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	49
17.5	/.is multiple choice	20	21.8.1 interfaces-hypbmsec	
17.6	/.is enum	21	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	22	21.10 marks	56
17.11	/.store as number	22	21.11 Interface tocloft: \tocsetup	58
17.12	/.store as number in cs	22	21.11.1 Specific handlers for tocloft	
17.13	/.store as glue	22	21.12 Interface makecell: \repeatcell	64
17.14	/.store as dimen	22	21.12.1 countvalues	
17.15	/.store as glue in cs	23	21.12.2 \repeatcell	
17.16	/.store as dimen in cs	23	21.12.3 colortbl: patch for \CT@cellc	
17.17	/.estore as dimen in cs	23	21.13 Interface enumitem	70
18	Interface etoolbox	23	21.14 Interface truncate	71
18.1	New macros provided with \providecommand	23	21.15 interface fancyvrb	73
19	Interface scrfile	24	21.16 Interface appendix	73
19.1	Star forms for \BeforeFile and \AfterFile	24	21.17 Interface embedfile: \embedthisfile	74
19.2	Allowing \BeforeFile and \AfterFile when \@input is invoked	24	21.18 Interface environ	75
20	List of corrected or patched commands	24	21.19 Interface umrand	75
20.1	LaTeX	24	21.20 Interface scrfile	75
20.2	array	25	21.21 Interface pgfkeys	76
20.3	bookmark	25	21.21.1 interfaces-pgfkeys	
20.4	colortbl	25	21.22 Interface etoolbox	89
20.5	hypbmsec	25	22 History	91
20.6	enumitem	26	[2010/10/20 v1.0]	91
20.7	pgfkeys	26	[2010/10/20 v1.0]	91
			23 References	91
			README	91
			24 Index	92

1 Interface L^AT_EX

```
\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:

<code>l</code> = left	or for <i>ragged2e</i> settings: <code>L</code>
<code>c</code> = center	or for <i>ragged2e</i> settings: <code>C</code>
<code>r</code> = right	or for <i>ragged2e</i> settings: <code>R</code>
<code>j</code> = justify	or for <i>ragged2e</i> settings: <code>J</code>

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|>{\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[l]{\itshape}
to left                        &1          &2          &3          &4          & \\
\rowFont[c]{\sffamily\bfseries}% overwrites \mdseries
to center                      &1          &2          &3          &4          & \\
\rowcolor{MediumSpringGreen}
\rowFont[r]{\larger\upshape\bottomstrut}% overwrites \itshape
to right                       &1          &2          &3          &4          & \\ \hline
\end{tabular}
```

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

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

### ~~~~~ EXPLANATIONS FOR L<sup>A</sup>T<sub>E</sub>X ~~~~~

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

```
\hskip...\dollarbegin \the\toks i \ignorespaces #\unskip \the\toks i+1 \relax\dollarend... &
```

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:

$$\underbrace{>\{\backslash\bfseries\color{red}\}}_{\backslash\toks<i>} \text{ r } \underbrace{<\{\color{black}\backslash,\backslash\$}\}_{\backslash\toks<i+1>}}$$

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 `\toks(i)` a control sequence named `\itf@cellleft` and to each such `\toks(i+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 positioned), 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 |
| | | | plohandlers |
| | | | 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#1{\markright{#1}}   break=\goodbreak      %=&gt; this defines \def\sectionbreak{\goodbreak}   thesection=\Roman{chapter}.\arabic{section} } %redefines \def\thesection{...}</pre> | <pre>this is the same as:  shape = hang</pre> |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------|

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

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

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

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

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

~~~~~ **MEMO** ~~~~~

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

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

| | | |
|------------------------|-----------------------|---|
| bookmark/bold | bold bookmark text | the text of the bookmark for that sectionning command will be bold |
| bookmark/italic | italic bookmark text | the text of the bookmark for that sectionning command will be in <i>italic</i> |
| bookmark/color | colored bookmark text | the text of the bookmark for that sectionning command will be colorized |
| bookmark/off | disable the bookmark | no bookmark for that sectionning command alternatively you may set bookmark/level to a huge value... |
| bookmark/style | use a <i>style</i> | the <i>style</i> must be previously declared with <code>\bookmarkdefinestyle</code> |
| bookmark/level | 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**:

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

Furthermore the following code:

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

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

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

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

```

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

```

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

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

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

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

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

6 Interface ⇄ `hypbmsec`

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

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

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

(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`. (For chapters, this is the same if `titlesec` formats the chapter titles, *ie.* if `\sectionformat\chapter` or `\titleformat\chapter` has been invoked. Otherwise, `\lastchaptertitle`, `\lastchapterbookmark` and `\lastchaptertoc` are set at `\stepcounter{chapter}`).

Values are automatically reset to empty 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. I leave the reader to his thoughts about the necessity of such `\marks` in certain situations...

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 (\cftaftertocitle...)
    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  $4.5\mu$  and blank value is the same as \cftnodots
    dot=modifies \cftdot, the dot used for leaders produced by \cftdotfill
    pagestyle=page style for the first page (\tocloftpagestyle)
    mark=the mark command: \cftmarktoc
    example:\chaptermark\contentsname
    before|before*|after|after*=hooks to execute before or after the command \tableofcontents
    twocolumns|columns|multicol=twocolumns=true is the same as columns=2 Makes a multicolumn table of contents. Package multicol is required!
    columns/sep|columns/sep+|columnsep etc.=sets (or add) a  $\langle dimen \rangle$  to \columnsep
    columns/rule|columns/rule+|columnseprule etc.=sets (or add) a  $\langle dimen \rangle$  to \columnseprule
    columns/color|columnseprulecolor=modifies \columnseprulecolor
    multicolsep|multicolsep+ etc.=modifies (or add) a  $\langle glue \rangle$  to \multicolsep
    \multicolsep is the space added before and after multicol environment. Its default value is 0pt plus 4pt minus 3pt.
    multicol/begin|twocolumn/before=hooks just after \begin{multicols} ...
    multicol/end|twocolumn/after=... and before \end{multicols}
    label=the label for reference to the table of contents
    ref=the reference printed by \ref when referencing the table of contents
    (default is \contentsname. In any case, \nameref prints \contentsname.).
    bookmark/text=the bookmark text for the table of contents (if empty, no bookmark is set)
    bookmark/bold|bookmark/italic|bookmark/color=the bookmark style for the table of contents
    bookmark/level=the bookmark level: either a number or section or chapter
    Default is chapter, or section if the class does not provide chapters.
}
```

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

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

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

~~~~~ MEMO ~~~~~

numbered entries are typeset with this pseudo code:

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

and for not numbered entries:

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

**\lofsetup** and **\lotsetup** behave the same.

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

This command is provided to setup the parameters of user-defined lists created with **tocloft**'s **\newlistof** and printed with **\listofX** (where **X** is the name given to the list). The keys are the same as **\tocsetup**, but there is no sub-keys for section, subsection *etc.*.

## 8 Interface **fancyhdr** : **\pagesetup**

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

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

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

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

**foot/left/odd** is the same as **odd/foot/left** *etc.* but **twoside** option is not checked... Even pages are ignored in **oneside** documents: this is **L<sup>A</sup>T<sub>E</sub>X**, 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  $\varepsilon$ -TeX `\marks` registers.  $\varepsilon$ -TeX defines 32 768 marks registers, making the use of them far more comfortable than “old”  $\LaTeX$  `\markboth`, `\markright`, `\leftmark` and `\rightmark`. Who gets the correct – I mean *expected* – result with such commands ? Not me, I can say !

```
\marksthe{<named-mark>}{<content>}
\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.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Those assertions are true only if you mark the section title at `\sectionmark` and at `\sectionbreak`: this is easy when using `interfaces-hypbmsec` and `\lastsectiontitle`...

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`,
- `$\pgfmath$` ( $\leadsto$  `\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>}`  $\leftarrow$  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 `\crr\hline` or `\tabularnewline`...)  
**end of row sequence is never inserted after the last row.**  
`text/row1|text/row2|text/row3 etc.` = If you want to change the text on the specified rows  
`text/col1|text/col2|text/col3 etc.` = If you want to change the text on the specified columns  
`font/row1|font/row2|font/row3 etc.` = If you want to change the font on the specified rows  
`font/col1|font/col2|font/col3 etc.` = If you want to change the font on the specified columns  
`cr/row1|cr/row2|cr/row3 etc.` = If you want to change the end-of-line sequence on the specified rows  
`transpose` = Do you wish to invert rows and columns ? Then `transpose=true` ☺.  
`any other key` = can be used to make calculations using `\numexpr` or `$\pgfmathparse$` or `$$\FPeval$`. Pay attention to put your formula inside curly brackets `{$formula$}` if it contains a comma `,` or an equal `=` sign! }

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10.1 † Conversion to scaled points

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

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

10.2 † Variable length cell

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

| Mother | Father | Children |
|---------|--------|--------------------------|
| Johanne | Georg | Priscilla
John
Tom |

| Mother | Father | Children |
|--------|--------|------------------------------------|
| Nout | Geb | Isis
Osiris
Seth
Nephthys |

10.3 † Empty rows

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

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

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

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

10.4 † color palette

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

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

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

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

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

`\pgfmth` 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{\BMI}\BMI,
  text/row1 = \c, font/row1 = \color{NavyBlue}, cr/row1 = \\ \hline,
  text/col1 = \r, font/col1 = \color{DarkViolet},
} \\ \hline
\end{tabular}\quad
\begin{tabular}{|c|}
\repeatcell{denutrition,skinniness,normal,obesity,serious obesity,morbid obesity}{%
  text=\cellcolor{\c}{\c},\transpose,font=\large\bfseries\bottomstrut}
\end{tabular}

```

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

| denutrition |
|-----------------|
| skinniness |
| normal |
| obesity |
| serious obesity |
| morbid obesity |

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

The same with **transpose**

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

11 Interface **enumitem** : **\setlist**

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

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

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

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

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

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

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

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

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

or

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

\interitem

Just like \mathcal{MS} **\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.

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

`\truncatext` is the same as `\truncate` but it leaves `{text}` to its natural width if it has not to be truncated (*ie.* if `⟨width⟩ ≤ width of ⟨text⟩`). Example:

truncate: `\truncate{⟨5cm⟩}{⟨Hello world!⟩}` Hello word!
truncatext: `\truncatext{⟨5cm⟩}{⟨Hello world!⟩}` 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 | adusted width | γ (truncated?) | Returned width |
|--|----------------|---|--|---|
| text ₁ | w ₁ | $w'_i = \min\left(w_1, \frac{x_1}{\sum x_i} \times \langle total\ width \rangle\right)$ | $\gamma_1 = \begin{cases} 1 & \text{if } w'_1 < w_1 \\ 0 & \text{otherwise} \end{cases}$ | $\omega_1 = w'_1 + \gamma_1 \cdot \Phi \cdot \frac{x_1}{\sum \gamma_i x_i}$ |
| text ₂ | w ₂ | $w'_i = \min\left(w_2, \frac{x_2}{\sum x_i} \times \langle total\ width \rangle\right)$ | $\gamma_2 = \begin{cases} 1 & \text{if } w'_2 < w_2 \\ 0 & \text{otherwise} \end{cases}$ | $\omega_2 = w'_2 + \gamma_2 \cdot \Phi \cdot \frac{x_2}{\sum \gamma_i x_i}$ |
| text ₃ | w ₃ | $w'_i = \min\left(w_3, \frac{x_3}{\sum x_i} \times \langle total\ width \rangle\right)$ | $\gamma_3 = \begin{cases} 1 & \text{if } w'_3 < w_3 \\ 0 & \text{otherwise} \end{cases}$ | $\omega_3 = w'_3 + \gamma_3 \cdot \Phi \cdot \frac{x_3}{\sum \gamma_i x_i}$ |
| text ₄ | w ₄ | $w'_i = \min\left(w_4, \frac{x_4}{\sum x_i} \times \langle total\ width \rangle\right)$ | $\gamma_4 = \begin{cases} 1 & \text{if } w'_4 < w_4 \\ 0 & \text{otherwise} \end{cases}$ | $\omega_4 = w'_4 + \gamma_4 \cdot \Phi \cdot \frac{x_4}{\sum \gamma_i x_i}$ |
| $\Phi = \text{freespace} = \langle total\ width \rangle - \sum w'_i$ | | | | $\sum \omega_i \leq \langle total\ width \rangle$ |

Another example of a tabular made with `\repeatcell`

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

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

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

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

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

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

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

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

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

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

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

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

13 Interface `appendix` and the appendices environment

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

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

14 Interface `embedfile` : `\embedthisfile`

`\embedthisfile`
`\embedthisfile*`

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

`\embedthisfile*` embeds only the main file.

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

15 Interface `environ`

`environ` defines `\Renewenvirom` but I found out it was usefull to have `\ReNewEnvirom` (note the case!)

Therefore: `\let\ReNewEnvirom\RenewEnvirom` 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 on choice can be selected at a time |
| <code>/.is enum</code> | <code>/.is choice</code> shortcut for enumerated keys
also provides the <code>/key/.number</code> key to get the numeric index of the selected choice. |
| <code>/.store as code</code>
<code>/.estore as code</code> | store value (<i>expanded value</i>) in the key. Created <code>+</code> and <code>*</code> keys to append code |
| <code>/.store as code in cs</code>
<code>/.estore as code in cs</code> | store value (<i>expanded value</i>) in a cs-name rather than in the key. Created <code>+</code> and <code>*</code> keys to append code |
| <code>/.store as color</code>
<code>/.store as color command</code> | store the color (<i>the color command</i>) in the key. |
| <code>/.store as color in cs</code>
<code>/.store as color command in cs</code> | the same, store in the cs-name rather than in the key |
| <code>/.store as number</code>
<code>/.estore as number</code> | store an integer in the key (<code>-</code> <code>+</code> and <code>*</code> keys are provided to add, substract and multiply) |
| <code>/.store as number in cs</code>
<code>/.estore as number in cs</code> | the same, but stores into a cs-name rather than in the key |
| <code>/.defcounter</code> | the same, but stores into a L ^A T _E X counter |
| <code>/.store as glue</code>
<code>/.estore as glue</code> | store a glue in the key (<code>+</code> <code>-</code> and <code>*</code> keys are provided to add, substract or scale) |
| <code>/.store as glue in cs</code>
<code>/.estore as glue in cs</code> | the same, store in a given named control sequence |
| <code>/.store as dimen</code>
<code>/.estore as dimen</code> | store a dimen in the key (with <code>+</code> <code>-</code> and <code>*</code> keys for addition, subtraction and scaling) |
| <code>/.store as dimen in cs</code>
<code>/.estore as dimen in cs</code> | the same, store in a given named control sequence rather than in the key itself |

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

17.1 † Handler definition: `/.blank`

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

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

The following make blank value behave the same as default :

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

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

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

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

17.2 † Handler definition: `/.is private`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
\pgfkeys{/path/format={bold,italic,color=...}}
this is the same as:
\pgfkeys{/path/format=bold,/path/format=italic,path/format={color=...}}
```

The `bookmark` key of `\sectionformat` (in `interface-titlesec`) is a good example.

17.6 † Handler definition: `/.is enum`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

When specifying a color, did you say:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

18 Interface ⇄ etoolbox - etoolbox-like commands

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

18.1† New macros provided with `\providecommand`

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

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

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

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

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

this allows loops of the form:

```
\in@true
\@whiles\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^AT_EX

| | Patched command | Type | Importance | Module | ref. page |
|---|--|------|-------------------------------------|-------------------------------|-----------|
| ❶ | <code>\@pkgextension</code>
<code>\@clsextension</code>
<code>\@ifpackageloaded</code>
<code>\@ifclassloaded</code> | | minor
optimisation or correction | <code>interfaces-LaTeX</code> | 75 |

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

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

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

| | Patched command | Type | Importance | Module | ref. page |
|---|-------------------------------|----------------|---------------------------------|--|-----------|
| ❷ | <code>\addcontentsline</code> | <i>prepend</i> | important
but no side effect | <code>interface-titlesec</code>
<code>interface-bookmark</code> | 44 |

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

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

| | Patched command | Type | Importance | Module | ref. page |
|---|----------------------|------|---------------------------------|--------------------------------|-----------|
| ❸ | <code>\@input</code> | | important
but no side effect | <code>interface-scrfile</code> | 76 |

The `\input` primitive (saved by L^AT_EX in control sequence `\@@input`) is surrounded by `\scr@load@hook{before}...\scr@load@hook{after}`:

Original:

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

Modified:

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

20.2 † array

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

`\prepnext@tok` is replaced by `\itf@prepnext@tok` for the purpose of inserting `\itf@celllalign` and `\itf@celllleft` 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 | interface-bookmark | 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 (*ie.* `\aftergroup`) ?

20.4 † colortbl

| | Patched command | Type | Importance | Module | ref. page |
|---|--|--------------|---------------------------------|------------------------------------|-----------|
| ⑥ | <code>\CT@cellc</code>
<code>\endarray</code> | <i>corr.</i> | important
but no side effect | interface-makecell | 70 |

`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>
<code>\hbs@process</code> | <i>prepend</i> | important
but no side effect | interfaces-hypbmsec
interfaces-titlesec | 50 |

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

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

| | Patched command | Type | Importance | Module | ref. page |
|---|--------------------------|----------------|---------------------------------|--|-----------|
| ⑧ | <code>\ttl@select</code> | <i>prepend</i> | important
but no side effect | interfaces-hypbmsec
interfaces-titlesec | 50 |

The update of `\lastsectiontitle`, `\lastsectionbookmark` and `\lastsectiontoc` occurs at the beginning of `\ttl@select`. At that time, those macros receive the values for the current section that were stored at the beginning of `\hbs@process`.

20.6 † enumitem

⑨

| Patched command | Type | Importance | Module | ref. page |
|-----------------------|--------------|-------------------------------------|-------------------------------------|-----------|
| <code>\setlist</code> | <i>redef</i> | minor
optimisation or correction | interfaces-enumitem | 71 |

`\setlist` is redefined by [interfaces-enumitem](#) in order to extend the syntax.

20.7 † pgfkeys.code.tex

⑩

| Patched command | Type | Importance | Module | ref. page |
|--|--------------|------------|------------------------------------|-----------|
| <code>\pgfkeys@unpack</code>
<code>\pgfkeys@case@three</code>
<code>\pgfkeys@exp@call</code> | <i>redef</i> | important | interfaces-pgfkeys | 78 |

`\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   [2010/10/20 v1.0 - 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{hyrbmsec}
19 \itf@tp{environ}
20 \itf@tp{truncate}
21 \itf@tp{appendix}
22 \itf@tp{embedfile}
23 \itf@tp{fancyvrb}
24 \ifrelax\@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   [2010/10/10 v1.0 Basic macros for interfaces package (FC)]
32 \RequirePackage{etex,ltxcmds,etoolbox,pgfkeys,scrfile}
33 \AtEndOfPackage{%
34   \RequirePackage{interfaces-etoolbox}%
35   \RequirePackage{interfaces-scrfile}%
36   \RequirePackage{interfaces-pgfkeys}
37   \RequirePackage{interfaces-tikz}%
38   \RequirePackage{interfaces-umrand}% There is a \BeforePackage here !
39}% \AtEndOfPackage
```

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

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

```

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

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

```

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

```

65 \newrobustcmd*\itf@AfterPackage[1]{%
66   \csname\ltx@ifpackageloaded{#1}\ltx@secondoftwo\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     \crrc \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 risen 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{\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   \let\itf@tp#2%
112   #1\itf@tp{#3}%
113   {\let#2\itf@tp
114     \AtBeginDocument{#4}% in order to keep this info even after \dump
115   }
116   {#5}%
117}% \itf@patch@boundary (for \pretocmd and \apptocmd)
118 \def\itf@patch@inside#1#2#3#4#5#6{%
119   \let\itf@tp#2%
120   #1\itf@tp{#3}{#4}%
121   {\let#2\itf@tp
122     \AtBeginDocument{#5}%
123   }
124   {#6}%
125}% \itf@patch@inside (for \patchcmd)

```

21.2.3 Helper macros

```
126 \newskip\itf@sk
127 \newmuskip\itf@musk
128 \newtoks\itf@tk
129 \newcount\itf@cnt
```

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

```
130 %% H E L P E R M A C R O S
131 \begingroup\makeatother
132 \csgdef{\itf@strip@at}#1@{}
133 \endgroup
134 \def\itf@strip@dot#1.#2\@nil{#1}
```

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

USAGE: `\itf@expandonceifcs\macrotest`

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

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

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

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

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

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

```
144 %% G E T A N U M B E R (if possible)
145 %% USAGE: \itf@number\count<text>\relax\@nnil (like \@defaultunits)
146 \def\itf@number#1#2\relax\@nnil{% \relax is to mimick the syntax of \@defaultunits
147   \afterassignment\itf@number@#1\number0#2\relax\itf@number@
148 }
149 \def\itf@number@#1\relax#2\itf@number@{%
150   \ifblank{#1}
151     {\itf@number@@{}}#2\itf@number@@\@nnil
152     \csname ltx@\ifx\@let@token\itf@number@@
153       first\else second\fi oftwo\endcsname}
154   \ltx@secondoftwo
155 }% \itf@number@
156 \def\itf@number@@#1{\futurelet\@let@token\itf@number@@@}
157 \def\itf@number@@@{%
158   \csname \ifx\@let@token\relax itf@number@@%
159     \else remove@to@nnil%
160   \fi \endcsname
161 }% \itf@number@@@
```

21.2.5 Units and math units

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

```
162%%  U N I T S   F O R   D I M E N S
163 \def\itf@strip@mu{\expandafter\itf@rem@mu\the}
164 \begingroup \makeother\M\@makeother\U
165   \lowercase{%
166     \gdef\itf@rem@mu#1.#2MU{#1\ifnum#2>\z@.#2\fi}%
167   }%
168 \endgroup
```

`\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`.

```
169%% \skip = \itf@units@mutopt \muskip
170 \def\itf@units@pttomu#1{\gluetomu\glueexpr
171   \glueexpr#1\relax*1800/%
172   \expandafter\itf@strip@dot\the\dimexpr
173     \fontdimen6\textfont2*100\relax.\@nil
174   \relax
175 }% \itf@units@pttomu
176 \def\itf@units@mutopt#1{\glueexpr
177   \mutoglue\muexpr#1\relax*
178   \expandafter\itf@strip@dot\the\dimexpr
179     \fontdimen6\textfont2*100\relax.\@nil/1800
180   \relax
181 }% \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.

```
182%% \muskip = \itf@units@mu{ \skip or \muskip or number}
183 \def\itf@units@mu#1#2{\begingroup\makeother\P
184   \makeother\T\makeother\S\makeother\B\makeother\C
185   \makeother\M\makeother\D\makeother\I\makeother\N
186   \edef\itf@tp{#2}%
187   \lowercase{\@tfor\x:={PT}{PC}{IN}{BP}{CM}{MM}{DD}{CC}{SP}\do}{%
188     \@expandtwoargs\in@\x\itf@tp\ifin@\expandafter\@break@tfor\fi
189   }%
190   \expandafter\endgroup
191   \ifin@
192     #1=\itf@units@pttomu{#2}%
193   \else
194     \@defaultunits#1 #2mu\relax\@nnil
195   \fi
196 }% \itf@units@mu
```

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

```
197%% \skip = \itf@units@pt( \muskip or \skip or number )
198 \def\itf@units@pt#1#2{\begingroup\makeother\M\makeother\U
199   \lowercase{\@expandtwoargs\in@{MU}}{#2}%
200   \expandafter\endgroup
201   \ifin@
202     #1=\itf@units@mutopt{#2}%
203   \else
204     \@defaultunits#1 #2pt\relax\@nnil
205   \fi
206 }% \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.

```

207%% glue * 1.5 = glue * 98302 / 65536
208%% glue * 3/2 = glue * 3/2
209\def\itf@units@scale#1!\itf@units@scale@normalize#1/\@nnil/\@nil}
210\def\itf@units@scale@normalize#1/#2/#3\@nil{%
211  \ifx#2\@nnil
212    \number\dimexpr#1pt\relax/65536
213  \else#1/#2
214  \fi
215}% \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`).

```

216 \AtBeginDocument{%
217   \ltx@ifpackageloaded{xcolor}
218     {\@ifundefinedcolor{default}
219       {\colorlet{default}{.}}
220       {}}%
221   {\ltx@ifpackageloaded{color}
222     {\@ifundefined{\string\color @default}
223       {\expandafter\def\csname\string\color @default\endcsname{\default@color}}
224       {}}%
225   {}}%
226 }

227 </base>

```

21.3 † Interface L^AT_EX

```

228 < *LaTeX>
229 \ProvidesPackage{interfaces-LaTeX}
230 [2010/10/10 v1.0 pgfkeys macros for LaTeX (FC)]
231 \RequirePackage{interfaces-base}
232 \itf@pgfkeys

```

21.3.1 `\change font`

`\change font` 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.

```

233 \newrobustcmd*\change font[1]{\itf@pgfkeys
234   \pgfkeys{/itf/LaTeX/change font}{#1}\select font
235 }% \change font
236 \AfterPackage*{hyperref}
237   {\pdfstringdefDisableCommands{\let\change font\@gobble}}

```



```

238 \pgfqkeys{/itf/LaTeX/changefont}{%
239   encoding/.code=\fontencoding{#1},encoding/.value required,%
240   enc/.alias=encoding,%
241   family/.code=\fontfamily{#1},family/.value required,%
242   fam/.alias=family,%
243   series/.code=\fontseries{#1},series/.value required,%
244   ser/.alias=series,%
245   shape/.code=\fontshape{#1},shape/.value required,%
246   sha/.alias=shape,%
247   size/.estore as dimen in cs=f@size,%
248   size/.append code=\set@fontsize\f@linespread\f@size\f@baselineskip,%
249   siz/.alias*=size,%
250   baselineskip/.store as glue in cs=f@baselineskip,%
251   baselineskip/.append code=\set@fontsize\f@linespread\f@size\f@baselineskip,%
252   skip/.alias*=baselineskip,%
253   linespread/.estore as dimen in cs=f@linespread,%
254   linespread/.add code={\ifx\f@linespread\empty\let\f@linespread\@ne\fi}%
255                       {\edef\f@linespread{\expandafter\rem@pt\f@linespread}%
256                        \set@fontsize\f@linespread\f@size\f@baselineskip},%
257   linespread+/.prefix code=\ifx\f@linespread\empty\let\f@linespread\@ne\fi,%
258   linespread-/.prefix code=\ifx\f@linespread\empty\let\f@linespread\@ne\fi,%
259   linespread*/.prefix code=\ifx\f@linespread\empty\let\f@linespread\@ne\fi,%
260   spread/.alias*=linespread,%
261 }% \pgfqkeys definitions for \changefont

```

21.3.2 `\rowfont`

`\rowfont` `\rowfont` uses the control sequences `\itf@celllalign`, `\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@celllalign` 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`.

```

262 \newtoks\itf@everycr@tok % better than colortbl turnaround for \everycr
263 \newcommand*\rowfont{\noalign{\ifnum0='}\fi\itf@rowfont}
264 \newrobustcmd*\itf@rowfont[2][]{%
265   \global\itf@everycr@tok\everycr
266   \global\let\itf@cellleft\itf@cellleft
267   \global\let\itf@cellright\itf@cellright
268   \gdef\itf@cellfont{#2}%
269   \ifcsname itf@cell@#1\endcsname % row alignment
270     \csname itf@cell@#1\endcsname
271   \else\lowercase{%
272     \ifcsname itf@cell@#1\endcsname
273       \csname itf@cell@#1\endcsname
274     \fi
275   }\fi
276   \toks@\expandafter{\itf@cellleft\itf@cellfont}% inside \noalign group ok
277   \xdef\itf@cellleft{\the\toks@}%
278   \global\everycr\expandafter\expandafter\expandafter{%
279     \expandafter\the\expandafter\everycr\expandafter\itf@rowfont@reset}%
280   \ifnum0='{ \fi}% end of noalign group
281 }% \rowfont

```

`\itf@rowfont@reset` This macro resets `\itf@celllalign`, `\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.

```

282 \def\itf@rowfont@reset{%
283   \noalign{%
284     \global\let\itf@cellleft\itf@@cellleft
285     \global\let\itf@cellright\itf@@cellright
286     \global\let\itf@cellfont\@empty
287     \global\let\itf@celllalign\@empty
288     \global\let\itf@cellralign\@empty
289     \global\everycr\itf@everycr@tok
290     \global\itf@everycr@tok{}}%
291   }%
292 }% \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@celllalign`, `\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.

$$\underbrace{>\{\backslash\bfseries\color{red}\}}_{\backslash\text{toks}<i>} \quad \text{r} \quad \underbrace{<\{\backslash\color{black}\backslash,\backslash\$}\}_{\backslash\text{toks}<i+1>}$$

When a column is inserted in the tabular preamble (`\@preamble`), the T_EX 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: $i = \text{count@} = \text{@tempcnta}$: the token to prepare (ie. $\backslash\text{toks}<i+1>$) is the right one of a “normal” column. The switch `\ifitf@cellright` is set to `true`.

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

or: $i = \text{count@} = \text{@tempcnta} + 1$: the token to prepare ($\backslash\text{toks}<i+1>$) 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 $\backslash\text{toks}<i>$ 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 `\ititf@cellright` is reset to `false`. May be `\prepnext@tok` will be expanded again (by `\save@decl`): if it happens, then again $\text{count@} = \text{@tempcnta} + 1$ (same case) but `\ifitf@cellright` is false and nothing is changed.

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

The fact that $|\text{count@} - \text{@tempcnta}| > 1$ tells us that the previous token $\backslash\text{toks}<i>$ 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 `\@one` and initialize the token to prepare ($\backslash\text{toks}\text{count@} = \backslash\text{toks}<i+1>$) to an empty one.

```

293 \newif\ifitf@cellright
294 \def\itf@prepnext@tok{% only if array is loaded
295   \itf@cnt = \count@
296   \advance \itf@cnt -\@tempcnta
297   \ifcase \itf@cnt           % (case 0): previous token is left
298     \itf@cellrighttrue      % consequently the token to prepare is right !
299     \toks\count@\expandafter{% augment the previous (left) token
300       \expandafter\itf@celllalign
301       \the\toks\count@

```

```

302         \itf@cellleft}%
303     \or                                     % (case 1)
304         \ifitf@cellright      % previous token is right
305         \itf@cellrightfalse% consequently the token to prepare is not right !
306         \toks\count@=\expandafter{% augment the previous (right) token
307             \the\toks\count@
308             \itf@cellright
309             \itf@cellralign}%
310     \fi
311 \else % special column: do not change the token
312     \itf@cellrightfalse % the token to prepare cannot be right !
313 \fi
314 \advance \count@ \@ne \toks\count@{% \prepnext@tok original def
315}% \itf@prepnext@tok
316 \AfterPackage*{array}{%
317     \global\let\array@prepnext@tok\prepnext@tok % frozen original definition
318     \global\let\prepnext@tok\itf@prepnext@tok
319}% \AfterPackage*{array}

```

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

`\itf@celllalign`

`\itf@cellright`

`\itf@cellralign`

`\itf@cellfont`

```

320 \let\itf@cellleft\@empty
321 \let\itf@cellright\@empty
322 \def\itf@celllalign{\itf@cellleft}% the row font spec. applies to pre-column material
323 \let\itf@cellralign\@empty
324 \let\itf@cellfont\@empty

```

`\itf@cell@l` Setup macros to modify the alignment. The skips inserted to make the standard alignment specified in the

`\itf@cell@c` tabular preamble are not the same with standard `array` tabulars and `colortbl` tabulars, hence the switch

`\itf@cell@r` `\ifitf@colortbl`.

`\itf@cell@j`

```

325 \let\ifitf@colortbl\iffalse
326 \AfterPackage*{colortbl}{\let\ifitf@colortbl\iftrue}
327 \def\itf@cell@l{% force alignment to left
328     \gdef\itf@celllalign{\itf@removehfil
329         \raggedright\arraybackslash
330         \itf@cellleft}%
331     \gdef\itf@cellralign{\itf@flush1\itf@ignorehfil}%
332     \toks@\expandafter{\itf@cellleft\raggedright\arraybackslash}% local (noalign group)
333     \xdef\itf@cellleft{\the\toks@}%
334}% \itf@cell@l
335 \def\itf@cell@c{% force alignment to center
336     \gdef\itf@celllalign{\itf@removehfil
337         \centering\arraybackslash
338         \itf@flush{.5}\itf@cellleft}%
339     \gdef\itf@cellralign{\itf@flush{.5}\itf@ignorehfil}%
340     \toks@\expandafter{\itf@cellleft\centering\arraybackslash}% local (noalign group)
341     \xdef\itf@cellleft{\the\toks@}%
342}% \itf@cell@c
343 \def\itf@cell@r{% force alignment to right
344     \gdef\itf@celllalign{\itf@removehfil
345         \raggedleft\arraybackslash
346         \itf@flush1\itf@cellleft}%
347     \gdef\itf@cellralign{\itf@ignorehfil}%
348     \toks@\expandafter{\itf@cellleft\raggedleft\arraybackslash}% local (noalign group)
349     \xdef\itf@cellleft{\the\toks@}%
350}% \itf@cell@r
351 \def\itf@cell@j{% force justification (for p, m, b columns)
352     \gdef\itf@celllalign{\itf@justify\itf@cellleft}%
353     \global\let\itf@cellralign\@empty
354     \toks@\expandafter{\itf@cellleft\itf@justify}% local (noalign group)
355     \xdef\itf@cellleft{\the\toks@}%

```

```

356}% \itf@cell@j
357\def\itf@justify{%
358  \leftskip\z@skip \@rightskip\leftskip \rightskip\@rightskip
359  \parfillskip\@flushglue
360}% \itf@justify
361\AfterPackage*{ragged2e}{%
362\def\itf@cell@L{% force alignment to left (ragged2e)
363  \gdef\itf@celllalign{\itf@removehfil
364    \RaggedRight\arraybackslash
365    \itf@cellleft}%
366  \gdef\itf@celllralign{\itf@flush1\itf@ignorehfil}%
367  \toks@\expandafter{\itf@cellleft\RaggedRight\arraybackslash}%
368  \xdef\itf@cellleft{\the\toks@}%
369}% \itf@cell@L
370\def\itf@cell@C{% force alignment to center (ragged2e)
371  \gdef\itf@celllalign{\itf@removehfil
372    \Centering\arraybackslash
373    \itf@flush{.5}\itf@cellleft}%
374  \gdef\itf@celllralign{\itf@flush{.5}\itf@ignorehfil}%
375  \toks@\expandafter{\itf@cellleft\Centering\arraybackslash}%
376  \xdef\itf@cellleft{\the\toks@}%
377}% \itf@cell@C
378\def\itf@cell@R{% force alignment to right (ragged2e)
379  \gdef\itf@celllalign{\itf@removehfil
380    \RaggedLeft\arraybackslash
381    \itf@flush1\itf@cellleft}%
382  \gdef\itf@celllralign{\itf@ignorehfil}%
383  \toks@\expandafter{\itf@cellleft\RaggedLeft\arraybackslash}%
384  \xdef\itf@cellleft{\the\toks@}%
385}% \itf@cell@R
386\def\itf@cell@J{% force justification (ragged2e)
387  \gdef\itf@celllalign{\justifying\arraybackslash\itf@cellleft}%
388  \global\let\itf@celllralign\@empty
389  \toks@\expandafter{\itf@cellleft\justifying\arraybackslash}%
390  \xdef\itf@cellleft{\the\toks@}%
391}% \itf@cell@J
392}% AfterPackage ragged2e
393\def\itf@flush#1{%
394  \ifitf@colortbl % colortbl uses \hfill rather than \hfil
395    \hskip \ifnum\currentgrouptype>13 \stretch{#1}%
396    \else\ifdim#1pt<1pt \itf@cellskip
397    \else \stretch{#1}
398    \fi\fi \relax
399  \else % array.sty
400    \ifnum \currentgrouptype>13\relax
401      \hfil \hskip1sp
402    \fi
403    \fi
404}% \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.

```

405\newskip\itf@cellskip
406\let\itf@hfil\hfil
407\let\itf@hfill\hfill
408\let\itf@hskip\hskip
409\def\itf@removehfil{%
410  \ifitf@colortbl
411    \unkern \itf@cellskip = \lastskip

```

```

412 \ifnum\gluestretchorder\itf@cellskip = \tw@ \hskip-\itf@cellskip
413 \else \itf@cellskip = \z@skip
414 \fi
415 \else
416 \ifdim\lastskip=1sp\unskip\fi
417 \ifnum\gluestretchorder\lastskip = \@ne
418 \hfilneg % \hfilneg for array.sty but not for colortbl...
419 \fi
420 \fi
421}% \itf@removehfil
422\def\itf@nohfil{% \hfil -> do nothing + restore original \hfil
423 \def\hfil{\let\hfil\itf@hfil}% local to (alignment template) group
424}% \itf@nohfil
425\AfterPackage*{colortbl}{%
426 \def\itf@nohfil{%
427 \def\hfil{\let\hfil\itf@hfil}% local to (alignment template) group
428 \def\hfill{\let\hfill\itf@hfill}% (colortbl uses \hfill) pfff...
429 \def\hskip##1\relax{\let\hskip\itf@hskip}% local
430}% AfterPackage colortbl
431\def\itf@ignorehfil{%
432 \aftergroup\itf@nohfil
433}% \itf@ignorehfil

434</LaTeX>

```

21.4 † Interface TikZ (pgfmanual)

```

435<*tikz>
436\ProvidesPackage{interfaces-tikz}
437 [2010/09/21 v1.0 load of tikz libraries easily (FC)]
438\RequirePackage{interfaces-base}
439\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.

```

440%% Some catcode checking
441\AfterPackage*{tikz}{%
442 \AtEndPreamble{%
443 \tikzset{every picture/.add style={%
444 \utils/exec={\@makeother\!\@makeother\:\}}}%
445 }%
446}%

```

21.4.2 List of pgf/TikZ packages and libraries

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

```

447\newcommand*\alltikzpackages{%
448 pgf,pgfmath,pgfcalendar,pgfplots,pgfpages,tikz%
449}% \alltikzpackages

```

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

```

450\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.

```

451 \newcommand*\alltikzlibraries{%
452   arrows,%
453   chains,%
454   er,%
455   scopes,%
456   snakes,%
457   shapes,%
458   shapes.misc,%
459   shapes.geometric,%
460   shapes.arrows,%
461   shapes.multipart,%
462   shapes.callouts,%
463   shapes.symbols,%
464   shapes.gates.logic.US,%
465   shapes.gates.logic.IEC,%
466   decorations,%
467   decorations.pathmorphing,%
468   decorations.pathreplacing,%
469   decorations.shapes,%
470   decorations.text,%
471   decorations.markings,%
472   decorations.fractals,%
473   decorations.footprints,%
474   positioning,trees,%
475   automata,%
476   patterns,%
477   fadings,%
478   shadows,%
479   through,%
480   backgrounds,%
481   petri,%
482   fit,%
483   mindmap,%
484   folding,%
485   plohandlers,%
486   plotmarks,%
487   calc,%
488   matrix%
489 %%   topath,placements,%
490}% \alltikzlibraries

```

21.4.3 pgfkeys definitions for `\usetikz`

```

491 \pgfqkeys{/itf/tikz}{%
492   basic/.code={%
493     \usetikzlibrary{calc,%
494                     arrows,%           for arrows
495                     backgrounds,%     for framed
496                     scopes%          for scopes and shortcut : { [scope] ... }
497   },%
498   basic/.value forbidden,% \alltikzpackages
499   base/.alias=basic,%
500   all/.code={%
501     \expandafter\RequirePackage\expandafter{\alltikzpackages}%
502     \expandafter\usepgfmodule\expandafter{\allpgfmodules}% see pgfcorescopes.code.tex
503     \expandafter\usetikzlibrary\expandafter{\alltikzlibraries}% from: tikz.code.tex

```



```

505   },%
506   .unknown/.code=\itf@usetikzcode\pgfkeyscurrentname
507}% \pgfkeys for \usetikz

\usetikz

508 \newrobustcmd*\usetikz[1]{%
509   \RequirePackage{tikz}%
510   \ifblank{#1}
511     {\pgfqkeys{/itf/tikz}{all}}
512     {\pgfqkeys{/itf/tikz}{#1}}%
513   \itf@pgfkeys% overwritten by \input primitive otherwise
514}% \usetikz

```

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

```

515 \newrobustcmd*\itf@usetikzcode[1]{\begingroup
516   \edef\x{%
517     \noexpand\in@{, #1,}{, \alltikzpackages,}%
518     \noexpand\ifin@ \noexpand\usepackage{#1}%
519     \noexpand\else
520       \noexpand\in@{, #1,}{, \allpgfmodules,}%
521       \noexpand\ifin@ \noexpand\usepgfmodule{#1}%
522       \noexpand\else \noexpand\usetikzlibrary{#1}% \usetikzlibrary by default
523       \noexpand\fi
524     \noexpand\fi
525   }\expandafter\endgroup\x
526}% \itf@usetikzcode

527 \</tikz>

```

21.5 † Interface ⇄ titlesec : \sectionformat

```

528 \*titlesec
529 \ProvidesPackage{interfaces-titlesec}
530 [2010/10/20 v1.0 pgfkeys macros for titlesec (FC)]
531 \RequirePackage{interfaces-base}
532 \RequirePackage{auxhook}% for fix for \ttl@append (odd/even pages, not implemented in in
533 \itf@Check@Package{titlesec}
534 \itf@pgfkeys

```

21.5.1 titlesec modifications

```

535 %%   t i t l e s e c   " p a t c h e s "
536 \robustify\titleclass \robustify\titleformat \robustify\titlespacing
537 \AddLineBeginMainAux{%
538   \string\providecommand\string\ttl@append[2]{}%
539 }
540 \BeforeFile*{\ttlkeys.def}{%
541   \catcode32 9\relax % ignore spaces
542   \count@\endlinechar\endlinechar\m@ne % and line feeds
543 }
544 \AfterFile*{\ttlkeys.def}{% restore spaces and line feeds
545   \catcode32 10\relax
546   \endlinechar\count@
547 }

```

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

```

548 %%-----
549 %% runinifempty => like hang but no paragraph break (like runin) if title is empty
550 \def\ttl@runinifempty{%

```

```

551 \csname ttlh@ifx\ttl@savetitle@empty runin\else hang\fi\endcsname
552}% \ttl@runinifempty

```

21.5.2 `\sectionformat`

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

```

553 \newrobustcmd*\sectionformat[1]{\@ifstar
554   {\itf@sectionformat{#1}{/*}}
555   {\itf@sectionformat{#1}{}}}%
556}% \sectionformat
557 \def\itf@sectionformat#1#2{\@testopt
558   {\expandafter\expandafter\expandafter%
559     \itf@sectionformat\expandafter\@gobble\string#1\@nil{#2}}}%
560}% \sectionformat
561 \def\itf@sectionformat#1\@nil#2[#3]#4{\itf@pgfkeys
562   \ifblank{#3}
563     {\pgfkeys{/itf/titlesec/#1#2}
564       {\itf/titlesec/@setup={#1}{#2},#4,/itf/titlesec/@update={#1}{#2}}}
565     {\pgfkeys{/itf/titlesec/#1#2}
566       {\itf/titlesec/@setup={#1}{#2},shape={#3},#4,/itf/titlesec/@update={#1}{#2}}}
567}% \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
568%% private keys @setup and @update
569 \pgfkeys{/itf/titlesec}{%
570   @setup/.code 2 args=\itf@ttl@setup{#1}{#2},%
571   @update/.code 2 args=\expandafter\itf@ttl@update
572     \csname#1\endcsname{/itf/titlesec/#1}{#2}%
573 }
574 \newrobustcmd*\itf@ttl@setup[2]{%
575   \pgfkeysifdefined{/itf/titlesec/#1#2/shape}{%
576     {\itf@ttl@newsection{#1}{#2}}}%
577   \ifcsundef{ttls@#1#2}{% no previous spacing settings
578     {\expandafter\expandafter\expandafter%
579       \itf@ttl@getspacing\csname ttls@#1#2\endcsname{/itf/titlesec/#1#2}}}%
580   \ifcsundef{ttlf@#1#2}{% no previous format settings
581     {\expandafter\expandafter\expandafter%
582       \itf@ttl@getformat\csname ttlf@#1#2\endcsname{/itf/titlesec/#1#2}}}%
583}% \itf@ttl@setup
584 \def\itf@ttl@setkey#1#2{\begingroup
585   \def\x{\pgfkeysvalueof{#1}}\def\y{#2}%
586   \expandafter\endgroup\ifx\x\y\else
587     \pgfkeyssetvalue{#1}{#2}\fi
588}% \itf@ttl@setkey
589 \def\itf@ttl@getspacing#1#2#3#4#5#6{%
590   \itf@ttl@setkey{#6/left}{#1}%
591   \itf@ttl@setkey{#6/right}{#2}%
592   \itf@ttl@setkey{#6/top}{#3}%
593   \itf@ttl@setkey{#6/bottom}{#4}%
594   \itf@ttl@setkey{#6/indent first}{#5}%
595}% \itf@ttl@getspacing
596 \def\itf@ttl@getformat#1#2{%
597   \csname ltx@ifx\pgfkeysvalueof{#1}first\else second\fi oftwo\endcsname
598     {\itf@ttl@get@format{}}
599     {\itf@ttl@get@format{#1}{#2}}}%
600}% \itf@ttl@getformat
601 \def\itf@ttl@get@format#1#2#3#4#5#6#7{%
602   \if\relax\detokenize{#1}\relax\else

```

```

603 \pgfkeyssetvalue{#7/shape}{\expandafter\itf@strip@at\string#1}%
604 \fi
605 \itf@ttl@setkey{#7/font}{#2}%
606 \itf@ttl@setkey{#7/label}{#3}%
607 \itf@ttl@setkey{#7/sep}{#4}%
608 \itf@ttl@setkey{#7/before}{#5}%
609 \itf@ttl@setkey{#7/after}{#6}%
610 }% \itf@ttl@get@format

```

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

```

611 \newrobustcmd*\itf@ttl@update[3]{%
612   \ifblank{#3}
613     {\ttl@getkeys{#1}{titlesec}}
614     {\ttl@getkeys{name=#1,numberless}{titlesec}}%
615   \pgfkeysgetvalue{#2#3/shape}\itf@tp
616   \ifcsundef{ttlh@\itf@tp}
617     {\begingroup \makeatletter
618       \InputIfFileExists{\itf@tp.tss}{\}%
619       \PackageError{titlesec}{Unknown shape}%
620       {Shapes are defined in files with extension tss\MessageBreak
621        Either you have misspelled the shape\MessageBreak
622        or there is no a \itf@tp.tss file}}%
623     \endgroup}
624   {\}%
625   \ifttl@explicit\@temptokena{\csdef{ttl@ttl@a}##1}%
626   \else\@temptokena{\csdef{ttl@ttl@a}}%
627   \fi
628   \the\@temptokena{%
629     \pgfkeysvalueof{#2#3/shape@cs}%
630     {\pgfkeysvalueof{#2#3/font}}%
631     {\pgfkeysvalueof{#2#3/label}}%
632     {\pgfkeysvalueof{#2#3/sep}}%
633     {\pgfkeysvalueof{#2#3/before}}%
634     {\pgfkeysvalueof{#2#3/after}}}%
635   \csname ttl@compat\ttl@a\endcsname
636   \ttl@spacing@ii
637     {\pgfkeysvalueof{#2#3/indent first}}%
638     {\pgfkeysvalueof{#2#3/left}}%
639     {\pgfkeysvalueof{#2#3/top}}%
640     {\pgfkeysvalueof{#2#3/bottom}}%
641     [\pgfkeysvalueof{#2#3/right}}%
642 }% \itf@ttl@update

```

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

- change the class of an existing sectionning command;

or:

- create a new sectionning command.

```

643 %%-----
644 %%  t i t l e c l a s s
645 \newrobustcmd*\itf@ttl@titleclass{\@ifstar
646   {\itf@ttl@titleclass@i*}
647   {\itf@ttl@titleclass@i}}%
648 }% \itf@ttl@titleclass
649 \def\itf@ttl@titleclass@i#1#2#3[#4]#5\@nil{\ifblank{#4}
650   {%
651     \expandafter\titleclass\csname#2\endcsname{#3}%
652   }
653   {%

```

```

654 \expandafter\titleclass\csname#2\endcsname{#3}[{#4}]% #4 = sup-level
655 \if\relax#1\relax % not starred
656 \expandafter\expandafter\expandafter%
657 \itf@ttl@titleclass@ii\expandafter\@gobble\string#4\relax{#2}% subsection\re
658 \fi
659 }%
660 \expandafter\providecommand\csname l@#2\endcsname[2]{}% no entry in toc
661 }% \itf@ttl@titleclass@i
662 \def\itf@ttl@titleclass@ii#1\relax#2{%
663 \ifcsundef{c@#2}
664 {\newcounter{#2}[{#1}]}% creates counter for #1
665 {\@addtoreset{#2}{#1}}%
666 \csxdef{the#2}{%
667 \expandafter\noexpand\csname the#1\endcsname
668 .\noexpand\@arabic\expandafter\noexpand\csname c@#2\endcsname
669 }% \the#2
670 \ifcsundef{l@#1}{}%
671 \expandafter\providecommand\csname l@#2\endcsname[2]{%
672 \csname l@#1\endcsname{##1}{##2}%
673 }% \l@#2 = \l@newsection is (by default) the same as \l@sup-level
674 }%
675 }% \itf@ttl@titleclass@ii

```

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

- 1) create the new sectioning 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.

```

676 %-----
677 % C r e a t i n g   n e w   s e c t i o n s
678 \def\itf@ttl@createsection#1{% sectioning command undefined: temporarily defined
679 \expandafter\titleclass\csname#1\endcsname{straight}[*]% in straight class
680 \itf@AfterPackage{hypbmsec}{%
681 \AtBeginDocument{%
682 \csletcs{hbs@#1}{#1}%
683 \csxdef{#1}{\noexpand\hbs@do\expandafter\noexpand\csname hbs@#1\endcsname}%
684 }%
685 }%
686 \itf@AfterPackage{interfaces-tocloft}{%
687 \itf@cft@newsection{#1}{toc}%
688 }%
689 \itf@AfterPackage{interfaces-hypbmsec}{%
690 \cslet{last#1title}\@empty
691 \cslet{last#1bookmark}\@empty
692 \cslet{last#1toc}\@empty
693 }%
694 }% \itf@ttl@createsection

```

21.5.3 Keys for `\sectionformat`

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

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

```

695 %-----
696 %%   D e f i n i n g   k e y s   f o r   e a c h   s e c t i o n
697 \def\itf@ttl@newsection#1#2{% #1=section, #2=/* or empty
698   \ifcsundef{#1}% sectioning command undefined: temporarily defined in straight class
699     {\itf@ttl@createsection{#1}}
700     {}%
701   \ifblank{#2}{%
702     \pgfqkeys{/itf/titlesec/#1}{% these keys do not exist for starred/numberless section
703       class/.code=\itf@ttl@titleclass{#1}##1[]\@nil,% with \@addtoreset
704       class/.value required,% if sup-level is given
705       class*/.code=\itf@ttl@titleclass*{#1}##1[]\@nil,% without \@addtoreset
706       class*/.value required,% even if sup-level is given
707       shape/.initial=,% initial shapes set by titlesec.sty for not numbered sections.
708       the#1/.store in cs=the#1,%
709       the#1/.default=\pgfkeysvalueof{/itf/titlesec/#1/label},%
710     }
711     {% starred (numberless) unless titlesec, we do not put the default shape to hang,
712     \pgfkeysifdefined{/itf/titlesec/#1/shape}% but to the shape of the not starred
713     {\pgfkeysgetvalue{/itf/titlesec/#1/shape}\itf@tp}% sectioning command
714     {\def\itf@tp{hang}}}%
715     \pgfkeyslet{/itf/titlesec/#1#2/shape}\itf@tp%
716   }%
717   \pgfqkeys{/itf/titlesec/#1#2}{%
718     shape/.value required,%
719     shape@cs/.initial=\csname ttlh@pgfkeysvalueof{/itf/titlesec/#1#2/shape}\endcsname,%
720     shape@cs/.is private=interface-titlesec,% constant
721     font/.store as code,%
722     format/.alias*=font,%
723     label/.store as code,%
724     sep/.estore as glue,%
725     labelsep/.alias*=sep,%
726     before/.store as code,%
727     after/.store as code,%
728     indent first/.true=\pgfkeyssetvalue{/itf/titlesec/#1#2/indent first}{\@ne},%
729     indent first/.false=\pgfkeyssetvalue{/itf/titlesec/#1#2/indent first}{\z@},%
730     identfirst/.alias=indent first,%
731     indent first=true,%
732     left/.estore as glue,%
733     top/.estore as glue,%
734     bottom/.estore as glue,%
735     right/.estore as glue,%
736     break/.code=\ifblank{##1}
737       {\cslet{#1break}\@empty}
738       {\protected\csdef{#1break}{##1}},%
739     break*/.code=\protected\apptocs{#1break}{##1},%
740     break+/.alias=break*,%
741     mark/.code=\ifblank{##1}
742       {\cslet{#1mark}\@gobble}
743       {\protected\long\csdef{#1mark}####1{##1}},%
744     mark/.default=,%
745   }% \pgfkeys
746   \ifblank{#2}{% no default marks for starred (numberless) \section*
747   \itf@AfterPackage{interfaces-marks}{% requires interfaces-marks
748     \pgfkeys{/itf/titlesec/#1/mark/.default=\marksthe{#1}{##1}}%
749   }% AfterPackage interfaces-mark
750   \itf@AfterPackage{interfaces-bookmark}{% requires interfaces-bookmark
751   \pgfqkeys{/itf/titlesec/#1}{% does not exist for starred sections
752     bookmark/.is multiple choice,%

```

```

753 bookmark/level/.estore in cs=toclevel@#1,% depth of the bookmark
754     bookmark/level/.default=,%
755     bookmark/level/.blank=\ifcvoid{toclevel@#1}{99}{\csuse{toclevel@#1}},%
756     bookmark/toclevel/.alias=bookmark/level,%
757 bookmark/bold/.code=\ifblank{##1}
758     {\pgfkeyslet{/itf/titlesec/#1/bookmark/bold}\@empty}
759     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/bold}{bold=##1},}%
760     bookmark/bold/.default=true,%
761 bookmark/bold*/.code=\ifblank{##1}
762     {\pgfkeyslet{/itf/titlesec/#1/bookmark/bold*}\@empty}
763     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/bold*}{bold*=##1},}%
764     bookmark/bold*/.default=true,%
765 bookmark/italic/.code=\ifblank{##1}
766     {\pgfkeyslet{/itf/titlesec/#1/bookmark/italic}\@empty}
767     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/italic}{italic=##1},}%
768     bookmark/italic/.default=true,%
769 bookmark/italic*/.code=\ifblank{##1}
770     {\pgfkeyslet{/itf/titlesec/#1/bookmark/italic*}\@empty}
771     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/italic}{italic*=##1},}%
772     bookmark/italic*/.default=true,%
773 bookmark/color/.code=\ifblank{##1}
774     {\pgfkeyslet{/itf/titlesec/#1/bookmark/color}\@empty}
775     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/color}{color=##1},}%
776     bookmark/color/.default=,%
777 bookmark/color*/.code=\ifblank{##1}
778     {\pgfkeyslet{/itf/titlesec/#1/bookmark/color*}\@empty}
779     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/color*}{color*=##1},}%
780     bookmark/color*/.default=,%
781 bookmark/on/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on}{disable=false},
782     bookmark/on/.value forbidden,%
783 bookmark/off/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on}{disable=true},
784     bookmark/off/.value forbidden,%
785     bookmark/none/.alias=bookmark/off,%
786 bookmark/style/.code=\ifblank{##1}
787     {\pgfkeyslet{/itf/titlesec/#1/bookmark/style}\@empty}
788     {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/style}{style=##1},}%
789     bookmark/style/.default=,%
790 bookmark/.unknown/.code=\pgfkeysEsetvalue{/itf/titlesec/#1/bookmark/other}{\pgfkey
791 bookmark/bold/.initial=,%
792 bookmark/bold*/.initial=,%
793 bookmark/italic/.initial=,%
794 bookmark/italic*/.initial=,%
795 bookmark/color/.initial=,%
796 bookmark/color*/.initial=,%
797 bookmark/on/.initial=,%
798 bookmark/style/.initial=,%
799 }%% \AfterPackage*{interfaces-bookmark}
800 }% not starred (numberless) \section
801 {}% starred/numberless \section*: do not create the keys
802 }% \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 sectioning command.

```

803 %% bookmarks styles for sectionning commands:
804 \@makeother\#
805 \itf@AfterPackage{interfaces-bookmark}{%
806     \itf@patch\itf@preapptocmd\addcontentsline           % #2=section name in
807         {\begingroup\itf@ttl@BKM{/itf/titlesec/#2/bookmark}}% \addcontentsline
808         {\endgroup}
809         {\PackageInfo{interfaces-titlesec}
810          {Command \string\addcontentsline\space successfully patched!}}

```



```

811      {\PackageWarning{interfaces-titlesec}
812       {Patch of \string\addcontentsline\space failed
813        \MessageBreak bookmark-style specs for sections
814        \MessageBreak will be unavailable}}}%
815 }% AfterPackage interfaces-bookmark
816 \catcode'\#=6
817 \def\itf@ttl@BKM#1{%
818   \pgfkeysifdefined{#1/.@cmd}
819   {%
820     \edef\itf@tp{% do nothing if the keys are not set (shortcut)
821       \pgfkeysvalueof{#1/bold}%
822       \pgfkeysvalueof{#1/bold*}%
823       \pgfkeysvalueof{#1/italic}%
824       \pgfkeysvalueof{#1/italic*}%
825       \pgfkeysvalueof{#1/color}%
826       \pgfkeysvalueof{#1/color*}%
827       \pgfkeysvalueof{#1/on}%
828       \pgfkeysvalueof{#1/style}}}%
829   \ifx\itf@tp\@empty
830   \else
831     \global\pgfkeyslet{#1/bold*}\@empty
832     \global\pgfkeyslet{#1/italic*}\@empty
833     \global\pgfkeyslet{#1/color*}\@empty
834     \expandafter\bookmarksetup\expandafter{\itf@tp}%
835     \pgfkeysifdefined{/itf/bookmark/*}%
836     {% \bookmarksetup{color*=} overwrites \sectionformat...{bookmark/color*=}
837       \pgfkeysgetvalue{/itf/bookmark/*}\itf@tp
838       \expandafter\bookmarksetup\expandafter{\itf@tp}%
839     }
840     {}%
841   \fi}
842   {}%
843 }% \itf@ttl@BKM

```

21.5.4 \bottomtitles

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

```

844 %%-----
845 %%   b o t t o m t i t l e s
846 \newrobustcmd*\bottomtitles{\@ifnextchar\bgroup
847   \itf@setbottomtitlespace
848   {\afterassignment\itf@set@bottomtitle@space\itf@sk}%
849 }% \bottomtitles
850 \def\itf@setbottomtitlespace#1{%
851   \@defaultunits\itf@sk#1\textheight\relax\@nnil
852   \itf@set@bottomtitle@space
853 }% \itf@setbottomtitlespace
854 \def\itf@set@bottomtitle@space{\edef\bottomtitlespace{\the\itf@sk}}
855 </titlesec>

```

21.6 † Interface `hyperref`

```

856 <*hyperref>
857 \ProvidesPackage{interfaces-hyperref}
858   [2010/08/08 v1.0 provides raised hyperdef (FC)]
859 \RequirePackage{interfaces-base}
860 \itf@Check@Package{hyperref}

```

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

```

861 \newrobustcmd\raisedhyperdef[4][\begingroup
862   \ifblank{#1}
863     {}
864     {%
865       \expandafter\def\expandafter\HyperRaiseLinkHook\expandafter{%
866         \expandafter\def\expandafter\HyperRaiseLinkHook
867           \expandafter{\HyperRaiseLinkHook}% restore current defin
868         \HyperRaiseLinkHook
869         \advance\noexpand\HyperRaiseLinkLength by\dimexpr#1\relax\relax}%
870     }%
871   \ifx \\\#2\\\def \Hy@AnchorName {#3}\else
872     \def \Hy@AnchorName {#2.#3}%
873   \fi
874   \ltx@leavevmode
875   \Hy@raisedlink{\hyper@anchorstart{\Hy@AnchorName}\hyper@anchorend}%
876   \endgroup
877   #4%
878 }% \raisedhyperdef

879 </hyperref>

```

21.7 † Interface `bookmark`

```

880 <*bookmark>
881 \ProvidesPackage{interfaces-bookmark}
882   [2010/09/08 v1.0 more control on bookmarks (FC)]
883 \RequirePackage{interfaces-base}
884 \RequirePackage{interfaces-hyperref}
885 \RequirePackage{refcount}
886 \itf@Check@Package{bookmark}
887 \itf@pgfkeys

```

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

```

888 \expandafter\itf@patch\expandafter\patchcmd\csname\string\bookmark\endcsname
889   {\immediate\write\@mainaux}{\itf@BKM@write}
890   {\PackageInfo{interfaces-bookmark}
891     {Command \expandafter\string\csname\string\bookmark\endcsname\space
892       successfully patched!}}
893   {\PackageWarning{interfaces-bookmark}
894     {Command \expandafter\string\csname\string\bookmark\endcsname\space
895       could not be patched\MessageBreak
896       Loading of interfaces-bookmark is canceled}%
897     \endinput}

```

21.7.1 star keys* definitions

```

898 \global\let\itf@BKM@enable\@ne % 0 = disable*, 1=enable, \@M = disable
899 \pgfqkeys{/itf/bookmark}{%
900   disable/.true=\global\let\itf@BKM@enable\@M,%
901   disable/.false=\global\let\itf@BKM@enable\@ne,%
902   disable*/.true=\global\let\itf@BKM@enable\z@,%
903   disable*/.false=\global\let\itf@BKM@enable\@ne,%
904   atend/.true=\ifnum\itf@BKM@enable>0\edef\itf@BKM@enable{\number-\itf@BKM@enable}\fi,%
905   atend/.false=\ifnum\itf@BKM@enable<0\edef\itf@BKM@enable{\number-\itf@BKM@enable}\fi,%
906   bold*/.true=\pgfqkeys{/itf/bookmark}{@save={bold=true}},%
907   bold*/.false=\pgfqkeys{/itf/bookmark}{@save={bold=false}},%
908   bold*/@blank/.code=\pgfqkeys{/itf/bookmark/@restore}{bold},%
909   bold*/.blank=@blank,%
910   italic*/.true=\pgfqkeys{/itf/bookmark}{@save={italic=true}},%
911   italic*/.false=\pgfqkeys{/itf/bookmark}{@save={italic=false}},%
912   italic*/@blank/.code=\pgfqkeys{/itf/bookmark/@restore}{italic},%
913   italic*/.blank=@blank,%
914   color*/.code={\ifblank{#1}
915     {\pgfqkeys{/itf/bookmark/@restore}{color}}
916     {\pgfqkeys{/itf/bookmark}{@save={color={#1}}}}}%
917 },%
918 @save/.code={%
919   \pgfkeysifdefined{/itf/bookmark/@save/color}{%
920     {\pgfkeyslet{/itf/bookmark/@save/color}\BKM@color}%
921   \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{%
922     {\pgfkeyslet{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}%
923   \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}{%
924     {\pgfkeysEsetvalue{/itf/bookmark/@save/grouplevel}{\the\currentgrouplevel}}%
925   \pgfkeysaddvalue{/itf/bookmark/*}{\{, #1}%
926   \bookmarksetup{#1}%
927 },%
928 @restore/color/.code=%
929   \pgfkeysifdefined{/itf/bookmark/@save/color}{%
930     {\pgfkeysgetvalue{/itf/bookmark/@save/color}\BKM@color}%
931     {}},%
932 @restore/FontStyle/.code=%
933   \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{%
934     {\pgfkeysgetvalue{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}%
935     {}},%
936 @restore/.code={%
937 %% \pgfqkeys{/itf/bookmark/@restore}{color,FontStyle}% repeating the following lines
938 \pgfkeysifdefined{/itf/bookmark/@save/color}% is quicker than setting a /.style ...
939   {\pgfkeysgetvalue{/itf/bookmark/@save/color}\BKM@color}%
940   {}%
941 \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{%
942   {\pgfkeysgetvalue{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}%
943   {}%
944 \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}{%
945   {% shortcut: \bookmarksetup{bold*} called inside main group (level 0)
946   \ifnum\pgfkeysvalueof{/itf/bookmark/@save/grouplevel}=\z@
947     \global\pgfkeyslet{/itf/bookmark/@save/FontStyle}\@undefined
948     \global\pgfkeyslet{/itf/bookmark/@save/color}\@undefined
949     \global\pgfkeyslet{/itf/bookmark/@save/grouplevel}\@undefined
950     \global\pgfkeyslet{/itf/bookmark/*}\@undefined
951     \global\let\BKM@color\BKM@color
952     \global\let\BS@BKM@FontStyle\BS@BKM@FontStyle
953   \else % restore after each group as needed
954     \pgfkeyslet{/itf/bookmark/@save/FontStyle}\@undefined
955     \pgfkeyslet{/itf/bookmark/@save/color}\@undefined
956     \pgfkeyslet{/itf/bookmark/*}\@undefined
957     \ifnum\currentgrouplevel>\z@
958       \aftergroup\itf@BKM@restore

```

```

959         \else
960         \pgfkeyslet{/itf/bookmark/@save/grouplevel}\@undefined
961     \fi\fi
962     }
963     {}%
964 },%
965 @restore/bold/.code={% bold is "hidden" in \BS@BKM@FontStyle
966     \beginngroup
967     \edef\itf@tp{\bookmarkget{bold}}%
968     \pgfqkeys{/itf/bookmark/@restore}{FontStyle}%
969     \count@=\bookmarkget{bold}%
970     \edef\itf@tp{\endgroup
971         \ifnum\itf@tp=\count@
972         \else \noexpand\bookmarksetup{%
973             bold\ifcase\count@ =false\fi}\fi
974     }\itf@tp},%
975 @restore/italic/.code={% italic is "hidden" in \BS@BKM@FontStyle
976     \beginngroup
977     \edef\itf@tp{\bookmarkget{italic}}%
978     \pgfqkeys{/itf/bookmark/@restore}{FontStyle}%
979     \count@=\bookmarkget{italic}%
980     \edef\itf@tp{\endgroup
981         \ifnum\itf@tp=\count@
982         \else \noexpand\bookmarksetup{%
983             italic\ifcase\count@ =false\fi}\fi
984     }\itf@tp},%
985 }
986 \def\itf@BKM@restore{%
987     \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}
988     {\pgfkeysalso{/itf/bookmark/@restore}}
989     {}%
990 }% \itf@BKM@restore
991 %% Plug the BKM keys in the interfaces pgfkeys (not efficient but quickly written...)
992 \define@key{BKM}{disable}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/disable={#1}}}
993 \define@key{BKM}{disable*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/disable*={#1}}}
994 \define@key{BKM}{bold*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/bold*={#1}}}
995 \define@key{BKM}{italic*}[true]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/italic*={#1}}}
996 \define@key{BKM}{color*}[]{\itf@pgfkeys\pgfkeysalso{/itf/bookmark/color*={#1}}}
997 \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.

```

998 \def\itf@BKM@write#1{%
999     \aftergroup\itf@BKM@restore
1000     \ifcase\itf@BKM@enable
1001         \global\let\itf@BKM@enable\@ne
1002     \or
1003         \immediate\write\@mainaux{#1}%
1004     \else
1005         \ifcase -\itf@BKM@enable % (case -0 = 0 tested before)
1006         \or % (case -1 = enable but At End)
1007             \let\itf@BKM@enable\@ne
1008             \edef\itf@tp{\immediate\write\@mainaux{#1}}%
1009             \expandafter\BookmarkAtEnd\expandafter{\itf@tp}%
1010         \fi
1011     \fi
1012 }% \itf@BKM@write

```

21.7.2 ref and pageref keys

`\itf@BKM@pageref`

```

1013%% pageref= key for \bookmark (using refcount : \getpagerefnnumber)
1014\def\itf@BKM@pageref[#1]{%
1015  \expandafter\itf@BKM@@pageref
1016    \csname r@#1\endcsname{#1}%
1017}% \itf@BKM@pageref
1018\def\itf@BKM@@pageref#1#2{%
1019  \ifx#1\relax
1020    \protect\G@refundefinedtrue
1021    \@latex@warning{%
1022      Hyper reference ‘#2’ on page \thepage \space undefined
1023    }%
1024  }%
1025  \fi
1026  \edef\BKM@page{\getrefbykeydefault{#2}{page}{1}}%
1027}% \itf@BKM@pageref
1028\define@key{BKM}{pageref}{%
1029  \Hy@babelnormalise\itf@BKM@pageref[{#1}]}

```

`\itf@BKM@ref`

```

1030%% ref= key for \bookmark
1031\def\itf@BKM@ref[#1]{%
1032  \expandafter\itf@BKM@@ref
1033    \csname r@#1\endcsname{#1}%
1034}% \itf@BKM@ref
1035\def\itf@BKM@@ref#1#2{%
1036  \ifx#1\relax
1037    \protect\G@refundefinedtrue
1038    \@latex@warning{%
1039      Hyper reference ‘#2’ on page \thepage \space undefined
1040    }%
1041  }%
1042  \def\BKM@dest{??}%
1043  \else
1044    \edef\BKM@dest{\expandafter\@fourthoffive#1\@empty\@empty}%
1045  \fi
1046}% \BKM@ref
1047\define@key{BKM}{ref}{%
1048  \Hy@babelnormalise\itf@BKM@ref[{#1}]}

```

1049 </bookmark>

21.8 † Interface **hypbmsec**

21.8.1 interfaces-hypbmsec

```

1050<{*hypbmsec}
1051\ProvidesPackage{interfaces-hypbmsec}
1052  [2010/10/05 v1.0 lastsection etc. (FC)]
1053\RequirePackage{interfaces-base}
1054\itf@Check@Package{hypbmsec}
1055\AfterPackage*{interfaces-titlesec}{% requires interface-titlesec
1056  \input{interfaces-hypbmsec.code.tex}%
1057}%
1058</hypbmsec>

```

21.8.2 interfaces-hypbmsec.code.tex

```

1059 <*hypbmsec.code.tex>
1060 \ProvidesFile{hypbmsec.code.tex}
1061 [2010/10/05 v1.0 code for interfaces-hypbmsec (FC)]

```

\hbs@do (patch)

```

1062 \itf@patch\pretocmd\hbs@do{%
1063   \edef\itf@tp{\expandafter\itf@strip@at\string#1}%
1064   }%
1065   {\PackageInfo{interfaces-hypbmsec}
1066     {Command \string\hbs@do space successfully patched}}
1067   {\PackageWarning{interfaces-hypbmsec}
1068     {Command \string\hbs@do space could not be patched
1069       \MessageBreak loading of 'interfaces-hypbmsec' aborted
1070       \MessageBreak \string\lastsection space will not work}%
1071   \endinput}%

```

\hbs@process (patch)

```

1072 \itf@patch\pretocmd\hbs@process{%
1073   \itf@hbs@next{#1}%
1074   }%
1075   {\PackageInfo{interfaces-hypbmsec}
1076     {Command \string\hbs@process space successfully patched}}
1077   {\PackageWarning{interfaces-hypbmsec}
1078     {Command \string\hbs@process space could not be patched
1079       \MessageBreak loading of 'interfaces-hypbmsec' aborted
1080       \MessageBreak \string\lastsection space will not work}%
1081   \endinput}%

```

\itf@hbs@next

```

1082 \long\def\itf@hbs@next#1{%
1083   \csdef{itf@next\itf@tp title}{#1}%
1084   \ifx\hbs@tocstring\relax
1085     \csletcs{itf@next\itf@tp toc}{itf@next\itf@tp}%
1086   \else
1087     \cslet{itf@next\itf@tp toc}\hbs@tocstring
1088   \fi
1089   \ifx\hbs@bmstring\relax
1090     \csletcs{itf@next\itf@tp bookmark}{itf@next\itf@tp toc}%
1091   \else
1092     \cslet{itf@next\itf@tp bookmark}\hbs@bmstring
1093   \fi
1094   \expandafter\expandafter\expandafter
1095   \expandafter\expandafter\expandafter
1096   \expandafter\itf@hbs@chkclass
1097     \expandafter\expandafter\expandafter\string
1098     \expandafter\hbs@seccmd\string\ttl@\@nil
1099   {}
1100   {\expandafter\itf@last@from@next\expandafter{\itf@tp}}%
1101 }% \itf@hbs@next
1102 \edef\itf@tp{%
1103   \long\def\noexpand\itf@hbs@chkclass##1\string\ttl@##2\noexpand\@nil{%
1104     \noexpand\ifblank{##1}%
1105   }% \itf@hbs@chkclass
1106 }\itf@tp

```

\ttl@select (patch)

```

1107 \itf@patch\pretocmd\ttl@select{%

```



```

1108 \itf@last@from@next{#1}}
1109 {\PackageInfo{interfaces-hypbmsec}
1110   {Command \string\ttl@select\space successfully patched}}
1111 {\PackageWarning{interfaces-hypbmsec}
1112   {Command \string\ttl@select\space could not be patched
1113   \MessageBreak loading of ‘interfaces-hypbmsec’ aborted
1114   \MessageBreak \string\lastsection\space will not work}%
1115   \endinput}%

```

\itf@last@from@next

```

1116 \def\itf@last@from@next#1{%
1117   \expandafter\let\csname last#1title\expandafter\endcsname
1118     \csname itf@next#1title\endcsname
1119   \expandafter\let\csname itf@last#1toc\expandafter\endcsname
1120     \csname itf@next#1toc\endcsname
1121   \expandafter\let\csname last#1bookmark\expandafter\endcsname
1122     \csname itf@next#1bookmark\endcsname
1123   \expandafter\def\csname lastsectionname\endcsname{#1}%
1124 }% \itf@last@from@next

```

```

1125 %%-----
1126 %% \lastsection reinitialization (at \stepcounter-> \cl@section)
1127 \def\itf@hbs@resetlast#1{%
1128   \ifcsname last#1\endcsname
1129     \global\expandafter\let
1130       \csname last#1\endcsname \@empty
1131   \fi
1132 }% \itf@hbs@resetlast
1133 \def\itf@hbs@reset#1{%
1134   \itf@hbs@resetlast{#1title}%
1135   \itf@hbs@resetlast{#1bookmark}%
1136   \itf@hbs@resetlast{#1toc}%
1137   \csuse{cl@#1}%
1138 }% \itf@hbs@reset

```

```

1139 \let\ltx@stpelt\@stpelt
1140 \def\itf@hbs@stpelt#1{%
1141   \let\@elt\itf@hbs@reset
1142   \@elt{#1}%
1143   \let\@elt\itf@stpelt
1144 }% \itf@hbs@stpelt
1145 \def\itf@stpelt#1{%
1146   \ltx@stpelt{#1}%
1147   \itf@hbs@stpelt{#1}%
1148 }% \itf@stpelt
1149 \let\@stpelt\itf@stpelt
1150 %%-----

```

```

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

```

```
1160 </hypbmsec.code.tex>
```

21.9 † Interface fancyhdr : \pagesetup

```

1161 \*fancyhdr
1162 \ProvidesPackage{interfaces-fancyhdr}
1163   [2010/07/22 v1.0 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@gb1\relax % Just in case...
1171   \@ifstar
1172     {\itf@pagesetup{}}
1173     {\itf@pagesetup{reset,}}%
1174 }% \pagesetup
1175 \def\itf@pagesetup#1{\@testopt
1176   {\itf@pagesetup{#1}}}%
1177 }% \itf@pagesetup
1178 \def\itf@pagesetup#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@pagesetup
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 \pagesetup

```

1204 \pgfkeys{%
1205   /handlers/.fancy@hf/.code=%
1206   \pgfkeysalso{\pgfkeyscurrentpath/.store as code=%}
1207   \csedef{f@ncy#1}{%
1208     \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath/font}
1209     \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\strut},%
1210 }% handler for fancyhdr interface
1211 \pgfqkeys{/itf/fancyhdr}{% keys for \pagesetup
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 rule/color/.style={head/rule/color={#1},foot/rule/color={#1}},%
1269 foot/rule/skip/.store as glue in cs=footruleskip,%
1270 foot/rule/skip/.default=.3\normalbaselineskip,%
1271 foot/rule/skip/.blank=z@,%
1272 foot/ruleskip/.alias*=foot/rule/skip,% => puis les alias&style: /.alias
1273 footruleskip/.alias*=foot/rule/skip,% crée auss
1274 font/.style={head/font={#1},foot/font={#1}},%
1275 head/.style={head/left={#1},head/center={#1},head/right={#1}},%
1276 head/font/.style={head/left/font={#1},head/center/font={#1},head/right/font={#1}},%
1277 head/font+/.style={head/left/font+={#1},head/center/font+={#1},head/right/font+={#1}},%
1278 nohead/.style={head=,head/font=,head/rule=},
1279 foot/.style={foot/left={#1},foot/center={#1},foot/right={#1}},%
1280 foot/font/.style={foot/left/font={#1},foot/center/font={#1},foot/right/font={#1}},%
1281 foot/font+/.style={foot/left/font+={#1},foot/center/font+={#1},foot/right/font+={#1}},%
1282 nofoot/.style={foot=,foot/font=,foot/rule=},
1283 left/.style={head/left={#1},foot/left={#1}},%
1284 left/font/.style={head/left/font={#1},foot/left/font={#1}},%
1285 left/font+/.style={head/left/font+={#1},foot/left/font+={#1}},%
1286 center/.style={head/center={#1},foot/center={#1}},%
1287 center/font/.style={head/center/font={#1},foot/center/font={#1}},%
1288 center/font+/.style={head/center/font+={#1},foot/center/font+={#1}},%
1289 right/.style={head/right={#1},foot/right={#1}},%
1290 right/font/.style={head/right/font={#1},foot/right/font={#1}},%
1291 right/font+/.style={head/right/font+={#1},foot/right/font+={#1}},%
1292 head/left/.style={head/left/even={#1},head/left/odd={#1}},%
1293 head/left/font/.style={head/left/even/font={#1},head/left/odd/font={#1}},%
1294 head/left/font+/.style={head/left/even/font+={#1},head/left/odd/font+={#1}},%
1295 head/center/.style={head/center/even={#1},head/center/odd={#1}},%
1296 head/center/font/.style={head/center/even/font={#1},head/center/odd/font={#1}},%
1297 head/center/font+/.style={head/center/even/font+={#1},head/center/odd/font+={#1}},%
1298 head/right/.style={head/right/even={#1},head/right/odd={#1}},%
1299 head/right/font/.style={head/right/even/font={#1},head/right/odd/font={#1}},%
1300 head/right/font+/.style={head/right/even/font+={#1},head/right/odd/font+={#1}},%
1301 foot/left/.style={foot/left/even={#1},foot/left/odd={#1}},%
1302 foot/left/font/.style={foot/left/even/font={#1},foot/left/odd/font={#1}},%
1303 foot/left/font+/.style={foot/left/even/font+={#1},foot/left/odd/font+={#1}},%
1304 foot/center/.style={foot/center/even={#1},foot/center/odd={#1}},%
1305 foot/center/font/.style={foot/center/even/font={#1},foot/center/odd/font={#1}},%
1306 foot/center/font+/.style={foot/center/even/font+={#1},foot/center/odd/font+={#1}},%
1307 foot/right/.style={foot/right/even={#1},foot/right/odd={#1}},%
1308 foot/right/font/.style={foot/right/even/font={#1},foot/right/odd/font={#1}},%
1309 foot/right/font+/.style={foot/right/even/font+={#1},foot/right/odd/font+={#1}},%
1310 head/left/odd/.fancy@hf=olh,%
1311 head/left/odd/font/.store as code=+,%
1312 odd/head/left/.alias=head/left/odd,%
1313 odd/head/left/font/.alias*=head/left/odd/font,%
1314 head/odd/left/.alias=head/left/odd,%
1315 head/odd/left/font/.alias*=head/left/odd/font,%
1316 head/center/odd/.fancy@hf=och,%
1317 head/center/odd/font/.store as code=+,%
1318 odd/head/center/.alias=head/center/odd,%
1319 odd/head/center/font/.alias*=head/center/odd/font,%
1320 head/odd/center/.alias=head/center/odd,%
1321 head/odd/center/font/.alias*=head/center/odd/font,%
1322 head/right/odd/.fancy@hf=orh,%
1323 head/right/odd/font/.store as code=+,%
1324 odd/head/right/.alias=head/right/odd,%
1325 odd/head/right/font/.alias*=head/right/odd/font,%
1326 head/odd/right/.alias=head/right/odd,%
1327 head/odd/right/font/.alias*=head/right/odd/font,%

```

```

1328 head/left/even/.fancy@hf=elh,%
1329 head/left/even/font/.store as code=+,%
1330     even/head/left/.alias=head/left/even,%
1331     even/head/left/font/.alias*=head/left/even/font,%
1332     head/even/left/.alias=head/left/even,%
1333     head/even/left/font/.alias*=head/left/even/font,%
1334 head/center/even/.fancy@hf=ech,%
1335 head/center/even/font/.store as code=+,%
1336     even/head/center/.alias=head/center/even,%
1337     even/head/center/font/.alias*=head/center/even/font,%
1338     head/even/center/.alias=head/center/even,%
1339     head/even/center/font/.alias*=head/center/even/font,%
1340 head/right/even/.fancy@hf=erh,%
1341 head/right/even/font/.store as code=+,%
1342     even/head/right/.alias=head/right/even,%
1343     even/head/right/font/.alias*=head/right/even/font,%
1344     head/even/right/.alias=head/right/even,%
1345     even/head/right/font/.alias*=head/right/even/font,%
1346 foot/left/odd/.fancy@hf=olf,%
1347 foot/left/odd/font/.store as code=+,%
1348     odd/foot/left/.alias=foot/left/odd,%
1349     odd/foot/left/font/.alias*=foot/left/odd/font,%
1350     foot/odd/left/.alias=foot/left/odd,%
1351     foot/odd/left/font/.alias*=foot/left/odd/font,%
1352 foot/center/odd/.fancy@hf=ocf,%
1353 foot/center/odd/font/.store as code=+,%
1354     odd/foot/center/.alias=foot/center/odd,%
1355     odd/foot/center/font/.alias*=foot/center/odd/font,%
1356     foot/odd/center/.alias=foot/center/odd,%
1357     foot/odd/center/font/.alias*=foot/center/odd/font,%
1358 foot/right/odd/.fancy@hf=orf,%
1359 foot/right/odd/font/.store as code=+,%
1360     odd/foot/right/.alias=foot/right/odd,%
1361     odd/foot/right/font/.alias*=foot/right/odd/font,%
1362     odd/foot/right/.alias=foot/right/odd,%
1363     odd/foot/right/font/.alias*=foot/right/odd/font,%
1364 foot/left/even/.fancy@hf=elf,%
1365 foot/left/even/font/.store as code=+,%
1366     even/foot/left/.alias=foot/left/even,%
1367     even/foot/left/font/.alias*=foot/left/even/font,%
1368     foot/even/left/.alias=foot/left/even,%
1369     foot/even/left/font/.alias*=foot/left/even/font,%
1370 foot/center/even/.fancy@hf=ecf,%
1371 foot/center/even/font/.store as code=+,%
1372     even/foot/center/.alias=foot/center/even,%
1373     even/foot/center/font/.alias*=foot/center/even/font,%
1374     foot/even/center/.alias=foot/center/even,%
1375     foot/even/center/font/.alias*=foot/center/even/font,%
1376 foot/right/even/.fancy@hf=erf,%
1377 foot/right/even/font/.store as code=+,%
1378     even/foot/right/.alias=foot/right/even,%
1379     even/foot/right/font/.alias*=foot/right/even/font,%
1380     foot/even/right/.alias=foot/right/even,%
1381     foot/even/right/font/.alias*=foot/right/even/font,%
1382 offset/.style={head/offset={#1},foot/offset={#1}},%
1383 head/offset/.style={head/left/offset={#1},head/right/offset={#1}},%
1384     offset/head/.alias=head/offset,%
1385 foot/offset/.style={foot/left/offset={#1},foot/right/offset={#1}},%
1386     offset/foot/.alias=foot/offset,%
1387     head/left/offset/.style={head/left/even/offset={#1},head/left/odd/offset={#1}},%
1388     offset/head/left/.alias=head/left/offset,%
1389     head/right/offset/.style={head/right/even/offset={#1},head/right/odd/offset={#1}},%
1390     offset/head/right/.alias=head/right/offset,%

```

```

1391   foot/left/offset/.style={foot/left/even/offset={#1},foot/left/odd/offset={#1}},%
1392   offset/foot/left/.alias=foot/left/offset,%
1393   foot/right/offset/.style={foot/right/even/offset={#1},foot/right/odd/offset={#1}},%
1394   offset/foot/right/.alias=foot/right/offset,%
1395   head/left/even/offset/.store in skip cs=f@ncy0@elh,%
1396   offset/head/left/even/.alias=head/left/even/offset,%
1397   head/left/odd/offset/.store in skip cs=f@ncy0@olh,%
1398   offset/head/left/odd/.alias=head/left/odd/offset,%
1399   head/right/even/offset/.store in skip cs=f@ncy0@erh,%
1400   offset/head/right/even/.alias=head/right/even/offset,%
1401   head/right/odd/offset/.store in skip cs=f@ncy0@orh,%
1402   offset/head/right/odd/.alias=head/right/odd/offset,%
1403   foot/left/even/offset/.store in skip cs=f@ncy0@elf,%
1404   offset/foot/left/even/.alias=foot/left/even/offset,%
1405   foot/left/odd/offset/.store in skip cs=f@ncy0@olf,%
1406   offset/foot/left/odd/.alias=foot/left/odd/offset,%
1407   foot/right/even/offset/.store in skip cs=f@ncy0@erf,%
1408   offset/foot/right/even/.alias=foot/right/even/offset,%
1409   foot/right/odd/offset/.store in skip cs=f@ncy0@orf,%
1410   offset/foot/right/odd/.alias=foot/right/odd/offset,%
1411   reset/.style={norule,head=,foot=,offset=\z@,foot/rule/skip,rule/font=,font=},%
1412   reset/.value forbidden,%
1413 }% keys for \pagesetup (stored into token \itf@tk)
1414 \AtBeginDocument{%
1415   \ifdim\headwidth<\z@
1416 %% This catches the case that \headwidth hasn't been initialized and the
1417 %% case that the user added something to \headwidth in the expectation that
1418 %% it was initialized to \textwidth. We compensate this now. This loses if
1419 %% the user intended to multiply it by a factor. But that case is more
1420 %% likely done by saying something like \headwidth=1.2\textwidth.
1421 %% The doc says you have to change \headwidth after the first call to
1422 %% \pagestyle{fancy}. This code is just to catch the most common cases were
1423 %% that requirement is violated.
1424   \global\advance\headwidth123456789sp\global\advance\headwidth\textwidth
1425 \fi}
1426 \fancyhdr

```

21.10 † marks

```

1427 <marks>
1428 \ProvidesPackage{interfaces-marks}
1429   [2010/10/12 v1.0 easier use of e-TeX marks (FC)]
1430 \RequirePackage{interfaces-base}

```

\marksthe

```

1431 \newrobustcmd\marksthe[2]{%
1432   \ifcsname marks@#1\endcsname\else
1433     \itf@marks@new{#1}%
1434   \fi
1435   \expandafter\marks\csname marks@#1\endcsname{\unexpanded{#2}}%
1436 }% \marksthe

```

\marksthecs

```

1437 \newrobustcmd*\marksthecs[2]{%
1438   \ifcsname marks@#1\endcsname\else
1439     \itf@marks@new{#1}%
1440   \fi
1441   \expandafter\marks\csname marks@#1\endcsname{%
1442     \unexpanded\expandafter\expandafter\expandafter{%
1443       \csname#2\endcsname}%
1444 }% \marksthecs

```


`\itf@marks@new`

```

1445 \def\itf@marks@new#1{%
1446   \expandafter\newmarks\csname marks@#1\endcsname
1447   \PackageInfo{interfaces-marks}
1448   {New marks ‘#1’ defined}
1449 }% \itf@marks@warning

```

`\thefirstmarks``\thebotmarks``\thetopmarks`

```

1450 \newcommand*\thefirstmarks[1]{%
1451   \ifcsname marks@#1\endcsname
1452   \expandafter\firstmarks\csname marks@#1\endcsname
1453   \fi
1454 }% \thefirstmarks
1455 \newcommand*\thebotmarks[1]{%
1456   \ifcsname marks@#1\endcsname
1457   \expandafter\botmarks\csname marks@#1\endcsname
1458   \fi
1459 }% \thebotmarks
1460 \newcommand*\thetopmarks[1]{%
1461   \ifcsname marks@#1\endcsname
1462   \expandafter\topmarks\csname marks@#1\endcsname
1463   \fi
1464 }% \thetopmarks

```

`\getthemarks`

```

1465 %% \getthemarks\firstmarks{section}\macro
1466 \newrobustcmd*\getthemarks[3]{%
1467   \ifcsname marks@#2\endcsname
1468   \expandafter\itf@getthemarks\expandafter#1\csname marks@#2\endcsname{#3}%
1469   \else \let#3\@undefined
1470   \fi
1471 }% \getthemarks
1472 \def\itf@getthemarks#1#2#3{%
1473   \expandafter\def\expandafter#3\expandafter{#1#2}%
1474 }% \itf@getthemarks

```

`\getthefirstmarks``\getthebotmarks``\getthetopmarks`

```

1475 \newrobustcmd*\getthefirstmarks[2]{\getthemarks\firstmarks{#1}{#2}}
1476 \newrobustcmd*\getthebotmarks[2]{\getthemarks\botmarks{#1}{#2}}
1477 \newrobustcmd*\getthetopmarks[2]{\getthemarks\topmarks{#1}{#2}}

```

`\ifmarksequal`

```

1478 %% \ifmarksequal\firstmarks\botmarks{subsection}{ true }{ false }
1479 \newrobustcmd*\ifmarksequal[3]{%
1480   \begingroup
1481     \getthemarks{#1}{#3}\@tempa
1482     \getthemarks{#2}{#3}\@tempb
1483   \expandafter\endgroup
1484   \csname ltx@\ifdefined\@tempa\ifdefined\@tempb
1485     \ifx\@tempa\@tempb first\else
1486     second\fi\else
1487     second\fi\else
1488     second\fi
1489   oftwo\endcsname
1490 }% \ifmarksequal

```

1491 </marks>

21.11 † Interface ⇄ **tocloft** : \tocsetup, \lofsetup and \lotsetup

```
1492 <*tocloft>
1493 \ProvidesPackage{interfaces-tocloft}
1494 [2010/07/23 v1.0 pgfkeys macros for tocloft (FC)]
1495 \RequirePackage{interfaces-base}
1496 \itf@Check@Package{tocloft}
1497 \itf@pgfkeys
```

```
\tocsetup The user commands.
\lofsetup
\lotsetup
\listofsetup
1498 \newrobustcmd\tocsetup[1]{%
1499   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/toc}{#1}}%
1500 }% \tocsetup
1501 \newrobustcmd\lofsetup[1]{%
1502   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/lof}{#1}}%
1503 }% \lofsetup
1504 \newrobustcmd\lotsetup[1]{%
1505   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/lot}{#1}}%
1506 }% \lotsetup
1507 \newrobustcmd\listofsetup[2]{%
1508   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/#1}{#2}}%
1509 }% \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.

```
1510 \def\itf@cft@newlistof#1{%
1511   \pgfqkeys{/itf/tocloft/#1}{%
1512     pagestyle/.code=\tocloftpagestyle{##1},%
1513     parskip/.store in skip cs=cftparskip,%
1514     parsep/.alias=parskip,%
1515     mark/.store in cs=cftmark#1,%
1516     marks/.alias=mark,%
1517     label/.store as code,%
1518     ref/.store as code,%
1519     ref/.initial=\csname cft#1title\endcsname,% overwritten (toc,lof,lot)
1520     ref/.default=\csname cft#1title\endcsname,% overwritten (toc,lof,lot)
1521     ref/.blank=\pgfkeysnovalue,%
1522     title/.store as code in cs=cft#1title,% overwritten below for toc, lof and lot
1523     title/top/.store in skip cs=cftbefore#1titleskip,%
1524     beforetitleskip/.alias*=title/top,%
1525     title/bottom/.store in skip cs=cftafter#1titleskip,%
1526     aftertitleskip/.alias*=title/bottom,%
1527     title/font/.store as code in cs=cft#1titlefont,%
1528     title/format/.alias*=title/font,%
1529     title/after/.store as code in cs=cftafter#1title,%
1530     dotsep/.code=\itf@units@mu\itf@musk{##1}% converts to mu
1531     \edef\cftdotsep{\itf@strip@mu\itf@musk},%
1532     dotsep/.default=4.5mu,%
1533     dotsep/.blank=\cftnodots,%
1534     dot/.store in=\cftdot,%
1535     before/.store as code,%
1536     after/.store as code,%
1537     columns/before@hook/.is private=interfaces-tocloft,%
1538     columns/after@hook/.is private=interfaces-tocloft,%
1539     columns/before@hook/.initial=,%
1540     columns/after@hook/.initial=,%
1541     columns/.code={\ltx@ifpackageloaded{multicol}}%
1542     \pgfkeysEsetvalue\pgfkeyscurrentkey{\number##1}%
1543     \ifnum\number##1>\@ne
```

```

1544 \pgfkeysgetvalue{/itf/tocloft/#1/columns/before@hook}\itf@tp
1545 \ifx\itf@tp\@empty
1546 \pgfkeyssetvalue{/itf/tocloft/#1/columns/before@hook}{%
1547 \begingroup
1548 \multicolsep\pgfkeysvalueof{/itf/tocloft/#1/multicolsep}\relax
1549 \begin{multicols}{\pgfkeysvalueof{/itf/tocloft/#1/columns}}%
1550 \columnseprule\pgfkeysvalueof{/itf/tocloft/#1/columns/rule}\relax
1551 \columnsep\pgfkeysvalueof{/itf/tocloft/#1/columns/sep}\relax
1552 \pgfkeysgetvalue{/itf/tocloft/#1/columns/rule/color}\columnseprulecolor
1553 \pgfkeysvalueof{/itf/tocloft/#1/multicols/begin}%
1554 }%
1555 \pgfkeyssetvalue{/itf/tocloft/#1/columns/after@hook}{%
1556 \pgfkeysvalueof{/itf/tocloft/#1/multicols/end}%
1557 \end{multicols}
1558 \endgroup
1559 }%
1560 \fi
1561 \else
1562 \pgfkeyslet\pgfkeyscurrentkey\@ne
1563 \pgfkeyslet{/itf/tocloft/#1/columns/before@hook}\@empty
1564 \pgfkeyslet{/itf/tocloft/#1/columns/after@hook}\@empty
1565 \fi}
1566 {\PackageError{interfaces-tocloft}
1567 {Key ‘columns’: package multicol is required}\@eha}%
1568 },%
1569 columns/.default=2,columns/.blank=1,%
1570 multicol/.alias=columns,%
1571 multicols/.alias=columns,%
1572 twocolumns/.true=\pgfkeysalso{columns=2},%
1573 twocolumns/.false=\pgfkeysalso{columns=1},%
1574 columns/rule/.store as dimen,%
1575 columns/rule/.default=\columnseprule,%
1576 columns/rule,% assign default
1577 columnseprule/.alias*=columns/rule,%
1578 columns/sep/.store as dimen,%
1579 columns/sep/.default=\columnsep,%
1580 columns/sep/.blank=10\p@,%
1581 columns/sep,% assign default
1582 columnsep/.alias*=columns/sep,%
1583 columns/rule/color/.store as color command,%
1584 columns/rule/color,% assign default (\normalcolor)
1585 columnseprulecolor/.alias*=columns/rule/color,%
1586 columns/rulecolor/.alias*=columns/rule/color,%
1587 multicolsep/.store as glue,%
1588 multicolsep/.default=0pt plus4pt minus3pt,%
1589 multicolsep,% sets default
1590 multicols/begin/.initial=,multicols/begin/.default=,%
1591 multicols/before/.alias=multicols/begin,%
1592 multicols/end/.initial=,multicols/end/.default=,%
1593 multicols/after/.alias=multicols/end,%
1594 }% \pgfkeys
1595 \itf@AfterPackage{interfaces-bookmark}{%
1596 \pgfkeys{/itf/tocloft/#1}{%
1597 bookmark/.is multiple choice,%
1598 bookmark/bold/.true=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{},{bold=true},%
1599 bookmark/bold/.false=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{},{bold=false},%
1600 bookmark/italic/.true=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{},{italic=true},%
1601 bookmark/italic/.false=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{},{italic=false},%
1602 bookmark/color/.code=\pgfkeysaddvalue{/itf/tocloft/#1/bookmark}{},{color={#1}},%
1603 bookmark/level/.code=\pgfkeysEaddvalue{/itf/tocloft/#1/bookmark}{},{level={#1}},%
1604 \ifcsname toclevel@##1\endcsname
1605 \csname toclevel@##1\endcsname
1606 \else##1\fi}},%

```

```

1607         bookmark/text/.store as code,%
1608         bookmark/text/.default=\contentsname,%
1609         bookmark/.unknown/.code=\pgfkeysEaddvalue{/itf/tocloft/#1/bookmark}{\pgfkeysCu
1610         bookmark/.initial=,%
1611     }% \pgfkeys
1612     }% AfterPackage interfaces-bookmark
1613     %% while making a format \jobname is expanded to the name of the format... Therefore,
1614     \AtEndPreamble{%
1615         \BeforeFile{\jobname.#1}{%
1616             \pgfkeysvalueof{/itf/tocloft/#1/before}%
1617             \pgfkeysvalueof{/itf/tocloft/#1/columns/before@hook}%
1618         }%
1619         \AfterFile{\jobname.#1}{%
1620             \pgfkeysvalueof{/itf/tocloft/#1/columns/after@hook}%
1621             \pgfkeysvalueof{/itf/tocloft/#1/after}%
1622         }%
1623     }%
1624     \ifcsname #1setup\endcsname\else
1625         \pgfqkeys{/itf/tocloft/#1}{%
1626             pagenumbers/.true=\@cftpnumon{#1},%
1627             pagenumbers/.false=\@cftpnumoff{#1},%
1628             pagenumber/.alias=pagenumbers
1629         }%
1630         \expandafter\pretocmd\csname @cftmake#1title\expandafter\endcsname
1631         \expandafter{\expandafter\itf@tocloft\csname cft#1title\endcsname{#1}{listof#1}}
1632         {\PackageInfo{interfaces-tocloft}
1633             {Command \expandafter\string\csname @cftmake#1title\endcsname
1634                 successfully augmented}}
1635         {\PackageWarning{interfaces-tocloft}
1636             {Command \expandafter\string\csname @cftmake#1title\endcsname
1637                 \space could not be patched
1638                 \MessageBreak Keys label and bookmark won't work for \string\listofsetup}}}%
1639     \fi
1640 }% \itf@cft@newlistof

```

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

```

1641 \itf@cft@newlistof{toc}
1642 \itf@cft@newlistof{lof}
1643 \itf@cft@newlistof{lot}

1644 %% Specific to toc only -----
1645 \pgfqkeys{/itf/tocloft/toc}{%
1646     depth/.defcounter=tocdepth,%
1647     depth/.value required,%
1648     tocdepth/.alias=depth,%
1649     title/.store as code in cs=contentsname,%
1650     contentsname/.alias=title,%
1651     ref/.initial=\contentsname,%
1652     ref/.default=\contentsname,%
1653     pagenumbers/.style={% boolean key
1654         part/pagenumbers={#1},chapter/pagenumbers={#1},section/pagenumbers={#1},%
1655         subsubsection/pagenumbers={#1},subsubsection/pagenumbers={#1},%
1656         paragraph/pagenumbers={#1},subparagraph/pagenumbers={#1}},%
1657     pagenumber/.alias=pagenumbers,%
1658 }%
1659 %% Specific to lof and lot -----
1660 \pgfqkeys{/itf/tocloft/lof}{%
1661     title/.store as code in cs=listfigurename,%
1662     listfigurename/.alias=title,%
1663     ref/.initial=\listfigurename,%
1664     ref/.default=\listfigurename,%
1665 }%
1666 \pgfqkeys{/itf/tocloft/lot}{%

```

```

1667 title/.store as code in cs=listtablename,%
1668 listtablename/.alias=title,%
1669 ref/.initial=\listtablename,%
1670 ref/.default=\listtablename,%
1671 }%
1672 \if@cftsubfigopt
1673 \pgfqkeys{/itf/tocloft}{%
1674   lof/pagenumbers/.style={figure/pagenumbers={#1},subfigure/pagenumbers={#1}},%
1675   lof/pagenumbers/.alias=lof/pagenumbers,%
1676   lot/pagenumbers/.style={figure/pagenumbers={#1},subfigure/pagenumbers={#1}},%
1677   lot/pagenumbers/.alias=lot/pagenumbers,%
1678 }%
1679 \else
1680 \pgfqkeys{/itf/tocloft}{%
1681   lof/pagenumbers/.style={figure/pagenumbers={#1}},%
1682   lof/pagenumbers/.alias=lof/pagenumbers,%
1683   lot/pagenumbers/.style={figure/pagenumbers={#1}},%
1684   lot/pagenumbers/.alias=lot/pagenumbers,%
1685 }%
1686 \fi

```

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

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

```

1687 \def\itf@cft@name#1.#2.#3\@nil{cft#1\itf@cft@secname{#2}#3}
1688 \def\itf@cft@secname#1{%
1689   \expandafter\itf@cft@sec@name\csname#1\endcsname{#1}%
1690 }% \itf@cft@secname
1691 \def\itf@cft@sec@name#1#2{%
1692   \ifx#1\chapter chap\else
1693     \ifx#1\section sec\else
1694       \ifx#1\subsection subsec\else
1695         \ifx#1\subsubsection subsubsec\else
1696           \ifx#1\paragraph para\else
1697             \ifx#1\subparagraph subpara\else
1698               \ifx#1\figure fig\else
1699                 \ifx#1\subfigure subfig\else
1700                   \ifx#1\table tab\else
1701                     \ifx#1\subtable subtab\else
1702                       #2\fi
1703   \fi\fi\fi\fi\fi\fi\fi\fi\fi
1704 }% \itf@cft@section@name

```

21.11.1 Specific handlers for tocloft

```

1705 %% tocloft setup for tables sections :
1706 \pgfqkeys{%
1707   /handlers/.cft@def/.style={\pgfkeyscurrentpath/.store in cs=\itf@cft@name #1\@nil},%
1708   /handlers/.cft@code/.style={\pgfkeyscurrentpath/.store as code in cs=\itf@cft@name #1\@nil},%
1709   /handlers/.cft@length/.style={\pgfkeyscurrentpath/.store in skip cs=\itf@cft@name #1\@nil},%
1710 }

```

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

```

1711 \def\itf@cft@newsection#1#2{% #1=part / chap / sec / subsec etc., #2 = #1 = toc | lof | lot
1712   \pgfqkeys{/itf/tocloft/#2/#1}{%
1713     skip/.cft@length=before.#1.skip,%
1714     before/.alias*=skip,%
1715     top/.alias*=skip,%
1716     indent/.cft@length=#1.indent,%

```

```

1717 font/.cft@code=.#1.font,%
1718 font/.append code={\pgfkeysalso{/itf/tocloft/#2/#1/pagewidth/font={##1}}},%
1719 format/.alias*=font,%
1720 number/width/.cft@length=.#1.numwidth,%
1721 numwidth/.alias*=number/width,%
1722 presnum/.cft@code=.#1.presnum,%
1723 number/before/.alias*=presnum,%
1724 aftersnum/.cft@code=.#1.aftersnum,%
1725 number/after/.alias*=aftersnum,%
1726 aftersnumb/.cft@code=.#1.aftersnumb,%
1727 title/before/.alias*=aftersnumb,%
1728 leader/.cft@def=.#1.leader,%
1729 leader/.default=\cftdotfill{\csname\itf@cft@name.#1.dotsep\@nil\endcsname},%
1730 leaders/.alias=leader,%
1731 dotsep/.code={\itf@units@mu\itf@musk{##1}}%
1732 \csedef{\itf@cft@name.#1.dotsep\@nil}{\itf@strip@mu\itf@musk}},%
1733 dotsep/.default=\cftdotsep,%
1734 dotsep/.blank=\cftnodots,%
1735 nodots/.code=\csedef{\itf@cft@name.#1.dotsep\@nil}{\cftnodots},%
1736 nodots/.value forbidden,%
1737 pagefont/.cft@code=.#1.pagefont,%
1738 page/font/.alias*=pagefont,%
1739 pagewidth/font/.alias*=pagefont,%
1740 afterpnum/.cft@code=.#1.afterpnum,%
1741 page/after/.alias*=afterpnum,%
1742 pagewidth/after/.alias*=afterpnum,%
1743 pagewidths/.on=\@cftpnumon{\itf@cft@secname{##1}},%
1744 pagewidths/.off=\@cftpnumoff{\itf@cft@secname{##1}},%
1745 pagewidth/.alias=pagewidths,%
1746 }% \pgfkeys
1747 }% \itf@cft@newsection

1748 \@for\itf@tp:=part,chap,chapter,sec,section,subsec,subsection,%
1749 subsubsec,subsubsection,para,paragraph,subpara,subparagraph%
1750 \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{toc}}%
1751 \if@cftsubfigopt % \ifpackagewith{tocloft}{subfig} ??
1752 \@for\itf@tp:=fig,figure,subfig,subfigure%
1753 \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lof}}%
1754 \@for\itf@tp:=tab,table,subtab,subtable%
1755 \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lot}}%
1756 \else
1757 \@for\itf@tp:=fig,figure%
1758 \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lof}}%
1759 \@for\itf@tp:=tab,table%
1760 \do{\expandafter\itf@cft@newsection\expandafter{\itf@tp}{lot}}%
1761 \fi

@cftmaketoctitle (patch)
@cftmakeloftitle (patch)
@cftmakelottitle (patch)
newlistentry (patch)
1762 \itf@patch\pretocmd\@cftmaketoctitle
1763 {\itf@tocloft\contentsname{toc}{tableofcontents}}
1764 {\PackageInfo{interfaces-tocloft}
1765 {Command \string\@cftmaketoctitle\space successfully augmented}}
1766 {\PackageWarning{interfaces-tocloft}
1767 {Command \string\@cftmaketoctitle\space could not be patched
1768 \MessageBreak label and bookmark keys for \string\tocsetup\space won't work}}%
1769 \itf@patch\pretocmd\@cftmakeloftitle
1770 {\itf@tocloft\listfigurename{lof}{listoffigures}}
1771 {\PackageInfo{interfaces-tocloft}
1772 {Command \string\@cftmakeloftitle\space successfully augmented}}
1773 {\PackageWarning{interfaces-tocloft}
1774 {Command \string\@cftmakeloftitle\space could not be patched
1775 \MessageBreak label and bookmark keys for \string\lofsetup\space won't work}}%

```



```

1776 \itf@patch\pretocmd\@cftmakelottitle
1777   {\itf@tocloft\listtablename{lot}{listoftables}}
1778   {\PackageInfo{interfaces-tocloft}
1779     {Command \string\@cftmakelottitle\space successfully augmented}}
1780   {\PackageWarning{interfaces-tocloft}
1781     {Command \string\@cftmakelottitle\space could not be patched
1782       \MessageBreak label and bookmark keys for \string\lotsetup\space won't work}}}%
1783 \@makeother\#
1784 \itf@patch\apptocmd\newlistentry
1785   {\itf@cft@newlistof{#2}}
1786   {\PackageInfo{interfaces-tocloft}
1787     {Command \string\newlistentry\space successfully augmented}}
1788   {\PackageWarning{interfaces-tocloft}
1789     {Command \string\newlistentry\space could not be patched
1790       \MessageBreak \string\listofsetup\space will not work}}}%
1791 \catcode'\#=6

```

`\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.

```

1792 \def\itf@tocloft#1#2#3{%
1793   \expandafter\def\expandafter#1\expandafter{%
1794     \expandafter\itf@cft@label
1795     \expandafter#1\expandafter{#1}{#2}{#3}}}%
1796 }% \itf@tocloft
1797 \long\def\itf@cft@label#1#2#3#4{%
1798   \def#1{#2}% restore original
1799   \begingroup
1800   \pgfkeysifdefined{/itf/tocloft/#3/bookmark}
1801     {%
1802       \raisedhyperdef{interfaces}{#4}{#2}%
1803       \pgfkeysgetvalue{/itf/tocloft/#3/bookmark/text}\itf@tp
1804       \ifx\itf@tp\@empty\else
1805         \pgfkeysgetvalue{/itf/tocloft/#3/bookmark}\@tempa
1806         \expandafter\bookmark\expandafter[%
1807           \expandafter{\@tempa,dest=interfaces.#4}]%
1808           {\pgfkeysvalueof{/itf/tocloft/#3/bookmark/text}}}%
1809       \global\BKM@startatroottrue
1810       \fi}
1811   {#2}%
1812   \pgfkeysgetvalue{/itf/tocloft/#3/label}\itf@tp
1813   \ifx\itf@tp\@empty\else
1814     \def\@currentHref{interfaces.#4}%
1815     \pgfkeysgetvalue{/itf/tocloft/#3/ref}\@currentlabel
1816     \def\@currentlabelname{#2}%
1817     \label{\pgfkeysvalueof{/itf/tocloft/#3/label}}}%
1818   \fi
1819   \endgroup
1820   \gdef\itf@tp{\def#1{#2}\let\itf@tp\@undefined}% restore original
1821   \aftergroup\itf@tp % simple group introduced by tocloft to print \contentsname
1822 }% \itf@cft@label

1823 </tocloft>

```

21.12 † Interface ⇄ **makecell** : \repeatcell

```

1824 \*makecell>
1825 \ProvidesPackage{interfaces-makecell}
1826   [2010/10/12 v1.0 more control on tabular cells (FC)]
1827 \RequirePackage{interfaces-base}
1828 \itf@Check@Package{makecell}
1829 \edef\itf@AtEnd{\catcode126 \the\catcode126}% 126 = ~
1830 \AtEndOfPackage{\itf@AtEnd}
1831 \catcode126 \active % 126 = ~
1832 \itf@pgfkeys

```

21.12.1 countvalues

\countvalues

```

1833 \def\countvalues#1{\number\numexpr
1834   \expandafter\itf@countvalues\expandafter0#1,\countvalues\@nnil
1835 }% \countvalues
1836 \def\itf@countvalues#1#2,#3\@nnil{%
1837   \ifx\countvalues#3%
1838     +#1\expandafter\remove@to@nnil
1839   \else
1840     +1\expandafter\itf@countvalues
1841   \fi#3\@nnil
1842 }% \itf@countvalues

```

\itf@getvalue

```

1843 \def\itf@getvalue#1#2{%
1844   \romannumeral-'\q\expandafter\itf@getv@lue\expandafter0#1,\itf@getvalue\@nil{#2-1}\@nnil
1845 }% \itf@getvalue
1846 \def\itf@getv@lue#1#2,#3\@nil#4\@nnil{%
1847   \ifnum\numexpr#1=\numexpr#4\relax
1848     #2\expandafter\remove@to@nnil
1849   \else\ifx\itf@getvalue#3%
1850     \expandafter\expandafter\expandafter\remove@to@nnil
1851   \else
1852     \expandafter\expandafter\expandafter\itf@getv@lue
1853   \fi\fi{#1+1}#3\@nil#4\@nnil
1854 }% \itf@getv@lue

```

21.12.2 \repeatcell

\repeatcell

```

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

```

```

1870 \countdef\itf@row=101% local
1871 \countdef\itf@cnt=102% local
1872 \toksdef\itf@cell=1% local
1873 \itf@pgfkeys
1874 \pgfkeys{/itf/makecell/repeatcell}{@reset={#2},#3}%
1875 \toks@{}% local
1876 \ifnum\itf@row>\z@ % number
1877 \edef\itf@rowloop{\noexpand\@whilenum\itf@row<\the\itf@row}%
1878 \def\itf@cell@init{\def\r{\row}\let\R@empty}%
1879 \else % list
1880 \def\itf@rowloop{\@for\r:=\itf@rows}%
1881 \expandafter\def\expandafter\itf@cell@init\expandafter{%
1882 \expandafter\def\expandafter\R\expandafter{%
1883 \expandafter\itf@getvalue\expandafter{\expandafter\@empty\itf@rows}}}%
1884 \fi
1885 \ifnum\itf@col>\z@ % number
1886 \edef\itf@colloop{\noexpand\@whilenum\itf@col<\the\itf@col}%
1887 \expandafter\def\expandafter\itf@cell@init\expandafter{%
1888 \itf@cell@init\def\c{\column}\let\C@empty}%
1889 \else % list
1890 \def\itf@colloop{\@for\c:=\itf@cols}%
1891 \expandafter\expandafter\expandafter\def
1892 \expandafter\expandafter\expandafter\itf@cell@init
1893 \expandafter\expandafter\expandafter{%
1894 \expandafter\itf@cell@init
1895 \expandafter\def\expandafter\C\expandafter{%
1896 \expandafter\itf@getvalue\expandafter{\expandafter\@empty\itf@cols}}}%
1897 \fi
1898 \itf@cell@init % in case they are used in formulas
1899 \ifitf@celltranspose
1900 \itf@col\z@
1901 \itf@colloop\do{%
1902 \ifnum\itf@col>\z@
1903 \pgfkeysifdefined{/itf/makecell/repeatcell/cr/row\the\itf@col}%
1904 {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr/row\the\itf@col}\itf@thecr}%
1905 {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr}\itf@thecr}%
1906 \toks@\expandafter\expandafter\expandafter{%
1907 \expandafter\the\expandafter\toks@\itf@thecr}%
1908 \fi
1909 \advance\itf@col\@ne
1910 \itf@row\z@
1911 \itf@rowloop\do{%
1912 \advance\itf@row\@ne
1913 \ifnum\itf@row>\@ne
1914 \toks@\expandafter{\the\toks@&}%
1915 \fi
1916 \toks@\expandafter\expandafter\expandafter{%
1917 \expandafter\the\expandafter\toks@
1918 \itf@cell@init\itf@ampactive}%
1919 \itf@cell@insert}%
1920 }%
1921 \else
1922 \itf@row\z@
1923 \itf@rowloop\do{%
1924 \ifnum\itf@row>\z@
1925 \pgfkeysifdefined{/itf/makecell/repeatcell/cr/row\the\itf@row}%
1926 {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr/row\the\itf@row}\itf@thecr}%
1927 {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr}\itf@thecr}%
1928 \toks@\expandafter\expandafter\expandafter{%
1929 \expandafter\the\expandafter\toks@\itf@thecr}%
1930 \fi
1931 \advance\itf@row\@ne
1932 \itf@col\z@

```

```

1933 \itf@colloop\do{%
1934 \advance\itf@col\@ne
1935 \ifnum\itf@col>\@ne
1936 \toks@\expandafter{\the\toks@&}%
1937 \fi
1938 \toks@\expandafter\expandafter\expandafter{%
1939 \expandafter\the\expandafter\toks@
1940 \itf@cell@init\itf@ampactive}%
1941 \itf@cell@insert}%
1942 }%
1943 \fi
1944 \itf@message{Code introduced by repeatcell in the \@currenvir:\the\toks@}% (to see the
1945 \expandafter\endgroup\the\toks@
1946 \ignorespaces
1947 }% \itf@repeatcell

```

\itf@message

```
1948 \let\itf@message\@gobble
```

Now the keys for \repeatcell:

```

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

```

```

1990 \pgfkeysEaddvalue{/itf/makecell/repeatcell/@calc}{}
1991 {\the\toks@ \expandafter \noexpand \csname \pgfkeyscurrentname \endcsname}%
1992 \expandafter \let \csname \pgfkeyscurrentname \endcsname \@undefined,% (order of evaluation)
1993 @reset/.code=% #1 are the columns
1994 \itf@celltransposefalse
1995 \let\itf@celltext\@empty
1996 \let\itf@cellfont\@empty
1997 \pgfkeyssetvalue{/itf/makecell/repeatcell/cr}{\}%
1998 \itf@row\@ne
1999 \pgfkeyssetvalue{/itf/makecell/repeatcell/@calc}{%
2000 \itf@cell@numcalc\itf@col\column
2001 \itf@cell@numcalc\itf@row\row}%
2002 \itf@number\itf@col#1\relax\@nnil
2003 {\ifnum\itf@col<\z@\itf@col\@ne\fi}
2004 {\itf@col\z@
2005 \def\itf@cols{#1}\itf@expandonceifcs\itf@cols},%
2006}% keys for \repeatcell

```

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

```

2007 \def\itf@cell@getformula#1{%
2008 \itf@cell@getformula #1$$$ \itf@cell@getformula $\relax\@nnil
2009 \expandafter \ifx \expandafter \toks@ \the \toks@
2010 \itf@cell@formulaerror{#1}\thr@@
2011 \fi
2012}% \itf@cell@getformula
2013 \def\itf@cell@getformula #1$#2$#3$#4$#5$#6{%
2014 \toks@{\toks@}%
2015 \ifx\itf@cell@getformula #4%
2016 \toks@{\itf@cell@numcalc{#1}}%
2017 \else\ifx $#4%
2018 \ifx\itf@cell@getformula #6%
2019 \ifdefined\pgfmathparse\else\itf@cell@formulaerror{#2}4\fi
2020 \toks@{\itf@cell@pgfcalc{#2}}%
2021 \else\ifx $#6%
2022 \ifdefined\FPeval\else\itf@cell@formulaerror{#3}5\fi
2023 \toks@{\itf@cell@FPcalc{#3}}%
2024 \fi\fi
2025 \fi\fi\remove@to\@nnil
2026}% \itf@cell@getformula
2027 \def\itf@cell@formulaerror#1#2{%
2028 \def\itf@tp{#1}\@onelevel@sanitize\itf@tp
2029 \ifcase#2% 0
2030 \PackageError{interfaces-makecell}
2031 {Missing formula for key \pgfkeyscurrentname!}\@ehd
2032 \or %1
2033 \PackageError{interfaces-makecell}
2034 {Computed key \pgfkeyscurrentname\space was left blank!}\@ehd
2035 \or %2
2036 \PackageError{interfaces-makecell}
2037 {Duplicate definition for computed key \pgfkeyscurrentname
2038 \MessageBreak \string\repeatcell\space will redefine the key}\@eha
2039 \or %3
2040 \PackageError{interfaces-makecell}
2041 {Incorrect formula in \string\repeatcell
2042 \MessageBreak Check: \itf@tp
2043 \MessageBreak Take care of commas and equal signs...}\@ehd
2044 \or %4
2045 \PackageError{interfaces-makecell}
2046 {Formula: \itf@tp
2047 \MessageBreak requires \string\pgfmathparse:\space\string\usetikz{pgfmath}}\@ehd
2048 \or %5
2049 \PackageError{interfaces-makecell}

```

```
2050      {Formula: \itf@tp
2051      \MessageBreak requires \string\FPeval:\space\string\usepackage{fp}}\@ehd
2052  \else
2053  \PackageError{interfaces-makecell}
2054      {Impossible value!}\@ehd
2055  \fi
2056 }% \itf@cell@formulaerror
```



```

\itf@cell@calc
\itf@cell@numcalc
\itf@cell@pgfcalc
\itf@cell@FPcalc
2057 \protected\def\itf@cell@calc#1#2{%
2058   \edef#1{\def\noexpand#1{#2}}%
2059   \toks@\expandafter\expandafter\expandafter{%
2060     \expandafter\the\expandafter\toks@#1}%
2061   #1%
2062 }% \itf@cell@calc
2063 \protected\def\itf@cell@numcalc#1#2{%
2064   \itf@cell@calc#2{\number\numexpr#1}%
2065 }% \itf@cell@numcalc
2066 \protected\def\itf@cell@pgfcalc#1#2{%
2067   \pgfmathparse{#1}%
2068   \itf@cell@calc#2\pgfmathresult
2069 }% \itf@cell@pgfcalc
2070 \protected\def\itf@cell@FPcalc#1#2{%
2071   \begingroup
2072   \setbox\z@\vbox{%\FPeval introduces some weird code, hence the calculation in a \vbox
2073     \FPeval\FPresult{#1}%
2074     \global\let\FPresult\FPresult}%
2075   \endgroup
2076   \itf@cell@calc#2\FPresult
2077 }% \itf@cell@FPcalc

```

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

```

2078 \def\itf@cell@getifdefined#1#2{%
2079   \pgfkeysifdefined{/itf/makecell/repeatcell/#1}
2080     {\pgfkeysgetvalue{/itf/makecell/repeatcell/#1}{#2}
2081       {\let#2\empty}}%
2082 }%
2083 \def\itf@cell@insert{%
2084   \pgfkeysvalueof{/itf/makecell/repeatcell/@calc}%
2085   \ifx\R\empty\else
2086     \toks@\expandafter\expandafter\expandafter{%
2087       \expandafter\the\expandafter\toks@
2088       \expandafter\def\expandafter\r\expandafter{\r}}%
2089   \fi
2090   \ifx\C\empty\else
2091     \toks@\expandafter\expandafter\expandafter{%
2092       \expandafter\the\expandafter\toks@
2093       \expandafter\def\expandafter\c\expandafter{\c}}%
2094   \fi
2095   \itf@cell@getifdefined{font/row\the\itf@row}\itf@cellfontrow
2096   \itf@cell@getifdefined{font/col\the\itf@col}\itf@cellfontcol
2097   \itf@cell@getifdefined{text/row\the\itf@row}\itf@cellrow
2098   \itf@cell@getifdefined{text/col\the\itf@col}\itf@cellcol
2099   \toks@\expandafter\expandafter\expandafter{%
2100     \expandafter\the\expandafter\toks@\itf@cellfont}%
2101   \ifx\itf@cellrow\empty
2102     \ifx\itf@cellcol\empty
2103       \itf@cell\expandafter\expandafter\expandafter{%
2104         \expandafter\itf@cellfontcol\itf@cellfontrow}%
2105       \let\itf@tp\itf@celltext
2106     \else
2107       \itf@cell\expandafter{\itf@cellfontcol}%
2108       \let\itf@tp\itf@cellcol
2109     \fi
2110   \else
2111     \ifx\itf@cellcol\empty
2112       \itf@cell\expandafter{\itf@cellfontrow}%
2113       \let\itf@tp\itf@cellrow
2114     \else

```

```

2115 \itf@cell}%
2116 \ifitf@celltranspose
2117 \edef\itf@tp{\noexpand\itf@diaghead(-2,1)
2118   {\expandonce\itf@cellfontrow\expandonce\itf@cellrow}
2119   {\expandonce\itf@cellfontcol\expandonce\itf@cellcol}}%
2120 \else
2121 \edef\itf@tp{\noexpand\itf@diaghead(-2,1)
2122   {\expandonce\itf@cellfontcol\expandonce\itf@cellcol}
2123   {\expandonce\itf@cellfontrow\expandonce\itf@cellrow}}%
2124 \fi
2125 \fi
2126 \fi
2127 \itf@cell\expandafter\expandafter\expandafter{%
2128   \expandafter\the\expandafter\itf@cell\itf@tp}%
2129 \edef\itf@tp{\itf@makecell{\the\itf@cell}}%
2130 \toks@\expandafter\expandafter\expandafter{%
2131   \expandafter\the\expandafter\toks@\itf@tp}%
2132}% \itf@cell@insert

```

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

```

2133 \def\itf@diaghead(#1)#2#3{%
2134   \begingroup
2135   \edef\itf@tp{%
2136     \noexpand\diaghead({#1}){\unexpanded{#2 #3}}%
2137       {\unexpanded{#2}}
2138       {\unexpanded{#3}}}%
2139   \expandafter\endgroup\itf@tp
2140}% \itf@diaghead

```

21.12.3 colortbl: patch for `\CT@cellc`

`\CT@cellc`

```

2141 \AfterPackage*{colortbl}{%
2142 \def\CT@cellc#1[#2]#3{% #1 is either \@firstofone or \@gobble, [#2]#3 = color spec
2143   \xdef\CT@cell@color{\noexpand\CT@color#1{[#2]}}{#3}%
2144   \global\let\noexpand\CT@cell@color\relax}%
2145 }}
2146 \</makecell>

```

21.13 † Interface ⇄ **enumitem** : `\setlist`

```

2147 <*enumitem>
2148 \ProvidesPackage{interfaces-enumitem}
2149 [2010/10/12 v1.0 more control on lists (FC)]
2150 \RequirePackage{interfaces-base}
2151 \itf@Check@Package{enumitem}
2152 \itf@pgfkeys

```

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

```

2153 \providerobustcmd*\interitem{%
2154   \item[]\hskip-\leftmargin
2155}% \interitem

```

`\setlist` (extension, star form)

```

2156 \renewrobustcmd*\setlist{\begingroup \@ifstar
2157   {\let\do\csgappto \itf@setlist}
2158   {\let\do\csgdef \itf@setlist}%
2159 }% \setlist
2160 \newrobustcmd*\itf@setlist[2][0]{%
2161   \let\enit@a\@empty
2162   \let\enit@b\@empty
2163   \@for\x:=#1\do{%
2164     \ifx\x\@empty\else
2165       \itf@number\count@\x\relax\@nnil
2166       {\edef\enit@b{\enit@b,\the\count@}}
2167       {\edef\enit@a{\enit@a,\x}}%
2168     \fi}%
2169   \ifx\enit@a\@empty\def\enit@a{list}\fi
2170   \ifx\enit@b\@empty\def\enit@b{0}\fi
2171   \@for\a:=\enit@a\do{%
2172     \ifx\a\@empty\else
2173       \@for\b:=\enit@b\do{%
2174         \ifx\b\@empty\else
2175           \ifcsundef{end\a}
2176             {\expandafter\itf@setlist@error\expandafter{\a}}
2177             {\do{enit@@\a\romannumeral\b}{#2}}}%
2178         \fi}%
2179     \fi}
2180   \endgroup
2181 }% \itf@setlist
2182 \def\itf@setlist@error#1{%
2183   \PackageError{interfaces-enumitem}
2184     {Environment ‘#1’ is undefined
2185     \MessageBreak Use \string\newlist\space before \string\setlist
2186     \MessageBreak Your command will be ignored}\@eha
2187 }% \itf@setlist@error

2188 </enumitem>

```

21.14 † Interface ⇄ **truncate**

```

2189 <*truncate>
2190 \ProvidesPackage{interfaces-truncate}

```

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

```

2191 \newrobustcmd\truncetext[3][\TruncateMarker]{% do nothing if width of #3 is less than #2
2192   \begingroup
2193     \hbadness\maxdimen\hfuzz\vsize
2194     \setbox\z@=\hbox to\dimexpr#2\relax{\ignorespaces#3\unskip}%
2195     \ifnum\badness>\@M
2196       \def\next{\truncate[#{1}]{#2}{#3}}%
2197     \else \def\next{#3}%
2198     \fi
2199     \expandafter\endgroup\next
2200 }% \truncetext

```

`\htruncadjust`

```

2201 \newrobustcmd\htruncadjust[1]{\@testopt
2202   {\itf@htruncadjust{#1}}}%
2203 }% \htruncadjust
2204 \long\def\itf@htruncadjust#1[#2]#3{% #1=total width, #2=[x_1,...,x_n], #3=text_1,...,text_n
2205   \begingroup
2206   \let\itf@trunc@width\@empty

```

```

2207 \let\itf@istruncated\@gobble % to remove the first comma
2208 \let\the@dimen\relax
2209 \count@ \z@ \dimen@ \z@
2210 \ifblank{#2}% no optional [x_1,...,x_n] parameters
2211 {%
2212   \@for\x:=#3\do{%
2213     \advance \count@ \@ne
2214     \setbox\z@=\hbox{\ignorespaces\x\unskip}%
2215     \dimen\count@ = \wd\z@
2216     \toks\count@\expandafter{\the\dimen\count@}%
2217     \advance\dimen@ \dimen\count@}%
2218   }
2219   {%
2220     \@for\x:=#2\do{%
2221       \advance \count@ \@ne
2222       \toks\count@\expandafter{\x\p@}%
2223       \advance\dimen@ \x\p@
2224       \dimen\count@ -\@ne\p@}%
2225     }%
2226     \advance\count@\@ne \dimen\itf@freespace=\count@
2227     \advance\count@\@ne \dimen\itf@tempdim=\count@
2228     \advance\count@\@ne \dimen\itf@sum=\count@
2229     \count@\z@ %%dimen0=\the\dimen@quad\number\dimen@\par
2230     \itf@freespace\z@ % free space left when w_i < x_i/sum(x_i) * total
2231     \@for\x:=#3\do{%
2232       \advance \count@ \@ne
2233       \ifdim \dimen\count@<\z@ % not measured before
2234         \setbox\z@=\hbox{\ignorespaces\x\unskip}%
2235         \dimen\count@ = \wd\z@
2236       \fi
2237       \itf@tempdim = \itf@scaledimen{#1}*\the\toks\count@/\dimen@\relax
2238       \ifdim \dimen\count@ > \itf@tempdim
2239         \dimen\count@ = \itf@tempdim
2240         \edef\itf@istruncated{\itf@istruncated,\the\count@}%
2241         \advance\itf@sum by\the\toks\count@
2242       \else
2243         \advance\itf@freespace \itf@tempdim
2244         \advance\itf@freespace -\dimen\count@
2245       \fi
2246       \edef\itf@trunc@width{\itf@trunc@width\the\count@\the@dimen\the\count@}%
2247     }%
2248   %% Free space=\the\itf@freespace\par
2249   \ifdim\itf@freespace > \z@ % reinsert free space proportionally
2250     \ifx\itf@istruncated\@gobble\else
2251       \@for\x:=\itf@istruncated\do{%
2252         \advance\dimen\x by\itf@scaledimen\itf@freespace*\the\toks\x/\itf@sum\relax
2253       }%
2254     \fi\fi
2255     \def\the@dimen{\the\dimen}%
2256     \edef\itf@trunc@width{\itf@trunc@width}%
2257     \expandafter\endgroup
2258     \expandafter\def
2259       \expandafter\itf@trunc@width
2260       \expandafter{\itf@trunc@width}%
2261   %% total=\the\dimexpr#1\relax
2262 }% \itf@htruncadjust
2263 \def\itf@scalefactor#1#2{\strip@pt\dimexpr#1*65536/\number#2\relax}
2264 \def\itf@scaledimen#1*#2/#3\relax{%
2265   \dimexpr\itf@scalefactor{#2}{#3}\dimexpr#1\relax\relax
2266 }% \itf@scaledimen

```

\htruncwidth

```

2267 \newcommand*\htruncwidth[1]{% purely expandable
2268   \expandafter\itf@htruncwidth\itf@trunc@width\relax\relax\@nnil{#1}%
2269 }% \htruncwidth
2270 \def\itf@htruncwidth#1#2#3\@nnil#4{%
2271   \ifx\relax#1\itf@trunc@nosuchwidth{#4}%
2272   \else\ifnum#1=#4
2273     #2%
2274   \else
2275     \itf@htruncwidth#3\@nnil{#4}%
2276   \fi\fi
2277 }% \itf@htruncwidth
2278 \def\itf@trunc@nosuchwidth#1{%
2279   \PackageError{interfaces-truncate}
2280     {You requested the width for text nr#1
2281     \MessageBreak but there is no such text
2282     \MessageBreak Therefore no such width!}\@ehd
2283 }% \itf@trunc@nosuchwidth

2284 </truncate>

```

21.15 interface fancyvrb

```

2285 <*fancyvrb>
2286 \ProvidesPackage{interfaces-fancyvrb}
2287   [2010/08/22 v1.0 Custom verbatim environments (FC)]
2288 \RequirePackage{interfaces-base}
2289 \itf@Check@Package{fancyvrb}
2290 \itf@pgfkeys
2291 \def\itf@tp#1{%
2292 \DefineVerbatimEnvironment{itf:VerbLines}{Verbatim}
2293   {gobble=#1,frame=lines,framesep=6pt,fontseries=m}
2294 \DefineVerbatimEnvironment{itf:Verb}{Verbatim}
2295   {gobble=#1,fontseries=m}
2296 \DefineVerbatimEnvironment{itf:Verb*}{Verbatim}
2297   {gobble=#1,fontseries=m,commandchars=$()}
2298 }
2299 \ltx@ifpackageloaded{doc}
2300   {\itf@tp{1}}
2301   {\itf@tp{0}}
2302 \AtBeginDocument{%
2303 \ifdefined\VerbLines\else
2304 \ifdefined\endVerbLines\else
2305   \letcs\VerbLines{itf:VerbLines}%
2306   \letcs\endVerbLines{enditf:VerbLines}%
2307 \fi\fi
2308 \ifdefined\Verb\else\ifdefined\endVerb\else
2309   \letcs\Verb{itf:Verb}%
2310   \letcs\endVerb{enditf:Verb}%
2311 \fi\fi
2312 \ifcsname Verb*\endcsname\else
2313 \ifcsname endVerb*\endcsname\else
2314   \csletcs{Verb*}{itf:Verb*}%
2315   \csletcs{endVerb*}{enditf:Verb*}%
2316 \fi\fi
2317 }
2318 </fancyvrb>

```

21.16 Interface **appendix** and the appendices environment

```

2319 <*appendix>
2320 \ProvidesPackage{interfaces-appendix}
2321   [2010/10/20 v1.0 easy control of appendices environment (FC)]
2322 \RequirePackage{interfaces-base}

```

```

2323 \itf@Check@Package{appendix}
2324 \itf@pgfkeys

```

\appendices

```

2325 \let\itf@appendices@pp\appendices
2326 \let\itf@endappendices@pp@endappendices
2327 \let\itf@appendices@pp\appendices
2328 \let\itf@endappendices@pp@endappendices
2329 \renewenvironment{appendices}[1][1]{%
2330   \pgfkeys{/itf/appendix/appendices}{#1}%
2331   \itf@appendices@pp
2332 }
2333 {\itf@endappendices@pp}%

2334 \pgfkeys{/itf/appendix/appendices}{%
2335   title/.store as code in cs=appendixpagename,%
2336   title/.append code=\ifx\appendixpagename\@empty \@dopage@ppfalse
2337   \else \@dopage@pptrue\fi,%
2338   title/.value required,%
2339   intoc/.is if=@dotoc@pp,%
2340   intoc/title/.store as code in cs=appendixtocname,%
2341   intoc/title/.append code=\ifx\appendixtocname\@empty \@dotoc@ppfalse
2342   \else \@dotoc@pptrue\fi,%
2343   intoc/title/.default=\pgfkeysvalueof{/itf/appendix/appendices/title},%
2344   section/.is enum={section,chapter},%
2345   section/.append code=%
2346     \ifcase\pgfkeysvalueof{/itf/appendix/appendices/section/.number}\relax
2347     \@chapter@ppfalse\or\@chapter@pptrue\fi
2348     \if@chapter@pp\ifundef\chapter{%
2349       \PackageError{interfaces-appendix}
2350       {You asked chapter-style appendices but your
2351       \MessageBreak current class does not provide chapters!}\@eha
2352       \pgfkeysalso{/itf/appendix/appendices/section}}{}%
2353     \fi},%
2354   chapter/.style={section=chapter},%
2355   chapter/.value forbidden,%
2356 }% keys for appendices environment

2357 </appendix>

```

21.17 † Interface ⇄ **embedfile** : **\embedthisfile**

```

2358 (*embedfile)
2359 \ProvidesPackage{interfaces-embedfile}
2360 [2010/06/20 v1.0 easy macros to embed files into pdf (FC)]
2361 \RequirePackage{interfaces-base}
2362 \itf@Check@Package{embedfile}
2363 \newrobustcmd*\embedthisfile{\@ifstar
2364   \embedthisfile@main
2365   \embedthisfile@all
2366 }% \embedthisfile
2367 \@onlypreamble\embedthisfile
2368 \def\embedthisfile@main{\embedfile{\jobname.tex}}
2369 \def\embedthisfile@all{%
2370   \embedthisfile@main
2371   \AfterPreamble{%
2372     \ltx@ifundefined{@listfiles}
2373     {\let\@listfiles\@empty}% to avoid: \global\let\@fileslist\relax in \document
2374     }{}%
2375   }%
2376   \gpreto\@enddocumenthook{%
2377     \let\do\itf@EmbedOne

```



```

2378 \expandafter\docsvlist\expandafter{\@filelist}%
2379 }%
2380 }% \embedthisfile@all
2381 \def\itf@EmbedOne#1{\itf@Embed@ne#1.tex\@nil}
2382 \def\itf@Embed@ne#1.tex#2\@nil{%
2383 \ltx@ifundefined{cp@#1}
2384 {}
2385 {\ifblank{#2}
2386 {}
2387 {\embedfile{#1.tex}}}%
2388 }% \itf@Embed@ne
2389 </embedfile>

```

21.18 † Interface **environ**

```

2390 <*environ>
2391 \ProvidesPackage{interfaces-environ}
2392 [2010/07/23 v1.0 ReNewEnviron = RenewEnviron (FC)]
2393 \RequirePackage{interfaces-base}
2394 \AfterPackage{environ}{
2395 \ifundef\ReNewEnviron
2396 {\let\ReNewEnviron\RenewEnviron}
2397 }%
2398 }%
2399 </environ>

```

21.19 † Interface **umrand**

```

2400 <*umrand>
2401 \ProvidesPackage{interfaces-umrand}
2402 [2010/07/23 v1.0 patch for umrand.sty ifnotempty (FC)]
2403 \RequirePackage{interfaces-base}
2404 \BeforePackage{umrand}{%
2405 \let\itf@newif\newif
2406 \def\itf@tp{\ifnotempty}%
2407 \def\newif#1{\def\x{#1}\ifx\x\itf@tp \expandafter\@gobble
2408 \else\expandafter\ltx@newif\fi{#1}}%
2409 }
2410 \AfterPackage{umrand}{%
2411 \let\newif\itf@newif
2412 \let\itf@newif\@undefined
2413 }
2414 </umrand>

```

21.20 † Interface **scrfile**

```

2415 <*scrfile>
2416 \ProvidesPackage{interfaces-scrfile}
2417 [2010/08/09 v1.0 scrfile hooks improved (FC)]
2418 \RequirePackage{interfaces-base}

```

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

`\@clsextension`

```

2419 \AfterEndPreamble{%
2420 \let\@pkgextension\ltx@pkgextension
2421 \let\@clsextension\ltx@clsextension
2422 \let\@ifpackageloaded\ltx@ifpackageloaded
2423 \let\@ifclassloaded\ltx@ifclassloaded
2424 }

```

`\scr@load@hook`

```

2425 \renewrobustcmd*\scr@load@hook[2]{%
2426 \csuse{#2-@#1}%

```

```

2427 \ifcsname#2-@#1-\string\keep\endcsname\else\global\csundef{#2-@#1}\fi
2428}% \scr@load@hook

```

\scr@set@load@hook (optimisation)

```

2429 \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

```

2430 \renewrobustcmd*\BeforeFile{\@ifstar
2431   {\itf@s@LoadFile{before}}
2432   {\itf@LoadFile{before}}}%
2433}% \BeforeFile
2434 \renewrobustcmd*\AfterFile{\@ifstar
2435   {\itf@s@LoadFile{after}}
2436   {\itf@LoadFile{after}}}%
2437}% \AfterFile

2438 \def\itf@s@LoadFile#1#2{%
2439   \global\cslet{#2-@#1-\string\keep}\@empty
2440   \scr@set@load@hook{#1}{#2}%
2441}% \itf@s@LoadFile
2442 \def\itf@LoadFile#1#2{%
2443   \global\csundef{#2-@#1-\string\keep}%
2444   \scr@set@load@hook{#1}{#2}%
2445}% \itf@LoadFile

```

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

```

2446 \renewcommand*\@input[1]{\IfFileExists{#1}
2447   {\scr@load@hook{before}{#1}%
2448    \@@input\@filef@und
2449    \scr@load@hook{after}{#1}}%
2450   {\typeout{No file #1.}}%
2451}% \@input

```

```

2452 </scrfile>

```

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

21.21.1 interfaces-pgfkeys

```

2453 <*pgfkeys>
2454 \ProvidesPackage{interfaces-pgfkeys}
2455   [2010/10/22 v1.1 pgfkeys handlers definitions (FC)]
2456 \RequirePackage{interfaces-base}
2457 \AtEndOfPackage{%
2458   \let\itf@pgfkeys@unpack\pgfkeys@unpack
2459   \let\itf@pgfkeys@case@three\pgfkeys@case@three
2460   \let\itf@pgfkeys@exp@call\pgfkeys@exp@call
2461   \let\itf@pgfkeysaddvalue\pgfkeysaddvalue
2462}% \AtEndOfPackage
2463 \def\itf@pgfkeys{%
2464   \global\let\pgfkeys@unpack\itf@pgfkeys@unpack
2465   \global\let\pgfkeys@case@three\itf@pgfkeys@case@three
2466   \global\let\pgfkeys@exp@call\itf@pgfkeys@exp@call
2467   \global\let\pgfkeysaddvalue\itf@pgfkeysaddvalue
2468}% \itf@pgfkeys
2469 \AtBeginDocument{%
2470   \itf@pgfkeys

```

```
2471 \let\itf@pgfkeys\relax
2472}% AtBeginDocument
```

Some new pgfkeys macros

`\pgfkeysmeaning`

```
2473 \providecommand*\pgfkeysmeaning[1]{%
2474   \pgfkeysifdefined{#1}
2475     {\expandafter\meaning\csname pgfk@#1\endcsname}
2476     {\meaning\undefined}%
2477}%
```

`\pgfkeysvalueof@unexpanded`

```
2478 \def\pgfkeysvalueof@unexpanded#1{%
2479   \unexpanded\expandafter\expandafter\expandafter{%
2480     \csname pgfk@#1\endcsname}%
2481}% \pgfkeysvalueof@unexpanded
```

Some pgfkeys macros redefinition/corrections

`\pgfkeysEsetvalue`

`\pgfkeysaddvalue` (bug correction)

`\pgfkeysEaddvalue`

```
2482 \newrobustcmd*\pgfkeysEsetvalue[2]{%
2483   \expandafter\edef\csname pgfk@#1\endcsname{#2}%
2484}% \pgfkeysEsetvalue
2485 \newrobustcmd*\pgfkeysaddvalue[3]{% \cspreappto{pgfk@#1}{#2}{#3} (bug in pgfkeys.code.tex)
2486   \edef\pgfkeys@global@temp{%
2487     \unexpanded{#2}%
2488     \ifcsname pgfk@#1\endcsname
2489       \unexpanded\expandafter\expandafter\expandafter{%
2490         \csname pgfk@#1\endcsname}%
2491     \fi
2492     \unexpanded{#3}}%
2493   \pgfkeyslet{#1}\pgfkeys@global@temp
2494}% \pgfkeysaddvalue
2495 \newrobustcmd*\pgfkeysEaddvalue[3]{% \csepreappto{pgfk@#1}{#2}{#3}
2496   \edef\pgfkeys@global@temp{%
2497     #2%
2498     \ifcsname pgfk@#1\endcsname
2499       \unexpanded\expandafter\expandafter\expandafter{%
2500         \csname pgfk@#1\endcsname}%
2501     \fi
2502     #3}%
2503   \pgfkeyslet{#1}\pgfkeys@global@temp
2504}% \pgfkeysaddvalue
```

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

```
2505%% missed a pair of braces around the argument in pgfkeys.code.tex (code could break if
2506\def\pgfkeys@exp@call#1{\pgfkeysalso{\pgfkeyscurrentpath={#1}}}
```

Handlers `/.blank`

```

2507%% The value is now checked by \itf@pgfkeys@default@blank@required:
2508 \def\pgfkeys@unpack#1=#2=#3\pgfkeys@stop{%
2509   \pgfkeys@spdef\pgfkeyscurrentkey{#1}%
2510   \edef\pgfkeyscurrentkey{\pgfkeyscurrentkey}%
2511   \ifx\pgfkeyscurrentkey\pgfkeys@empty
2512     % Skip
2513   \else%
2514     \pgfkeys@add@path@as@needed
2515     \pgfkeys@spdef\pgfkeyscurrentvalue{#2}%
2516     \itf@pgfkeys@default@blank@required % <FC>
2517                                     \pgfkeyscurrentkey % <FC>
2518                                     \pgfkeys@case@one % <FC>
2519   \fi
2520}% \pgfkeys@unpack

```

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

```

2521 \def\itf@pgfkeys@default@blank@required#1#2{%
2522   \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text% Hmm... no value
2523     \pgfkeysifdefined{#1/.@def}%
2524     {\pgfkeysgetvalue{#1/.@def}\pgfkeyscurrentvalue}
2525     {}% no default, so leave it
2526   \fi
2527   \ifx\pgfkeyscurrentvalue\@empty
2528     \pgfkeysifdefined{#1/.@blank}%
2529     {\pgfkeysgetvalue{#1/.@blank}\pgfkeyscurrentvalue}
2530     {}% no /.blank specified
2531   \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text% in case /.blank -> \pgfkeysnovalue
2532     \pgfkeysifdefined{#1/.@def}%
2533     {\pgfkeysgetvalue{#1/.@def}\pgfkeyscurrentvalue}
2534     {}% no default, so leave it
2535   \fi\fi
2536   \ifx\pgfkeyscurrentvalue\pgfkeysvaluerequired
2537     \pgfkeysvalueof{/errors/value required/.@cmd}\pgfkeyscurrentkey
2538                                           \pgfkeyscurrentvalue
2539                                           \pgfeov
2540   \else
2541     #2%
2542   \fi
2543}% \itf@pgfkeys@default@blank@required

```

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

```

2544%% To handle the cases of: /handler/.default= /handler/.blank= /handler/.value required,
2545%% it is necessary to redefine \pgfkeys@case@three
2546 \renewcommand*\pgfkeys@case@three{%
2547   \pgfkeys@split@path
2548   \pgfkeysifdefined{/handlers/\pgfkeyscurrentname/.@cmd}%
2549   {%
2550     \itf@pgfkeys@default@blank@required{/handlers/\pgfkeyscurrentname}%
2551     {\pgfkeysgetvalue{/handlers/\pgfkeyscurrentname/.@cmd}\pgfkeys@code
2552     \expandafter\pgfkeys@code\pgfkeyscurrentvalue\pgfeov}%
2553   }
2554   \pgfkeys@unknown % no handler, thus the key is unknown
2555}% \pgfkeys@case@three

```

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

```

2556 \pgfkeys{%
2557   /handlers/.alias/.code=\itf@pgfkeys@ifaliasexists{#1}{%
2558     \pgfkeysedef\pgfkeyscurrentpath{\noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}}%
2559     /handlers/.alias/.value required,%
2560   /handlers/.blank/.code=\pgfkeyssetvalue{\pgfkeyscurrentpath/.@blank}{#1},%
2561   /handlers/.blank/.default=\pgfkeysnovalue,%
2562 }% /.alias and /.blank
2563 \def\itf@pgfkeys@ifaliasexists#1#2{%
2564   \pgfkeysifdefined{\pgfkeysdefaultpath#1/.@cmd}{#2}%
2565   {\pgfkeysifdefined{\pgfkeysdefaultpath#1}%
2566     {#2}%
2567     {\itf@pgfkeys@aliaserror{#1}}}%
2568 }% \itf@pgfkeys@ifaliasexists
2569 \def\itf@pgfkeys@aliaserror#1{%
2570   \PackageError{interfaces-pgfkeys}%
2571     {You want the key \pgfkeyscurrentpath\space
2572     \MessageBreak to be an alias for \pgfkeysdefaultpath#1
2573     \MessageBreak but the key \pgfkeysdefaultpath#1
2574     \MessageBreak is not defined!}\@eha\@gobble
2575 }% \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).

```

2576 \pgfkeys{
2577   /handlers/.alias*/.code=\itf@pgfkeys@ifaliasexists{#1}{%
2578     \pgfkeysedef\pgfkeyscurrentpath{\noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}}%
2579     \@tfor\x:=+-%\do{%
2580       \pgfkeysifdefined{\pgfkeysdefaultpath#1\x/.@cmd}%
2581       {\pgfkeysedef{\pgfkeyscurrentpath\x}{%
2582         \noexpand\pgfkeysalso{\pgfkeysdefaultpath#1\x={##1}}}%
2583       }%
2584     }},%
2585   /handlers/.alias*/.value required,%
2586 }% /.alias*

```

Handlers definition: `/.is private`

```

2587 \pgfkeys{%
2588   /handlers/.is private/.code=\pgfkeysdef\pgfkeyscurrentpath{%
2589     \PackageError{#1}{Key \pgfkeyscurrentkey\space is private
2590     \MessageBreak and should not be modified}\@ehd},%
2591   /handlers/.is private to class/.code=\pgfkeysdef\pgfkeyscurrentpath{%
2592     \ClassError{#1}{Key \pgfkeyscurrentkey\space is private
2593     \MessageBreak and should not be modified}\@ehd}%
2594 }% /.is private

```

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

```

2595 \def\itf@pgfkeys@choice[#1]#2#3{%
2596   \pgfkeys{\pgfkeyscurrentpath/.cd,%
2597     .is choice,%
2598     #2/.code={#3},%
2599     .default/.expanded=\@firstoftwo#1}%
2600 }% \itf@pgfkeys@choice
2601 \pgfkeys{%
2602   /handlers/.true/.code=\itf@pgfkeys@choice[{true}{false}]{true}{#1},%
2603   /handlers/.false/.code=\itf@pgfkeys@choice[{true}{false}]{false}{#1},%
2604   /handlers/.on/.code=\itf@pgfkeys@choice[{on}{off}]{on}{#1},%
2605   /handlers/.off/.code=\itf@pgfkeys@choice[{on}{off}]{off}{#1},%
2606 }% /.true /.false

```

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

```

2607 \pgfkeys{%
2608   /handlers/.store in cs/.code=\pgfkeysdef\pgfkeyscurrentpath{\csdef{#1}{##1}},%
2609   /handlers/.estore in cs/.code=\pgfkeysdef\pgfkeyscurrentpath{\csedef{#1}{##1}},%
2610 }% /.store in cs

```

Handler definition: `/.store as code`

```

2611 %% /.store as code handlers
2612 \pgfkeys{%
2613   /handlers/.store as code/.code={%
2614     \pgfkeysedef\pgfkeyscurrentpath{%
2615       \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
2616     \pgfkeyslet\pgfkeyscurrentpath\@empty
2617     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\@empty
2618     \@tfor\x:=#1\do{%
2619       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2620         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
2621           \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
2622           \noexpand\unexpanded{##1}}}%
2623       \pgfkeyslet{\pgfkeyscurrentpath\x/.@def}\@empty
2624     }},%
2625   /handlers/.store as code/.default=*,%
2626 }% /.store as code

```

Handlers definition: `/.estore as code`

```

2627 \pgfkeys{%
2628   /handlers/.estore as code/.code={%
2629     \pgfkeysedef\pgfkeyscurrentpath{%
2630       \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{##1}}%
2631     \pgfkeyslet\pgfkeyscurrentpath\@empty
2632     \@tfor\x:=#1\do{%
2633       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2634         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
2635           \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
2636           \noexpand\unexpanded{##1}}}%
2637       }},%
2638   /handlers/.estore as code/.default=*,%
2639 }% /.estore as code

```


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

```

2640 \pgfkeys{%
2641   /handlers/.store as code in cs/.code={%
2642     \pgfkeysdef\pgfkeyscurrentpath{%
2643       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2644         \else\csdef{#1}{##1}\fi}%
2645     \@tfor\x:=*+\do{%
2646       \pgfkeysedef\pgfkeyscurrentpath\x}{%
2647         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
2648           \noexpand\ifcsname#1\endcsname
2649             \noexpand\expandafter\noexpand\expandonce
2650               \noexpand\csname#1\endcsname
2651             \noexpand\fi
2652           \noexpand\unexpanded{##1}}}%
2653       \pgfkeyslet{\pgfkeyscurrentpath\x/.@def}\@empty
2654     }},%
2655   /handlers/.store as code in cs/.value required,%
2656 }% /.store as code in cs
2657 \pgfkeys{%
2658   /handlers/.estore as code in cs/.code={%
2659     \pgfkeysdef\pgfkeyscurrentpath{%
2660       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2661         \else\csedef{#1}{##1}\fi}%
2662     \@tfor\x:=*+\do{%
2663       \pgfkeysedef\pgfkeyscurrentpath\x}{%
2664         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
2665           \noexpand\ifcsname#1\endcsname
2666             \noexpand\expandafter\noexpand\expandonce
2667               \noexpand\csname#1\endcsname
2668             \noexpand\fi
2669           \noexpand\unexpanded{##1}}}%
2670       }},% default if \pgfkeysnovalue which expands to \@empty
2671   /handlers/.estore as code in cs/.value required,%
2672 }% /.estore as code in cs

```

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

```

2673 \protected\def\itf@pgfk@color#1#2\@nil{%
2674   \ifx\@let@token\color
2675     \def\pgfkeyscurrentvalue{#2}%
2676   \else\ifx\@let@token\normalcolor
2677     \def\pgfkeyscurrentvalue{default}%
2678   \else\ifx\@let@token\pgfkeysnovalue
2679     \def\pgfkeyscurrentvalue{default}%
2680   \fi\fi\fi
2681 }% \itf@pgfk@color
2682 \pgfkeys{%
2683   /handlers/.store as color/.code=%
2684     \pgfkeysedef\pgfkeyscurrentpath{%
2685       \futurelet\noexpand\@let@token\itf@pgfk@color
2686         ##1\noexpand\pgfkeysnovalue\noexpand\@nil
2687       \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
2688 }% /.store as color

```

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

```

2689 \pgfkeys{%
2690   /handlers/.store as color in cs/.code=%
2691   \pgfkeysedef\pgfkeyscurrentpath{%
2692     \futurelet\noexpand\@let@token\itf@pgfk@color
2693     ##1\noexpand\pgfkeysnovalue\noexpand\@nil
2694     \cslet{#1}\pgfkeyscurrentvalue}%
2695 }% /.store as color in cs

```

Handlers definition: `/.store as color command`

```

2696 \protected\def\itf@pgfk@colorcmd#1\@nil{%
2697   \ifcat @\noexpand\@let@token
2698     \def\pgfkeyscurrentvalue{\color{#1}}%
2699   \else\ifcat 0\noexpand\@let@token
2700     \def\pgfkeyscurrentvalue{\color{#1}}%
2701   \else\ifx\@let@token\@nil
2702     \let\pgfkeyscurrentvalue\normalcolor
2703   \fi\fi\fi
2704 }% \itf@pgfk@colorcmd
2705 \pgfkeys{%
2706   /handlers/.store as color command/.code=%
2707   \pgfkeysedef\pgfkeyscurrentpath{%
2708     \futurelet\noexpand\@let@token\itf@pgfk@colorcmd
2709     ##1\noexpand\@nil
2710     \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
2711   \pgfkeyssetvalue{\pgfkeyscurrentpath/.@def}\normalcolor
2712 }% /.store as color command

```

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

```

2713 \pgfkeys{%
2714   /handlers/.store as color command in cs/.code=%
2715   \pgfkeysedef\pgfkeyscurrentpath{%
2716     \futurelet\noexpand\@let@token\itf@pgfk@colorcmd
2717     ##1\noexpand\@nil
2718     \cslet{#1}\noexpand\pgfkeyscurrentvalue}%
2719   \pgfkeyssetvalue{\pgfkeyscurrentpath/.@def}\normalcolor
2720 }% /.store as color command in cs

```

Handlers definition `/.store as number`

```

2721 \pgfkeys{%
2722   /handlers/.store as number/.code={%
2723     \pgfkeysedef\pgfkeyscurrentpath{%
2724       \normalcolor\pgfkeyssetvalue{\pgfkeyscurrentpath}{\numexpr##1\relax}}%
2725     \pgfkeyslet\pgfkeyscurrentpath\z@
2726     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
2727     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
2728     \@tfor\x:=#1\do{%
2729       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2730         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
2731         \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
2732         \x
2733         \numexpr\unexpanded{##1}\relax}}%
2734       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
2735       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@\fi}%
2736     },%
2737   /handlers/.store as number/.default=+-*,%
2738 }% /.store as number

```

Handlers definition: `/.estore as number`

```

2739 \pgfkeys{%
2740   /handlers/.estore as number/.code={%
2741     \pgfkeysdef\pgfkeyscurrentpath{%
2742       \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\number\numexpr##1\relax}}%
2743     \pgfkeyslet\pgfkeyscurrentpath\z@
2744     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
2745     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
2746     \@tfor\x:=#1\do{%
2747       \pgfkeysdef{\pgfkeyscurrentpath\x}{%
2748         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2749           \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}%
2750           \x
2751           \numexpr##1\relax}}%
2752     }},%
2753   /handlers/.estore as number/.default=+-*,%
2754 }% /.estore as number

```

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

```

2755 \pgfkeys{%
2756   /handlers/.store as number in cs/.code={%
2757     \pgfkeysdef\pgfkeyscurrentpath{%
2758       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2759       \else\csdef{#1}{\numexpr##1\relax}\fi}%
2760     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
2761     \@tfor\x:=+-*\do{%
2762       \pgfkeysdef{\pgfkeyscurrentpath\x}{%
2763         \letcs\noexpand\pgfkeys@temp{%
2764           \noexpand\ifcsname#1\endcsname
2765             #1\noexpand\else z@\noexpand\fi}%
2766         \edef\noexpand\pgfkeys@temp{%
2767           \noexpand\pgfkeysalso{\pgfkeyscurrentpath={%
2768             \noexpand\expandonce\noexpand\pgfkeys@temp
2769             \x
2770             \numexpr\noexpand\unexpanded{##1}\relax}}%
2771         }\noexpand\pgfkeys@temp}%
2772     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else z@\fi}%
2773     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else z@\fi}%
2774   }},%
2775   /handlers/.store as number in cs/.value required,%
2776 }% /.store as number in cs

```

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

```

2777 \pgfkeys{%
2778   /handlers/.estore as number in cs/.code={%
2779     \pgfkeysdef\pgfkeyscurrentpath{%
2780       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2781       \else\csdef{#1}{\number\numexpr##1\relax}\fi}%
2782     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
2783     \@tfor\x:=+-*\do{%
2784       \pgfkeysdef{\pgfkeyscurrentpath\x}{%
2785         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2786           \noexpand\csname\noexpand\ifcsname#1\endcsname
2787             #1\noexpand\else z@\noexpand\fi\endcsname
2788           \x
2789           \numexpr##1\relax}}%
2790     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else z@\fi}%
2791     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else z@\fi}%
2792   }},%

```

```

2793 /handlers/.estore as number in cs/.value required,%
2794 }% /.estore as number in cs

```

Handlers definition: **/.defcounter**

```

2795 \pgfkeys{%
2796 /handlers/.defcounter/.code=\pgfkeysdef\pgfkeyscurrentpath{\defcounter{#1}{##1}}%
2797 }% /.defcounter

```

Handlers definition: **/.store as glue**

```

2798 \pgfkeys{%
2799 /handlers/.store as glue/.code={%
2800 \pgfkeysedef\pgfkeyscurrentpath{%
2801 \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\glueexpr##1\relax}}%
2802 \pgfkeyslet\pgfkeyscurrentpath\z@skip
2803 \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@skip
2804 \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
2805 \@tfor\x:=#1\do{%
2806 \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2807 \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
2808 \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
2809 \x
2810 \if*\x\noexpand\itf@units@scale##1!
2811 \else\glueexpr\noexpand\unexpanded{##1}\relax\fi}}%
2812 \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
2813 \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
2814 }},%
2815 /handlers/.store as glue/.default=+-*,%
2816 }% /.store as glue

```

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

```

2817 \pgfkeys{%
2818 /handlers/.estore as glue/.code={%
2819 \pgfkeysedef{\pgfkeyscurrentpath}{%
2820 \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{%
2821 \noexpand\the\glueexpr##1\relax}}%
2822 \pgfkeyslet\pgfkeyscurrentpath\z@skip
2823 \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@skip
2824 \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
2825 \@tfor\x:=#1\do{%
2826 \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2827 \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2828 \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\x
2829 \if*\x\noexpand\itf@units@scale##1!
2830 \else\glueexpr##1\relax\fi}}%
2831 \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
2832 \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
2833 }},%
2834 /handlers/.estore as glue/.default=+-*,%
2835 }% /.estore as glue

```

Handlers definition: `/.store as dimen`

```

2836 \pgfkeys{%
2837   /handlers/.store as dimen/.code={%
2838     \pgfkeysedef\pgfkeyscurrentpath{%
2839       \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\dimexpr##1\relax}}%
2840     \pgfkeyslet\pgfkeyscurrentpath\z@
2841     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
2842     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
2843     \@tfor\x:=#1\do{%
2844       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2845         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
2846           \noexpand\pgfkeysvalueof@unexpanded{\pgfkeyscurrentpath}%
2847           \x
2848           \if*\x \noexpand\itf@units@scale##1!
2849           \else\dimexpr\noexpand\unexpanded{##1}\relax\fi}}%
2850       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
2851       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
2852     }},%
2853   /handlers/.store as dimen/.default=+-*,%
2854 }% /.store as dimen

```

Handlers definition: `/.estore as dimen`

```

2855 \pgfkeys{%
2856   /handlers/.estore as dimen/.code={%
2857     \pgfkeysedef\pgfkeyscurrentpath{%
2858       \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{%
2859         \noexpand\the\dimexpr##1\relax}}%
2860     \pgfkeyslet\pgfkeyscurrentpath\z@
2861     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
2862     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
2863     \@tfor\x:=#1\do{%
2864       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2865         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2866           \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\x
2867           \if*\x \noexpand\itf@units@scale##1!
2868           \else \dimexpr##1\relax\fi}}%
2869       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
2870       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
2871     }},%
2872   /handlers/.estore as dimen/.default=+-*,%
2873 }% /.estore as dimen

```

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

```

2874 \pgfkeys{%
2875   /handlers/.store as glue in cs/.code={%
2876     \pgfkeysdef\pgfkeyscurrentpath{%
2877       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2878       \else\csdef{#1}{\glueexpr##1\relax}\fi}%
2879     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
2880     \@tfor\x:=+-*\do{%
2881       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2882         \letcs\noexpand\pgfkeys@temp{%
2883           \noexpand\ifcsname#1\endcsname
2884             #1\noexpand\else \z@skip\noexpand\fi}%
2885         \edef\noexpand\pgfkeys@temp{%
2886           \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2887             \noexpand\expandonce\noexpand\pgfkeys@temp
2888             \x
2889             \if*\x\noexpand\itf@units@scale##1!

```

```

2890         \else\glueexpr\noexpand\unexpanded{##1}\relax\fi}}%
2891     }\noexpand\pgfkeys@temp}%
2892     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*x\@ne\else\z@skip\fi}%
2893     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*x\@ne\else\z@skip\fi}%
2894     }},%
2895     /handlers/.store as glue in cs/.value required,%
2896 }% /.store as glue in cs

```

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

```

2897 \pgfkeys{%
2898     /handlers/.estore as glue in cs/.code={%
2899         \pgfkeysdef\pgfkeyscurrentpath{%
2900             \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2901                 \else\csedef{##1}{\the\glueexpr##1\relax}\fi}%
2902         \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
2903         \@tfor\x:=+-%\do{%
2904             \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2905                 \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2906                     \noexpand\csname\noexpand\ifcsname#1\endcsname
2907                         #1\noexpand\else \z@skip\noexpand\fi\endcsname
2908                     \x
2909                     \if*x\noexpand\itf@units@scale##1!
2910                     \else\glueexpr##1\relax\fi}}%
2911             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*x\@ne\else\z@skip\fi}%
2912             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*x\@ne\else\z@skip\fi}%
2913         }},%
2914     /handlers/.etore as glue in cs/.value required,%
2915 }% /.estore as glue in cs

```

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

```

2916 \pgfkeys{%
2917     /handlers/.store in skip cs/.code={%
2918         \pgfkeysdef\pgfkeyscurrentpath{%
2919             \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2920                 \else\csname#1\endcsname\glueexpr##1\relax\fi}%
2921         \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
2922         \@tfor\x:=+-%\do{%
2923             \pgfkeysedef{\pgfkeyscurrentpath\x}{%
2924                 \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2925                     \csname#1\endcsname % should have been defined as a skip (not expandable)
2926                     \x
2927                     \if*x\noexpand\itf@units@scale##1!
2928                     \else\glueexpr##1\relax\fi}}%
2929             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*x\@ne\else\z@skip\fi}%
2930             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*x\@ne\else\z@skip\fi}%
2931         }},%
2932     /handlers/.store in skip cs/.value required,%
2933 }% /.store in skip cs

```


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

```

2934 \pgfkeys{%
2935   /handlers/.store as dimen in cs/.code={%
2936     \pgfkeysdef\pgfkeyscurrentpath{%
2937       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2938       \else\csdef{#1}{\dimexpr##1\relax}\fi}%
2939   \pgfkeyslet\pgfkeyscurrentpath/.@blank}\z@
2940   \@tfor\x:=+*\do{%
2941     \pgfkeysedef\pgfkeyscurrentpath\x{%
2942       \letcs\noexpand\pgfkeys@temp{%
2943         \ifcsname#1\endcsname#1\else \z@\fi}%
2944       \edef\pgfkeys@temp{%
2945         \noexpand\pgfkeysalso{\pgfkeyscurrentpath={%
2946           \noexpand\expandonce\noexpand\pgfkeys@temp
2947           \x
2948           \if*\x\noexpand\itf@units@scale##1!
2949           \else\dimexpr\noexpand\unexpanded{##1}\fi}}%
2950         }\noexpand\pgfkeys@temp}%
2951       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
2952       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
2953     }},%
2954   /handlers/.store as dimen in cs/.value required,%
2955 }% /.store as dimen in cs {csname}{+*}

```

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

```

2956 \pgfkeys{%
2957   /handlers/.estore as dimen in cs/.code={%
2958     \pgfkeysdef\pgfkeyscurrentpath{%
2959       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2960       \else \@defaultunits\itf@sk ##1pt\relax\@nnil
2961       \csdef{#1}{\the\dimexpr\itf@sk\relax}\fi}%
2962   \pgfkeyslet\pgfkeyscurrentpath/.@blank}\z@
2963   \@tfor\x:=+*\do{%
2964     \pgfkeysedef\pgfkeyscurrentpath\x{%
2965       \noexpand\ifcsname#1\endcsname
2966       \noexpand\@defaultunits\itf@sk
2967       \noexpand\csname#1\endcsname pt\relax\noexpand\@nnil
2968       \noexpand\else\itf@sk\z@\noexpand\fi
2969       \if*\x
2970       \itf@sk\dimexpr\itf@sk*\noexpand\itf@units@scale##1!\relax
2971       \else
2972       \noexpand\@defaultunits\@tempdima ##1pt\relax\noexpand\@nnil
2973       \advance\itf@sk\x\@tempdima
2974       \fi
2975       \noexpand\pgfkeysalso{\pgfkeyscurrentpath=\itf@sk}}%
2976     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
2977     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
2978   }},%
2979   /handlers/.estore as dimen in cs/.value required,%
2980 }% /.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.

```

2981 \pgfkeys{%
2982   /handlers/.store in dimen cs/.code={%
2983     \pgfkeysdef\pgfkeyscurrentpath{%
2984       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2985       \else\csname#1\endcsname\dimexpr##1\relax\fi}%
2986     \pgfkeyslet{/pgfkeyscurrentpath/.@blank}\z@
2987     \@tfor\x:=+-%\do{%
2988       \pgfkeysedef\pgfkeyscurrentpath\x}{%
2989         \if*\x \else
2990         \noexpand\@defaultunits\itf@sk ##1pt\relax\noexpand\@nnil
2991         \fi
2992         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
2993           \csname#1\endcsname % should have been defined as a dimen (not expandable)
2994           \x
2995           \if*\x \noexpand\itf@units@scale##1!
2996           \else\itf@sk\fi}}%
2997     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x\@ne\else\z@\fi}%
2998     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x\@ne\else\z@\fi}%
2999   },%
3000   /handlers/.store in dimen cs/.value required,%
3001 }% /.store in dimen cs

```

Handlers definition: `/.is multiple choice`

```

3002 \pgfkeys{%
3003   /handlers/.is multiple choice/.code=%
3004     \pgfkeys{\pgfkeyscurrentpath/.cd,%
3005       .unknown/.style={/errors/unknown choice value=\pgfkeyscurrentkey\pgfkeyscurrentvalue}%
3006       .code=\expandafter\pgfkeys\expandafter{\pgfkeyscurrentkey/.cd,####1}%
3007     },%
3008   /handlers/.is multiple choice/.value forbidden,%
3009 }% /.is multiple choice

```

Handlers definition: `/.is enum`

```

3010 \pgfkeys{%
3011   /handlers/.is enum/.code={%
3012     \itf@cnt\z@
3013     \@for\x:=\pgfkeyscurrentvalue\do{%
3014       \pgfkeysedef\pgfkeyscurrentpath/\x}{%
3015         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\x}%
3016         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath/.number}{\the\itf@cnt}}%
3017       \ifnum\itf@cnt=\z@
3018         \pgfkeyslet{\pgfkeyscurrentpath/.@def}\x
3019         \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\x
3020       \fi
3021       \advance\itf@cnt\@ne
3022     }%
3023     \pgfkeysalso{\pgfkeyscurrentpath/.is choice}},%
3024   /handlers/.is enum/.value required,%
3025 }% /.is enum

```

3026 </pgfkeys>

21.22 † Interface ⇄ etoolbox - etoolbox-like commands

```

3027 \*etoolbox
3028 \ProvidesPackage{interfaces-etoolbox}
3029 [2010/06/10 some more etoolbox-like commands (FC)]
3030 \RequirePackage{interfaces-base}
3031 \begingroup\catcode'\&=3
3032 \long\gdef\etb@doitem#1&\do{#1}}
3033 \endgroup

\preappto
\epreappto
\gpreappto
\xpreappto
3034 \providerobustcmd\preappto[3]{%
3035 \edef#1{\unexpanded{#2}\ifdefined#1\expandonce#1\fi\unexpanded{#3}}%
3036 }% \preappto
3037 \providerobustcmd\epreappto[3]{%
3038 \edef#1{#2\ifdefined#1\expandonce#1\fi#3}%
3039 }% \epreappto
3040 \providerobustcmd*\gpreappto{\global\preappto}
3041 \providerobustcmd*\xpreappto{\global\epreappto}

\cspreappto
\csepreappto
\csgpreappto
\csxpreappto
3042 \providerobustcmd*\cspreappto[1]{%
3043 \csname itf@ifcsname#1\endcsname
3044 preappto\else defpreapp\fi\expandafter\endcsname
3045 \csname#1\endcsname
3046 }% \cspreappto
3047 \long\def\itf@preappto#1#2#3{\edef#1{#2\expandonce#1#3}}
3048 \long\def\itf@defpreapp#1#2#3{\edef#1{#2#3}}
3049 \providerobustcmd*\csepreappto[1]{%
3050 \csname itf@ifcsname#1\endcsname
3051 preappto\else defpreapp\fi\expandafter\endcsname
3052 \csname#1\endcsname
3053 }% \csepreappto
3054 \providerobustcmd*\csgpreappto{\global\cspreappto}
3055 \providerobustcmd*\csxpreappto{\global\csepreappto}

\removefrom
\eremovefrom
\gremovefrom
\xremovefrom
3056 \providerobustcmd\removefrom[2]{\in@false
3057 \long\def\itf@tp##1\itf@tp{%
3058 \in@true
3059 \edef#1{\unexpanded{##1}}%
3060 }\expandafter\itf@removefrom\expandafter{#1}{#2}%
3061 }% \removefrom
3062 \providerobustcmd\eremovefrom[2]{\begingroup
3063 \edef\x{\endgroup\removefrom\noexpand#1{#2}%
3064 }\x
3065 }% \eremovefrom
3066 \providerobustcmd\gremovefrom[2]{%
3067 \removefrom{#1}{#2}%
3068 \global\let#1=#1%
3069 }% \gremovefrom
3070 \providerobustcmd\xremovefrom[2]{%
3071 \eremovefrom{#1}{#2}%
3072 \global\let#1=#1%
3073 }% \xremovefrom

```

```
\itf@removefrom
```

```
3074 \long\def\itf@removefrom#1#2{% removes #2 (code) from #1 (hook)
3075   \def\in@@##1#2##2\in@@{\itf@tp##1##2\in@@}%
3076   \def\in@@##1\in@@##2\in@@{%
3077     \ifx\in@@##2\else##1\fi
3078   }\expandafter\in@@\in@@#1\itf@tp\in@@##2\in@@\in@@
3079 }% \itf@removefrom
```

```
\removefromstart
```

```
\eremovefromstart
```

```
\gremovefromstart
```

```
\xremovefromstart
```

```
3080 \newrobustcmd*\removefromstart[2]{\in@false
3081   \def\itf@tp##1\itf@tp{%
3082     \in@true
3083     \edef#1{\unexpanded{##1}}%
3084   }\expandafter\itf@removefromstart\expandafter{#1}{#2}%
3085 }% \removefromstart
3086 \providerobustcmd*\eremovefromstart[2]{\begingroup
3087   \edef\x{\endgroup\removefromstart\noexpand#1{#2}%
3088   }\x
3089 }% \eremovefromstart
3090 \providerobustcmd*\gremovefromstart[2]{%
3091   \removefromstart{#1}{#2}%
3092   \global\let#1=#1%
3093 }% \gremovefromstart
3094 \providerobustcmd*\xremovefromstart[2]{%
3095   \eremovefromstart{#1}{#2}%
3096   \global\let#1=#1%
3097 }% \xremovefromstart
```

```
\itf@removefromstart
```

```
3098 \def\itf@removefromstart#1#2{% removes #2 (code) from #1 (hook)
3099   \def\in@@##1#2##2\in@@{##1\in@\itf@tp##2\in@@}%
3100   \def\in@@##1\in@@##2\in@@{%
3101     \ifx\in@@##1\fi
3102   }\expandafter\in@@\in@@#1\itf@tp\in@@##2\in@@
3103 }% \itf@remove@fromstart
```

```
3104 </etoolbox>
```

22 History

[2010/10/20 v1.0]

- First version.

[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^AT_EX.

`interfaces` is organised as different modules which are loaded after the master package is loaded. Each module can be loaded separately.

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

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

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

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

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

+ some macros to make easier the use of ε -T_EX `\marks` registers (useful with `\sectionformat` and `\pagesetup`).

`interfaces` is mainly based on `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 | |
|--|--|
| <code>\!</code> | 444 |
| <code>\#</code> | 804, 816, 1783, 1791 |
| <code>\&</code> | 3031 |
| <code>\@input</code> | 2448 |
| <code>\@M</code> | 898, 900, 2195 |
| <code>\@addtoreset</code> | 657, 665, 703, 705 |
| <code>\@arabic</code> | 668 |
| <code>\@arrayright</code> | 70, 77, 84 |
| <code>\@break@tfor</code> | 188 |
| <code>\@cftmakeloftitle</code> | 1769, 1772, 1774 |
| <code>\@cftmakeloftitle_(patch)</code> | <u>1762</u> |
| <code>\@cftmakelottitle</code> | 1776, 1779, 1781 |
| <code>\@cftmakelottitle_(patch)</code> | <u>1762</u> |
| <code>\@cftmaketoctitle</code> | 1762, 1765, 1767 |
| <code>\@cftmaketoctitle_(patch)</code> | <u>1762</u> |
| <code>\@cftpnumoff</code> | 1627, 1744 |
| <code>\@cftpnumon</code> | 1626, 1743 |
| <code>\@chapter@ppfalse</code> | 2347 |
| <code>\@chapter@pptrue</code> | 2347 |
| <code>\@classz</code> | 93 |
| <code>\@clsextension</code> | <u>2419</u> |
| <code>\@currentHref</code> | 1814 |
| <code>\@currentlabel</code> | 1815 |
| <code>\@currentlabelname</code> | 1816 |
| <code>\@currenvir</code> | 1944 |
| <code>\@declaredoptions</code> | 25 |
| <code>\@defaultunits</code> | 145, 146, 194, 204, 851, 2960, 2966, 2972, 2990 |
| <code>\@dopage@ppfalse</code> | 2336 |
| <code>\@dopage@pptrue</code> | 2337 |
| <code>\@dotoc@ppfalse</code> | 2341 |
| <code>\@dotoc@pptrue</code> | 2342 |
| <code>\@eha</code> | 1202, 1567, 2038, 2186, 2351, 2574 |
| <code>\@ehd</code> | 62, 2031, 2034, 2043, 2047, 2051, 2054, 2282, 2590, 2593 |
| <code>\@elt</code> | 1141, 1142, 1143 |
| <code>\@enddocumenthook</code> | 2376 |
| <code>\@expandtwoargs</code> | 188, 199 |
| <code>\@filef@und</code> | 2448 |
| <code>\@filelist</code> | 2378 |
| <code>\@fileslist</code> | 2373 |
| <code>\@firstofone</code> | 1867, 2142 |
| <code>\@firstoftwo</code> | 2599 |
| <code>\@flushglue</code> | 359 |
| <code>\@fourthoffive</code> | 1044 |
| <code>\@height</code> | 1232, 1253 |
| <code>\@ifclassloaded</code> | 2423 |
| <code>\@ifnextchar</code> | 846 |
| <code>\@ifpackageloaded</code> | 2422 |
| <code>\@ifpackagewith</code> | 1751 |
| <code>\@ifundefined</code> | 222 |
| <code>\@ifundefinedcolor</code> | 218 |
| <code>\@input</code> | 2446, 2451 |
| <code>\@input_(patch)</code> | <u>2446</u> |
| <code>\@latex@warning</code> | 1021, 1038 |
| <code>\@let@token</code> | 152, 156, 158, 2674, 2676, 2678, 2685, 2692, 2697, 2699, 2701, 2708, 2716 |
| <code>\@listfiles</code> | 2373 |
| <code>\@mainaux</code> | 889, 1003, 1008 |
| <code>\@makeother</code> .. | 164, 183, 184, 185, 198, 444, 804, 1783 |
| <code>\@nil</code> | 134, 173, 179, 209, 210, 559, 561, 649, 703, 705, 1098, 1103, 1687, 1707, 1708, 1709, 1729, 1732, 1735, 1844, 1846, 1853, 2381, 2382, 2673, 2686, 2693, 2696, 2701, 2709, 2717 |
| <code>\@nnil</code> | 135, 136, 141, 145, 146, 151, 194, 204, 209, 211, 851, 1834, 1836, 1841, 1844, 1846, 1853, 1955, 1974, 2002, 2008, 2165, 2268, 2270, 2275, 2960, 2967, 2972, 2990 |
| <code>\@onelevel@sanitize</code> | 2028 |
| <code>\@onlypreamble</code> | 52, 2367 |
| <code>\@pkgextension</code> | <u>2419</u> |
| <code>\@preamble</code> | 70, 78, 85 |
| <code>\@optionlist</code> | 24 |
| <code>\@rightskip</code> | 358 |
| <code>\@specialpagetrue</code> | 1198 |
| <code>\@specialstyle</code> | 1198 |
| <code>\@stpel</code> | 1139, 1149 |
| <code>\@tempb</code> | 1482, 1484, 1485 |
| <code>\@tempcnta</code> | 296 |
| <code>\@temptokena</code> | 625, 626, 628 |
| <code>\@tfor</code> | 187, 2579, 2618, 2632, 2645, 2662, 2728, 2746, 2761, 2783, 2805, 2825, 2843, 2863, 2880, 2903, 2922, 2940, 2963, 2987 |
| <code>\@undefined</code> | 947, 948, 949, 950, 954, 955, 956, 960, 1469, 1820, 1992, 2412, 2476 |
| <code>\@whilenum</code> | 1877, 1886 |
| <code>\@bookmark_(patch)</code> | <u>888</u> |
| <code>\~</code> | 1858 |
| A | |
| <code>\a</code> | 2171, 2172, 2175, 2176, 2177 |
| <code>\active</code> | 1831, 1862 |
| <code>\addcontentsline</code> | 806, 807, 810, 812 |
| <code>\addcontentsline_(patch)</code> | <u>803</u> |
| <code>\AddLineBeginMainAux</code> | 537 |
| <code>\adl@arrayrestore</code> | 76, 83 |
| <code>\adl@endarray</code> | 76, 83 |
| <code>\afterassignment</code> | 147, 848 |
| <code>\AfterEndPreamble</code> | 2419 |
| <code>\AfterFile</code> | 544, 1619, <u>2430</u> |
| <code>\aftergroup</code> | 432, 958, 999, 1821 |
| <code>\AfterPackage</code> | 55, 68, 72, 89, 236, 316, 319, 326, 361, 425, 441, 799, 1055, 2141, 2394, 2410 |
| <code>\AfterPreamble</code> | 1499, 1502, 1505, 1508, 2371 |
| <code>\allpgfmodules</code> | <u>450</u> , 503, 520 |
| <code>\alltikzlibraries</code> | <u>451</u> , 504 |
| <code>\alltikzpackages</code> | <u>447</u> , 499, 502, 517 |
| <code>\appendices</code> | <u>2325</u> |
| <code>\appendixpagename</code> | 2336 |
| <code>\appendixtocname</code> | 2341 |
| <code>\appto</code> | 1228, 1248 |
| <code>\apptocmd</code> | 98, 117, 1784 |
| <code>\apptocs</code> | 739 |
| <code>\array@prepnext@tok</code> | 317 |

\arraybackslash 329, 332, 337,
340, 345, 348, 364, 367, 372, 375, 380, 383, 387, 389
\AtEndOfPackage 6, 7, 9, 33, 1830, 2457
\AtEndPreamble 442, 1613, 1614

B

\B 184
\b 2173, 2174, 2177
\badness 2195
\BeforeFile 540, 1615, 2430
\BeforePackage 38, 2404
\BKM@@ref 1046
\BKM@color 920, 930, 939, 951
\BKM@dest 1042, 1044
\BKM@page 1026
\BKM@startatroottrue 1809
\bookmark 8, 888, 891, 894, 1013, 1030, 1806
\BookmarkAtEnd 1009
\bookmarkget 967, 969, 977, 979
\bookmarksetup ... 7, 834, 836, 838, 926, 945, 972, 982
\botmarks 1457, 1476, 1478
\bottomtitles 7, 844
\bottomtitlespace 854
\BS@BKM@FontStyle 922, 934, 942, 952, 965, 975

C

\C 184, 1888, 1895, 1969, 2090
\c 1888, 1890, 2093
\catcode .. 541, 545, 816, 1791, 1829, 1831, 1862, 3031
\Centering 372, 375
\centering 337, 340
\cftdot 1534
\cftdotfill 1729
\cftdotsep 1531, 1733
\cftnodots 1533, 1734, 1735
\changeFont 3, 233, 261
\chapter 1155, 1692, 2348
\cl@section 1126
\ClassError 2592
\color 222, 223, 1223, 1243, 2674, 2698, 2700
\colorlet 219
\column 1888, 1969, 2000
\columnsep 1551, 1579
\columnseprule 1550, 1575
\columnseprulecolor 1552
\contentsname 1608, 1651, 1652, 1763, 1821
\count 145
\count@ 295, 299, 301, 306,
307, 314, 542, 546, 969, 971, 973, 979, 981, 983,
2165, 2166, 2209, 2213, 2215, 2216, 2217, 2221,
2222, 2224, 2226, 2227, 2228, 2229, 2232, 2233,
2235, 2237, 2238, 2239, 2240, 2241, 2244, 2246
\countdef 1869, 1870, 1871
\countvalues 12, 1833
\crr 70, 1972
\csappto 1183
\csdef 625, 626, 738,
743, 1083, 1184, 1735, 2608, 2644, 2759, 2878, 2938
\csedef 1207, 1732, 2609, 2661, 2781, 2901, 2961
\csepreappto 2495, 3042
\csgappto 2157, 2429
\csgdef 132, 2158
\csgpreappto 3042

\cslet 690,
691, 692, 737, 742, 1087, 1092, 2439, 2694, 2718
\csletcs 682, 1085, 1090, 2314, 2315
\cspreappto 2485, 3042
\csundef 2427, 2443
\csuse 755, 1137, 2426
\csxdef 666, 683
\csxpreappto 3042
\CT@cell@color 2143, 2144
\CT@cellc 2141
\CT@color 2143
\CT@end 70, 76
\currentgrouplevel 924, 957
\currentgrouptype 395, 400

D

\D 185
\DeclareOption 6, 7, 9
\default@color 223
\defcounter 2796
\define@key . 992, 993, 994, 995, 996, 997, 1028, 1047
\DefineVerbatimEnvironment 2292, 2294, 2296
\detokenize 602
\diaghead 2136
\dimen 2215, 2216, 2217,
2224, 2233, 2235, 2238, 2239, 2244, 2252, 2255
\dimen@ 2209, 2217, 2223, 2229, 2237
\dimendef 2226, 2227, 2228
\dimexpr 172,
178, 212, 869, 2194, 2261, 2263, 2265, 2839,
2849, 2859, 2868, 2938, 2949, 2961, 2970, 2985
\docsvlist 2378
\document 2373
\dump 114

E

\embedfile 2368, 2387
\embedthisfile 18, 2363, 2366, 2367
\embedthisfile@all 2365, 2369, 2380
\embedthisfile@main 2364, 2368, 2370
\endappendices 2326, 2328
\endarray 68
\endlinechar 542, 546
\endVerb 2308, 2310
\endVerbLines 2304, 2306
\enit@a 2161, 2167, 2169, 2171
\enit@b 2162, 2166, 2170, 2173
\epreappto 3034
\eremovefrom 3056
\eremovefromstart 3080
\etb@doitem 3032
\everycr 262, 265, 278, 279, 289
\ExecuteOptions 25
\expandonce 2118, 2119, 2122, 2123,
2649, 2666, 2768, 2887, 2946, 3035, 3038, 3047

F

\f@baselineskip 248, 251, 256
\f@linespread .. 248, 251, 254, 255, 256, 257, 258, 259
\f@size 248, 251, 256
\fancy@gb1 1170
\fancyhf 1167
\fancyhfoffset 1168
\figure 1698

[\firstmarks](#) 1452, 1465, 1475, 1478
[\fontdimen](#) 173, 179
[\fontencoding](#) 239
[\fontfamily](#) 241
[\fontseries](#) 243
[\fontshape](#) 245
[\footrule](#) 1245, 1246, 1247, 1248
[\footrulewidth](#) 1251, 1252, 1253
[\FPeval](#) 2022, 2051, 2072, 2073
[\FPresult](#) 2073, 2074, 2076
[\futurelet](#) 103, 156, 2685, 2692, 2708, 2716

G

[\G@refundefinedtrue](#) 1020, 1037
[\getpagerefnumber](#) 1013
[\getrefbykeydefault](#) 1026
[\getthebotmarks](#) 1475
[\getthefirstmarks](#) 1475
[\getthemarks](#) .. 11, 1465, 1475, 1476, 1477, 1481, 1482
[\getthetopmarks](#) 1475
[\glueexpr](#) 170, 171, 176, 2801, 2811,
 2821, 2830, 2878, 2890, 2901, 2910, 2920, 2928
[\gluestretchorder](#) 412, 417
[\gluetomu](#) 170
[\gpreappto](#) 3034
[\gpreto](#) 2376
[\gremovefrom](#) 3056
[\gremovefromstart](#) 3080

H

[\hbadness](#) 2193
[\hbs@bmstring](#) 1089, 1092
[\hbs@do](#) 683, 1062, 1066, 1068
[\hbs@do_\(patch\)](#) 1062
[\hbs@process](#) 1072, 1076, 1078
[\hbs@process_\(patch\)](#) 1072
[\hbs@seccmd](#) 1098
[\hbs@tocstring](#) 1084, 1087
[\headrule](#) 1225, 1226, 1227, 1228
[\headrulewidth](#) 1231, 1232, 1233
[\headwidth](#)
 .. 1232, 1253, 1415, 1416, 1417, 1420, 1421, 1424
[\hfilneg](#) 418
[\hfuzz](#) 2193
[\hrule](#) 1232, 1253
[\htruncadjust](#) 17, 2201
[\htruncwidth](#) 2267
[\Hy@AnchorName](#) 871, 872, 875
[\Hy@babelnormalise](#) 1029, 1048
[\Hy@raisedlink](#) 875
[\hyper@anchorend](#) 875
[\hyper@anchorstart](#) 875
[\HyperRaiseLinkHook](#) 865, 866, 867, 868
[\HyperRaiseLinkLength](#) 869

I

[\I](#) 185
[\if@cftsubfigopt](#) 1672, 1751
[\if@chapter@pp](#) 2348
[\if@fancyplain](#) 1231, 1251
[\ifblank](#) 150, 510, 562, 612, 649,
 701, 736, 741, 746, 757, 761, 765, 769, 773, 777,
 786, 862, 914, 1104, 1178, 1182, 1866, 2210, 2385
[\ifcase](#) 297, 973, 983, 1000, 1005, 2029, 2346

[\ifcat](#) 139, 2697, 2699
[\ifcsname](#) 42, 45, 269, 272,
 1128, 1432, 1438, 1451, 1456, 1461, 1467, 1604,
 1624, 2312, 2313, 2427, 2488, 2498, 2648, 2665,
 2764, 2786, 2883, 2906, 2943, 2965, 3043, 3050
[\ifcsundef](#) 577, 580, 616, 663, 670, 698, 1192, 1196, 2175
[\ifcsvoid](#) 755
[\ifdim](#) 396, 416, 1415, 2233, 2238, 2249
[\iffalse](#) 325
[\IfFileExists](#) 2446
[\ifin@](#) 188, 191, 201, 518, 521, 1951, 1954, 1959
[\ifitf@cellright](#) 293, 304
[\ifitf@celltranspose](#) 1855, 1899, 2116
[\ifitf@colortbl](#) 325, 326, 394, 410
[\ifmarksequal](#) 11, 1478
[\ifnotempty](#) 2406
[\ifnum](#) .. 166, 263, 280, 395, 400, 412, 417, 904, 905,
 946, 957, 971, 981, 1543, 1847, 1876, 1885, 1902,
 1913, 1924, 1935, 1975, 2003, 2195, 2272, 3017
[\iftrue](#) 326
[\ifttl@explicit](#) 625
[\ifundef](#) 1155, 2348, 2395
[\ignorespaces](#) 1946, 2194, 2214, 2234
[\immediate](#) 889, 1003, 1008
[\in@](#) 188, 199, 517, 520, 3077, 3078, 3099, 3101
[\in@@](#) 3075, 3078, 3099, 3102
[\in@@@](#) 3075, 3076, 3078, 3099, 3100, 3102
[\in@false](#) 1957, 3056, 3080
[\in@true](#) 3058, 3082
[\input](#) 513, 1056
[\InputIfFileExists](#) 618
[\interitem](#) 16, 2153
[\itf@cellleft](#) 266, 284
[\itf@cellright](#) 267, 285
[\itf@AfterPackage](#)
 65, 680, 686, 689, 747, 750, 805, 1595
[\itf@ampactive](#) 1856, 1859, 1918, 1940
[\itf@appendices@pp](#) 2325, 2327, 2331
[\itf@AtEnd](#) 1829, 1830
[\itf@BKM@pageref](#) 1015, 1018, 1027
[\itf@BKM@ref](#) 1032, 1035
[\itf@BKM@enable](#) 898,
 900, 901, 902, 903, 904, 905, 1000, 1001, 1005, 1007
[\itf@BKM@pageref](#) 1013
[\itf@BKM@ref](#) 1030
[\itf@BKM@restore](#) 958, 986, 990, 999
[\itf@BKM@write](#) 889, 998
[\itf@cell](#) 1872, 2103, 2107, 2112, 2115, 2127, 2128, 2129
[\itf@cell@C](#) 370, 377
[\itf@cell@c](#) 325
[\itf@cell@calc](#) 2057
[\itf@cell@formulaerror](#)
 .. 1984, 1985, 1987, 2010, 2019, 2022, 2027, 2056
[\itf@cell@FPcalc](#) 2023, 2057
[\itf@cell@getformula](#) 2008, 2013, 2026
[\itf@cell@getformula](#) 1989, 2007
[\itf@cell@getifdefined](#) 2078, 2095, 2096, 2097, 2098
[\itf@cell@init](#) 1878,
 1881, 1887, 1888, 1892, 1894, 1898, 1918, 1940
[\itf@cell@insert](#) 1919, 1941, 2078
[\itf@cell@J](#) 386, 391
[\itf@cell@j](#) 325
[\itf@cell@L](#) 362, 369

| | | | |
|--------------------------------------|--|--|---|
| <code>\itf@cell@l</code> | 325 | <code>\itf@hbs@resetlast</code> | 1127, 1132, 1134, 1135, 1136 |
| <code>\itf@cell@numcalc</code> | 2000, 2001, 2016, 2057 | <code>\itf@hbs@stpelt</code> | 1140, 1144, 1147 |
| <code>\itf@cell@pgfcalc</code> | 2020, 2057 | <code>\itf@hfil</code> | 406, 423, 427 |
| <code>\itf@cell@R</code> | 378, 385 | <code>\itf@hfill</code> | 407, 428 |
| <code>\itf@cell@r</code> | 325 | <code>\itf@hskip</code> | 408, 429 |
| <code>\itf@cell@rowcolkey</code> | 1949, 1964, 1967, 1970, 1973 | <code>\itf@htruncadjust</code> | 2202, 2204, 2262 |
| <code>\itf@cellcol</code> | 2098, 2102, 2108, 2111, 2119, 2122 | <code>\itf@htruncwidth</code> | 2268, 2270, 2275, 2277 |
| <code>\itf@cellfont</code> | 268, 276, 286, 320, 1966, 1996, 2100 | <code>\itf@ignorehfil</code> | 331, 339, 347, 366, 374, 382, 405 |
| <code>\itf@cellfontcol</code> | 2096, 2104, 2107, 2119, 2122 | <code>\itf@istruncated</code> | 2207, 2240, 2250, 2251 |
| <code>\itf@cellfontrow</code> | 2095, 2104, 2112, 2118, 2123 | <code>\itf@justify</code> | 352, 354, 357, 360 |
| <code>\itf@cellllalign</code> | 287, 300, 320, 328, 336, 344, 352, 363, 371, 379 | <code>\itf@last@from@next</code> | 1100, 1108, 1116 |
| <code>\itf@cellllalign</code> | 387 | <code>\itf@LoadFile</code> | 2432, 2436, 2442, 2445 |
| <code>\itf@celllleft</code> | 266, 276, 277, 284, 302, 320, 330, 332, 333, 338, 340, 341, 346, 348, 349, 352, 354, 355, 365, 367, 368, 373, 375, 376, 381, 383, 384, 387, 389, 390 | <code>\itf@LoadInterface</code> | 49, 53, 56 |
| <code>\itf@cellralign</code> | 288, 309, 320, 331, 339, 347, 353, 366, 374, 382, 388 | <code>\itf@makecell</code> | 1867, 1868, 2129 |
| <code>\itf@cellright</code> | 267, 285, 308, 320 | <code>\itf@marks@new</code> | 1433, 1439, 1445 |
| <code>\itf@cellrightfalse</code> | 305, 312 | <code>\itf@marks@warning</code> | 1449 |
| <code>\itf@cellrighttrue</code> | 298 | <code>\itf@message</code> | 1944, 1948 |
| <code>\itf@cellrow</code> | 2097, 2101, 2113, 2118, 2123 | <code>\itf@musik</code> | 127, 1530, 1531, 1731, 1732 |
| <code>\itf@cellskip</code> | 396, 405, 411, 412, 413 | <code>\itf@newif</code> | 2405, 2411, 2412 |
| <code>\itf@celltext</code> | 1968, 1995, 2105 | <code>\itf@nohfil</code> | 422, 424, 426, 432 |
| <code>\itf@celltransposefalse</code> | 1994 | <code>\itf@number</code> | 144, 1955, 1974, 2002, 2165 |
| <code>\itf@cft@label</code> | 1792 | <code>\itf@number@</code> | 147, 149, 155 |
| <code>\itf@cft@name</code> | 1687, 1707, 1708, 1709, 1729, 1732, 1735 | <code>\itf@number@@</code> | 151, 152, 156 |
| <code>\itf@cft@newlistof</code> | 1510, 1641, 1642, 1643, 1785 | <code>\itf@number@@@</code> | 156, 157, 161 |
| <code>\itf@cft@newsection</code> | 687, 1711, 1750, 1753, 1755, 1758, 1760 | <code>\itf@pagesetup</code> | 1176, 1178, 1187 |
| <code>\itf@cft@sec@name</code> | 1689, 1691 | <code>\itf@pagesetup</code> | 1172, 1173, 1175, 1177 |
| <code>\itf@cft@secname</code> | 1687, 1688, 1690, 1743, 1744 | <code>\itf@patch</code> | 103, 806, 888, 1062, 1072, 1107, 1762, 1769, 1776, 1784 |
| <code>\itf@cft@section@name</code> | 1704 | <code>\itf@patch@</code> | 103, 104, 109 |
| <code>\itf@cft@sectionname</code> | 1687 | <code>\itf@patch@boundary</code> | 110, 117 |
| <code>\itf@Check@Package</code> | 57, 64, 533, 860, 886, 1054, 1165, 1496, 1828, 2151, 2289, 2323, 2362 | <code>\itf@patch@inside</code> | 118, 125 |
| <code>\itf@cnt</code> | 129, 295, 296, 297, 1871, 1955, 3012, 3016, 3017, 3021 | <code>\itf@pgfk@color</code> | 2673, 2681, 2685, 2692 |
| <code>\itf@col</code> | 1869, 1885, 1886, 1900, 1902, 1903, 1904, 1909, 1932, 1934, 1935, 2000, 2002, 2003, 2004, 2096, 2098 | <code>\itf@pgfk@colorcmd</code> | 2696, 2704, 2708, 2716 |
| <code>\itf@colloop</code> | 1886, 1890, 1901, 1933 | <code>\itf@pgfkeys</code> | 232, 233, 439, 513, 534, 561, 887, 992, 993, 994, 995, 996, 997, 1166, 1180, 1185, 1497, 1499, 1502, 1505, 1508, 1832, 1873, 2152, 2290, 2324, 2463, 2468, 2470, 2471 |
| <code>\itf@cols</code> | 1890, 1896, 2005 | <code>\itf@pgfkeys@aliaserror</code> | 2567, 2569, 2575 |
| <code>\itf@countvalues</code> | 1834, 1836, 1840, 1842 | <code>\itf@pgfkeys@case@three</code> | 2459, 2465 |
| <code>\itf@defpreapp</code> | 3048 | <code>\itf@pgfkeys@choice</code> | 2595, 2600, 2602, 2603, 2604, 2605 |
| <code>\itf@diaghead</code> | 2117, 2121, 2133 | <code>\itf@pgfkeys@default@blank@required</code> | 2507, 2516, 2521, 2550 |
| <code>\itf@Embed@ne</code> | 2381, 2382, 2388 | <code>\itf@pgfkeys@exp@call</code> | 2460, 2466 |
| <code>\itf@EmbedOne</code> | 2377, 2381 | <code>\itf@pgfkeys@ifaliasexists</code> | 2557, 2563, 2568, 2577 |
| <code>\itf@endappendices@pp</code> | 2326, 2328, 2333 | <code>\itf@pgfkeys@unpack</code> | 2458, 2464 |
| <code>\itf@everycr@tok</code> | 262, 265, 289, 290 | <code>\itf@pgfkeys@addvalue</code> | 2461, 2467 |
| <code>\itf@exp@ndonceifcs</code> | 135, 136, 143 | <code>\itf@preappto</code> | 3047 |
| <code>\itf@expandonceifcs</code> | 135, 1977, 2005 | <code>\itf@preapptocmd</code> | 96, 106, 806 |
| <code>\itf@flush</code> | 331, 338, 339, 346, 366, 373, 374, 381, 393, 404 | <code>\itf@prepnext@tok</code> | 293 |
| <code>\itf@freespace</code> | 2226, 2230, 2243, 2244, 2248, 2249, 2252 | <code>\itf@rem@mu</code> | 163, 166 |
| <code>\itf@getthemarks</code> | 1468, 1472, 1474 | <code>\itf@remove@fromstart</code> | 3103 |
| <code>\itf@getv@lue</code> | 1844, 1846, 1852, 1854 | <code>\itf@removefrom</code> | 3060, 3074 |
| <code>\itf@getvalue</code> | 1843, 1883, 1896 | <code>\itf@removefromstart</code> | 3084, 3098 |
| <code>\itf@hbs@chkclass</code> | 1096, 1103, 1105 | <code>\itf@removehfil</code> | 328, 336, 344, 363, 371, 379, 405 |
| <code>\itf@hbs@next</code> | 1073, 1082 | <code>\itf@repeatcell</code> | 1863, 1865, 1947 |
| <code>\itf@hbs@reset</code> | 1133, 1138, 1141 | <code>\itf@row</code> | 1870, 1876, 1877, 1910, 1912, 1913, 1922, 1924, 1925, 1926, 1931, 1974, 1975, 1976, 1998, 2001, 2095, 2097 |
| | | <code>\itf@rowfont</code> | 263, 264 |
| | | <code>\itf@rowfont@reset</code> | 279, 282 |
| | | <code>\itf@rowloop</code> | 1877, 1880, 1911, 1923 |
| | | <code>\itf@rows</code> | 1880, 1883, 1977 |

| | |
|---|--|
| <code>\itf@sLoadFile</code> | 2431, 2435, 2438, 2441 |
| <code>\itf@scaledimen</code> | 2237, 2252, 2264, 2266 |
| <code>\itf@scalefactor</code> | 2263, 2265 |
| <code>\itf@sectionformat</code> | 559, 561 |
| <code>\itf@sectionformat</code> | 554, 555, 557 |
| <code>\itf@set@bottomtitle@space</code> | 848, 852, 854 |
| <code>\itf@setbottomtitlespace</code> | 847, 850, 853 |
| <code>\itf@setlist</code> | 2157, 2158, 2160, 2181 |
| <code>\itf@setlist@error</code> | 2176, 2182, 2187 |
| <code>\itf@sk</code> | 126, 848, 851, 854, 2960,
2961, 2966, 2968, 2970, 2973, 2975, 2990, 2996 |
| <code>\itf@stpelt</code> | 1143, 1145, 1148, 1149 |
| <code>\itf@strip@at</code> | 130, 603, 1063 |
| <code>\itf@strip@dot</code> | 130, 172, 178 |
| <code>\itf@strip@mu</code> | 162, 1531, 1732 |
| <code>\itf@sum</code> | 2228, 2241, 2252 |
| <code>\itf@tempdim</code> | 2227, 2237, 2238, 2239, 2243 |
| <code>\itf@thecr</code> | 1904, 1905, 1907, 1926, 1927, 1929 |
| <code>\itf@tk</code> | 128, 1413 |
| <code>\itf@tocloft</code> | 1631, 1763, 1770, 1777, 1792, 1796 |
| <code>\itf@tp</code> | 8, 10, 11,
12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 41,
42, 43, 45, 46, 49, 103, 105, 106, 111, 112, 113,
119, 120, 121, 137, 138, 186, 188, 615, 616, 618,
622, 713, 714, 715, 820, 829, 834, 837, 838, 967,
970, 971, 974, 977, 980, 981, 984, 1008, 1009,
1063, 1083, 1085, 1087, 1090, 1092, 1100, 1102,
1106, 1151, 1152, 1153, 1154, 1544, 1545, 1748,
1750, 1752, 1753, 1754, 1755, 1757, 1758, 1759,
1760, 1803, 1804, 1812, 1813, 1820, 1821, 2028,
2042, 2046, 2050, 2105, 2108, 2113, 2117, 2121,
2128, 2129, 2131, 2135, 2139, 2291, 2300, 2301,
2406, 2407, 3057, 3075, 3078, 3081, 3099, 3102 |
| <code>\itf@trunc@nosuchwidth</code> | 2271, 2278, 2283 |
| <code>\itf@trunc@width</code> | 2206, 2246, 2256, 2259, 2260, 2268 |
| <code>\itf@ttl@BKM</code> | 807, 817, 843 |
| <code>\itf@ttl@createsection</code> | 676, 699 |
| <code>\itf@ttl@get@format</code> | 598, 599, 601, 610 |
| <code>\itf@ttl@getformat</code> | 568 |
| <code>\itf@ttl@getspacing</code> | 568 |
| <code>\itf@ttl@newsection</code> | 576, 695 |
| <code>\itf@ttl@sectionformat</code> | 567 |
| <code>\itf@ttl@setkey</code> | 584,
588, 590, 591, 592, 593, 594, 605, 606, 607, 608, 609 |
| <code>\itf@ttl@setup</code> | 568 |
| <code>\itf@ttl@titleclass</code> | 643, 703, 705 |
| <code>\itf@ttl@titleclass@i</code> | 646, 647, 649, 661 |
| <code>\itf@ttl@titleclass@ii</code> | 657, 662, 675 |
| <code>\itf@ttl@update</code> | 571, 611 |
| <code>\itf@undefinedpagestyle</code> ... | 1193, 1197, 1200, 1203 |
| <code>\itf@units@mu</code> | 182, 1530, 1731 |
| <code>\itf@units@mutopt</code> | 169, 202 |
| <code>\itf@units@normalize</code> | 215 |
| <code>\itf@units@pt</code> | 197 |
| <code>\itf@units@pttomu</code> | 169, 192 |
| <code>\itf@units@scale</code> | 207, 2810, 2829,
2848, 2867, 2889, 2909, 2927, 2948, 2970, 2995 |
| <code>\itf@units@scale@normalize</code> | 209, 210 |
| <code>\itf@usetikzcode</code> | 506, 515 |
| J | |
| <code>\jobname</code> | 1613, 1615, 1619, 2368 |
| <code>\justifying</code> | 387, 389 |
| K | |
| <code>\keep</code> | 2427, 2439, 2443 |
| L | |
| <code>\l@</code> | 673 |
| <code>\l@newsection</code> | 673 |
| <code>\l@sup</code> | 673 |
| <code>\label</code> | 1817 |
| <code>\lastchapterbookmark</code> | 1158 |
| <code>\lastchaptertitle</code> | 1156 |
| <code>\lastchaptertoc</code> | 1157 |
| <code>\lastsection</code> | 1070, 1080, 1114, 1126 |
| <code>\lastskip</code> | 411, 416, 417 |
| <code>\LaTeX</code> | 46 |
| <code>\lccode</code> | 1858 |
| <code>\leftmargin</code> | 2154 |
| <code>\leftskip</code> | 358 |
| <code>\letcs</code> | 2305, 2306, 2309, 2310, 2763, 2882, 2942 |
| <code>\listfigurename</code> | 1663, 1664, 1770 |
| <code>\listofsetup</code> | 10, 1498, 1638, 1790 |
| <code>\listtablename</code> | 1669, 1670, 1777 |
| <code>\LoadInterface</code> | 6, 7, 9, 40 |
| <code>\lofsetup</code> | 1498, 1775 |
| <code>\lotsetup</code> | 1498, 1782 |
| <code>\lowercase</code> | 165, 187, 199, 271, 1858 |
| <code>\ltx@clsextension</code> | 2421 |
| <code>\ltx@ifclassloaded</code> | 2423 |
| <code>\ltx@ifpackageloaded</code> | 53, 57, 66, 73, 90, 217, 221, 1541, 2299, 2422 |
| <code>\ltx@ifundefined</code> | 2372, 2383 |
| <code>\ltx@leavevmode</code> | 874 |
| <code>\ltx@newif</code> | 2408 |
| <code>\ltx@pkgextension</code> | 2420 |
| <code>\ltx@secondoftwo</code> | 154 |
| <code>\ltx@stpelt</code> | 1139, 1146 |
| M | |
| <code>\M</code> | 164, 185, 198 |
| <code>\macro</code> | 1465 |
| <code>\makecell</code> | 1868 |
| <code>\marks</code> | 43, 1435, 1441 |
| <code>\marksthe</code> | 11, 748, 1431 |
| <code>\marksthecs</code> | 1437 |
| <code>\maxdimen</code> | 2193 |
| <code>\meaning</code> | 2475, 2476 |
| <code>\muexpr</code> | 177 |
| <code>\multicolsep</code> | 1548 |
| <code>\muskip</code> | 169, 182, 197 |
| <code>\mutoglu</code> | 177 |
| N | |
| <code>\N</code> | 185 |
| <code>\newcount</code> | 129 |
| <code>\newcounter</code> | 664 |
| <code>\newlist</code> | 2185 |
| <code>\newlistentry</code> | 1784, 1787, 1789 |
| <code>\newlistentry_(patch)</code> | 1762 |
| <code>\newmarks</code> | 1446 |
| <code>\newmuskip</code> | 127 |
| <code>\newrobustcmd</code> | 40,
53, 57, 65, 96, 233, 264, 508, 515, 553, 574, 611,
645, 846, 861, 1169, 1431, 1437, 1466, 1475,
1476, 1477, 1479, 1498, 1501, 1504, 1507, 1860,
1865, 2160, 2191, 2201, 2363, 2482, 2495, 3080 |

`\newskip` 126, 405
`\newtoks` 128, 262
`\noalign` 263, 276, 283
`\normalbaselineskip` 1270
`\normalcolor` 1584, 2676, 2702, 2711, 2719, 2724
`\number` 147, 212, 904,
 905, 1542, 1543, 1833, 2064, 2229, 2263, 2742, 2781
`\numexpr` 1833, 1847, 2064,
 2724, 2733, 2742, 2751, 2759, 2770, 2781, 2789

P

`\P` 183
`\p@` 1580, 2222, 2223, 2224
`\PackageError` 59, 619, 1566, 2030, 2033, 2036, 2040,
 2045, 2049, 2053, 2183, 2279, 2349, 2570, 2589
`\PackageInfo` 809, 890, 1065,
 1075, 1109, 1447, 1632, 1764, 1771, 1778, 1786
`\PackageWarning` 91, 811,
 893, 1067, 1077, 1111, 1635, 1766, 1773, 1780, 1788
`\pagesetup` 10, 1169, 1211, 1413
`\pagestyle` 1184, 1192, 1213, 1422
`\paragraph` 1696
`\parfillskip` 359
`\patchcmd` 105, 125, 888
`\pdfstringdefDisableCommands` 237
`\pgfeov` 1962, 2539, 2552
`\pgfkeys` 261,
 507, 745, 748, 911, 1204, 1594, 1611, 1706, 2556,
 2576, 2587, 2596, 2601, 2607, 2612, 2627, 2640,
 2657, 2682, 2689, 2705, 2713, 2721, 2739, 2755,
 2777, 2795, 2798, 2817, 2836, 2855, 2874, 2897,
 2916, 2934, 2956, 2981, 3002, 3004, 3006, 3010
`\pgfkeys@add@path@as@needed` 2514
`\pgfkeys@case@one` 2518
`\pgfkeys@case@three` .. 2459, 2465, 2545, 2546, 2555
`\pgfkeys@code` 2551, 2552
`\pgfkeys@empty` 2511
`\pgfkeys@exp@call` 2460, 2466, 2505
`\pgfkeys@global@temp` 2486, 2493, 2496, 2503
`\pgfkeys@spdef` 2509, 2515
`\pgfkeys@split@path` 2547
`\pgfkeys@stop` 2508
`\pgfkeys@temp` 2763, 2766, 2768, 2771,
 2882, 2885, 2887, 2891, 2942, 2944, 2946, 2950
`\pgfkeys@unknown` 2554
`\pgfkeys@unpack` 2458, 2464, 2508, 2520
`\pgfkeysaddvalue` 925, 1598, 1599,
 1600, 1601, 1602, 2461, 2467, 2485, 2494, 2504
`\pgfkeysaddvalue_(bug_correction)` 2482
`\pgfkeysalso` 988, 992,
 993, 994, 995, 996, 997, 1206, 1572, 1573, 1718,
 2352, 2506, 2558, 2578, 2582, 2620, 2634, 2647,
 2664, 2730, 2748, 2767, 2785, 2807, 2827, 2845,
 2865, 2886, 2905, 2924, 2945, 2975, 2992, 3023
`\pgfkeyscurrentkey`
 ... 1542, 1562, 1956, 1960, 1986, 1988, 2509,
 2510, 2511, 2517, 2537, 2589, 2592, 3005, 3006
`\pgfkeyscurrentname`
 ... 506, 790, 1609, 1950, 1952, 1955,
 1991, 1992, 2031, 2034, 2037, 2548, 2550, 2551
`\pgfkeyscurrentpath`
 1206, 1208, 1209, 1707, 1708, 1709, 2506, 2558,
 2560, 2571, 2578, 2581, 2588, 2591, 2596, 2608,

 2609, 2614, 2615, 2616, 2617, 2619, 2620, 2621,
 2623, 2629, 2630, 2631, 2633, 2634, 2635, 2642,
 2646, 2647, 2653, 2659, 2663, 2664, 2684, 2687,
 2691, 2707, 2710, 2711, 2715, 2719, 2723, 2724,
 2725, 2726, 2727, 2729, 2730, 2731, 2734, 2735,
 2741, 2742, 2743, 2744, 2745, 2747, 2748, 2749,
 2757, 2760, 2762, 2767, 2772, 2773, 2779, 2782,
 2784, 2785, 2790, 2791, 2796, 2800, 2801, 2802,
 2803, 2804, 2806, 2807, 2808, 2812, 2813, 2819,
 2820, 2822, 2823, 2824, 2826, 2827, 2828, 2831,
 2832, 2838, 2839, 2840, 2841, 2842, 2844, 2845,
 2846, 2850, 2851, 2857, 2858, 2860, 2861, 2862,
 2864, 2865, 2866, 2869, 2870, 2876, 2879, 2881,
 2886, 2892, 2893, 2899, 2902, 2904, 2905, 2911,
 2912, 2918, 2921, 2923, 2924, 2929, 2930, 2936,
 2939, 2941, 2945, 2951, 2952, 2958, 2962, 2964,
 2975, 2976, 2977, 2983, 2986, 2988, 2992, 2997,
 2998, 3004, 3014, 3015, 3016, 3018, 3019, 3023
`\pgfkeyscurrentvalue`
 ... 1225, 1245, 1956, 1961, 1984, 1985, 2515,
 2522, 2524, 2527, 2529, 2531, 2533, 2536, 2538,
 2552, 2615, 2643, 2660, 2675, 2677, 2679, 2687,
 2694, 2698, 2700, 2702, 2710, 2718, 2758, 2780,
 2877, 2900, 2919, 2937, 2959, 2984, 3005, 3013
`\pgfkeysdef`
 ... 2588, 2591, 2608, 2609, 2642, 2659, 2757,
 2779, 2796, 2876, 2899, 2918, 2936, 2958, 2983
`\pgfkeysdefaultpath`
 ... 2558, 2564, 2565, 2572, 2573, 2578, 2580, 2582
`\pgfkeysEaddvalue` 1603, 1609, 1990, 2482
`\pgfkeysedef` 2558, 2578,
 2581, 2614, 2619, 2629, 2633, 2646, 2663, 2684,
 2691, 2707, 2715, 2723, 2729, 2741, 2747, 2762,
 2784, 2800, 2806, 2819, 2826, 2838, 2844, 2857,
 2864, 2881, 2904, 2923, 2941, 2964, 2988, 3014
`\pgfkeysEsetvalue` 603, 790,
 924, 1542, 2482, 2630, 2734, 2735, 2772, 2773,
 2790, 2791, 2812, 2813, 2820, 2831, 2832, 2850,
 2851, 2858, 2869, 2870, 2892, 2893, 2911, 2912,
 2929, 2930, 2951, 2952, 2976, 2977, 2997, 2998
`\pgfkeysgetvalue` ... 615, 713, 837, 930, 934, 939,
 942, 1544, 1552, 1803, 1805, 1812, 1815, 1904,
 1905, 1926, 1927, 2080, 2524, 2529, 2533, 2551
`\pgfkeysifdefined` 575,
 712, 818, 835, 919, 921, 923, 929, 933, 938,
 941, 944, 987, 1800, 1903, 1925, 1986, 2079,
 2474, 2523, 2528, 2532, 2548, 2564, 2565, 2580
`\pgfkeyslet` 715, 758, 762, 766, 770,
 774, 778, 787, 831, 832, 833, 920, 922, 947, 948,
 949, 950, 954, 955, 956, 960, 1562, 1563, 1564,
 1956, 1988, 2493, 2503, 2615, 2616, 2617, 2623,
 2631, 2653, 2687, 2710, 2725, 2726, 2727, 2743,
 2744, 2745, 2760, 2782, 2802, 2803, 2804, 2822,
 2823, 2824, 2840, 2841, 2842, 2860, 2861, 2862,
 2879, 2902, 2921, 2939, 2962, 2986, 3018, 3019
`\pgfkeysmeaning` 2473
`\pgfkeysnovalue`
 ... 1521, 2531, 2561, 2670, 2678, 2686, 2693
`\pgfkeysnovalue@text`
 ... 1984, 2522, 2531, 2643, 2660,
 2758, 2780, 2877, 2900, 2919, 2937, 2959, 2984
`\pgfkeyssetvalue`
 ... 587, 728, 729, 759, 763, 767, 771, 775,

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|--|--------------------------------------|--|-------------------------------|---|------------------------------------|------------------------------------|--------------------------|---------------------------------------|-----------------------------|--|------------------------------|---|--|---------------------|---------------------------------|----------|--|-------------------|----------------------|----------|--|-------------------------------|-------------------------------|--------------------------------|--------------------------------|---------------------------------|---------------------------------|--|-----------------------|---|--------------------------------|---|------------------------------------|------------------------------|---|--|--------------------------|--------------------------------------|------------------------------------|---------------------------|---------------------------|----------|--|------------------|---|---|--|--------------------------------------|---------------------------|---------------------------------------|---|---|-------------------------------|-----------------------------|--------------------------|-----------------------|------------------------------|------------------------------|-------------------------------|--------------------------|----------|--|------------------|---------------------------|-----------------------|------------------------------|---------------------------|---------------------------------------|-----------------------------|-----------------------------------|-------------------------------|-----------------------------|-------------------------------|-----------------------|------------------------|--|----------------------------|-----------------------------|----------------------------------|--------------------------------------|--|-------------------------|--------------------------------|--------------------------|--------------------------------|-------------------------------|----------------------------|--------------------------------|---------------------------------|------------------------------|--|------------------------------------|-------------------------------|---------------------------------|--------------------------------|--------------------------|-------------------------|----------|--|-----------------------|---|-----------------------|---|----------------------------------|
| 779, 781, 783, 788, 1546, 1555, 1997, 1999, 2560,
2711, 2719, 2724, 2742, 2801, 2839, 3015, 3016 | \pgfkeysvalueof 585, 597, 629, 630,
631, 632, 633, 634, 637, 638, 639, 640, 641, 709,
719, 821, 822, 823, 824, 825, 826, 827, 828, 946,
1208, 1209, 1227, 1247, 1548, 1549, 1550, 1551,
1553, 1556, 1616, 1617, 1620, 1621, 1808, 1817,
1960, 2084, 2343, 2346, 2537, 2749, 2828, 2866 | \pgfkeysvalueof@unexpanded
. 2478, 2621, 2635, 2731, 2808, 2846 | \pgfkeysvaluerequired 2536 | \pgfmathparse 2019, 2047, 2067 | \pgfmathresult 2068 | \pgfqkeys 234,
238, 491, 511, 512, 563, 565, 569, 702, 717, 751,
899, 906, 907, 908, 910, 912, 915, 916, 937, 968,
978, 1180, 1185, 1188, 1211, 1212, 1499, 1502,
1505, 1508, 1511, 1596, 1625, 1645, 1660, 1666,
1673, 1680, 1712, 1746, 1874, 1965, 2330, 2334 | \plainfootrulewidth 1251 | \plainheadrulewidth 1231 | \preappto 3034 | \prepnext@tok 314, 317, 318 | \preto 1227, 1247 | \pretocmd
. 97, 117, 1062, 1072, 1107, 1630, 1762, 1769, 1776 | \ProcessOptions 27 | \protected
. 738, 739, 743, 2057, 2063, 2066, 2070, 2673, 2696 | \providerobustcmd
2153, 3034, 3037, 3040, 3041, 3042, 3049, 3054,
3055, 3056, 3062, 3066, 3070, 3086, 3090, 3094 | \ps@ 1212 | \ps@@fancy 1180, 1185 | Q | | \q 1844 | \quad 2229 | R | | \R 1878, 1882, 2085 | \r 1878, 1880, 2088 | \RaggedLeft 380, 383 | \raggedleft 345, 348 | \RaggedRight 364, 367 | \raggedright 329, 332 | \raisedhyperdef 7, 861, 1802 | \rem@pt 255 | \remove@to@nnil 140, 1838, 1848, 1850, 2025 | \removefrom 23, 3056 | \removefromstart 1950, 1952, 3080 | \ReNewEnviron 2395, 2396 | \RenewEnviron 2396 | \renewrobustcmd
. . 1192, 1196, 2156, 2425, 2429, 2430, 2434, 2485 | \repeatcell 12, 1855, 2006, 2038, 2041 | \rightskip 358 | \robustify 536, 1167, 1168 | \romannumeral 1844, 2177 | \row 1878, 2001 | \rowfont 3, 262 | S | | \S 184 | \scr@load@hook 2425, 2447, 2449 | \scr@set@load@hook 2429, 2440, 2444 | \scr@set@load@hook_(optimisation) 2429 | \sectionformat 5, 553, 836 | \selectfont 234 | \set@fontsize 248, 251, 256 | \setlist 16, 2156, 2159, 2185 | \setlist_(extension,star_form) 2156 | \skip 169, 182, 197 | \stretch 395, 397 | \strip@pt 2263 | \strut 1209 | \subparagraph 1697 | \subsubfigure 1699 | \subsubsection 1695 | \subtable 1701 | T | | \T 184 | \tab 1857, 1861 | \table 1700 | \textfont 173, 179 | \textheight 851 | \the@dimen 2208, 2246, 2255 | \thebotmarks 1450 | \thefirstmarks 11, 1450 | \thepage 1022, 1039 | \thetopmarks 1450 | \thispagestyle 1196 | \thr@@ 2010 | \tikzset 443 | \titleclass 536, 651, 654, 679 | \titleformat 536 | \titlespacing 536 | \tocloftpagestyle 1512 | \tocsetup 9, 9, 1498, 1768 | \toks 299,
301, 306, 307, 314, 2216, 2222, 2237, 2241, 2252 | \toksdef 1872 | \topmarks 1462, 1477 | \truncate 2196 | \TruncateMarker 2191 | \trunctext 17, 2191 | \ttl@ 1098, 1103 | \ttl@a 625, 626, 635 | \ttl@getkeys 613, 614 | \ttl@savetitle 551 | \ttl@select 1107, 1110, 1112 | \ttl@select_(patch) 1107 | \ttl@spacing@ii 636 | \ttl@runinifempty 548 | \ttl@append 532, 538 | \tw@ 412, 1987 | \typeout 2450 | U | | \U 164, 198 | \unexpanded
. . . 1435, 1442, 2136, 2137, 2138, 2479, 2487,
2489, 2492, 2499, 2622, 2636, 2652, 2669, 2733,
2770, 2811, 2849, 2890, 2949, 3035, 3059, 3083 | \unkern 411 | \unskip 416, 2194, 2214, 2234 | \usepgfmodule 503, 521 |
|---|---|--|--------------------------------------|--|-------------------------------|---|------------------------------------|------------------------------------|--------------------------|---------------------------------------|-----------------------------|--|------------------------------|---|--|---------------------|---------------------------------|----------|--|-------------------|----------------------|----------|--|-------------------------------|-------------------------------|--------------------------------|--------------------------------|---------------------------------|---------------------------------|--|-----------------------|---|--------------------------------|---|------------------------------------|------------------------------|---|--|--------------------------|--------------------------------------|------------------------------------|---------------------------|---------------------------|----------|--|------------------|---|---|--|--------------------------------------|---------------------------|---------------------------------------|---|---|-------------------------------|-----------------------------|--------------------------|-----------------------|------------------------------|------------------------------|-------------------------------|--------------------------|----------|--|------------------|---------------------------|-----------------------|------------------------------|---------------------------|---------------------------------------|-----------------------------|-----------------------------------|-------------------------------|-----------------------------|-------------------------------|-----------------------|------------------------|--|----------------------------|-----------------------------|----------------------------------|--------------------------------------|--|-------------------------|--------------------------------|--------------------------|--------------------------------|-------------------------------|----------------------------|--------------------------------|---------------------------------|------------------------------|--|------------------------------------|-------------------------------|---------------------------------|--------------------------------|--------------------------|-------------------------|----------|--|-----------------------|---|-----------------------|---|----------------------------------|

| | | | |
|------------------------------------|---------------------------|--------------------------------------|--|
| <code>\usetikz</code> | 5, 507, <u>508</u> , 2047 | X | |
| <code>\usetikzlibrary</code> | 493, 504, 522 | <code>\xpreappto</code> | <u>3034</u> |
| | | <code>\xremovefrom</code> | <u>3056</u> |
| | | <code>\xremovefromstart</code> | <u>3080</u> |
| | V | | |
| <code>\vbox</code> | 2072 | | |
| <code>\Verb</code> | 2308, 2309 | Y | |
| <code>\Verblines</code> | 2303, 2305 | <code>\y</code> | 585, 586 |
| <code>\vsize</code> | 2193 | | |
| <code>\vskip</code> | 1233, 1252, 1253 | Z | |
| | W | <code>\z@skip</code> | 358, 413, 2802, 2803, 2804,
2812, 2813, 2822, 2823, 2824, 2831, 2832, 2879,
2892, 2893, 2902, 2911, 2912, 2921, 2929, 2930 |
| <code>\write</code> | 889, 1003, 1008 | | |