

The **interfaces*** package

interfaces with keys for other packages

FC

2010/12/07 – version 2.5 – release
pgf v2.1 / pgf v2.0 compatible

Abstract

interfaces provides a small number of user friendly macros to access features implemented in frequently used packages. Most of the macros are 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{**interfaces-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=\marksthe{section}{lastsectiontoc},% interfaces-hypbmsec
    mark=\marksthe{section}{#1},% interfaces-marks
}
\sectionformat)subsection[hang]{
    label=, labelsep=,
    font=\bfseries,
    bookmark=italic,
    top*=1.2,% a 120% scaling
    bottom+=0pt plus4pt,% add to previous value
    mark=\markright{#1},
}
\sectionformat)subsubsection[hang]{
    class=straight[\subparagraph],
    shape=block,
    label=,
    top=3pt plus2pt, bottom=1pt,
    font=\bfseries\sffamily,
    bookmark=off,% or false or none
    break=\allowbreak,
    mark=\marksthe{MiniSection}{#1},% see interfaces-marks
}
\bottomtitles=.2\textheight
\tocsetup{
    depth=3, title=Contents, pagenumbers=on,
    pagestyle=plain,% first page style: tocloftpagestyle
    title/top=1.5ex plus.5ex,
    title/bottom=0pt,
    section/indent=1em,
    section/number/width=1.5em,
    section/font+\larger\bfseries,% append to previous setting
    section/leaders,section/dotsep,
    subsection/indent=2em,
    subsection/number/width=2.5em,
    subsubsection/font=\scriptsize,
    subsection/leaders,subsection/dotsep,% default values
    subsubsection/pagenumbers=off,% no page numbers
    subsubsection/dotsep=% no leader
    columns=2,% same as twocolumns
    columns/sep=.5em,
    columns/rule/width=1pt,columns/rule/color=blue,
    label=toc,% gives a label to the t.o.c.
    bookmark={text=Contents,bold},% interfaces-bookmark
}
\tableofcontents
\mainmatter \pagestyle{document}
```

This documentation is produced with the +DocStrip+ utility.

- To get the 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

|                                                                     |    |                                                   |    |
|---------------------------------------------------------------------|----|---------------------------------------------------|----|
| <b>1 Interface L<sup>A</sup>T<sub>E</sub>X</b>                      | 3  | <b>21 Implementation</b>                          | 28 |
| <b>2 Interface TikZ</b>                                             | 3  | 21.1 Interfaces                                   | 28 |
| <b>3 Interface titlesec: \sectionformat</b>                         | 4  | 21.1.1 Requirements                               |    |
| <b>4 Interface hyperref</b>                                         | 6  | 21.1.2 The package options / list of interfaces   |    |
| <b>5 Interface bookmark</b>                                         | 6  | <b>21.2 Interfaces-base</b>                       | 28 |
| <b>6 Interface hypbmsec</b>                                         | 7  | 21.2.1 Interfaces patches                         |    |
| <b>7 Interface tocloft: \tocsetup</b>                               | 8  | 21.2.2 Helper macros                              |    |
| <b>8 Interface fancyhdr: \pagesetup</b>                             | 10 | 21.2.3 Numbers                                    |    |
| <b>9 Interface marks</b>                                            | 11 | 21.2.4 Units and math units                       |    |
| <b>10 Interface makecell: \repeatcell</b>                           | 12 | 21.2.5 Default color                              |    |
| 10.1 Conversion to scaled points                                    | 13 | <b>21.3 Interface L<sup>A</sup>T<sub>E</sub>X</b> | 33 |
| 10.2 Variable length cell                                           | 13 | 21.3.1 \changefont                                |    |
| 10.3 Empty rows                                                     | 13 | <b>21.4 Interface TikZ</b>                        | 34 |
| 10.4 color palette                                                  | 14 | 21.4.1 Catcode verifications                      |    |
| 10.5 tabularcalc-like tables (with dcolumn package)<br>and pgfmath  | 14 | 21.4.2 List of pgf/TikZ packages and libraries    |    |
| 10.6 Double entry array: the Body Mass Index                        | 15 | 21.4.3 pgfkeys definitions for \usetikz           |    |
| <b>11 Interface enumitem</b>                                        | 16 | <b>21.5 Interface titlesec: \sectionformat</b>    | 37 |
| <b>12 Interface truncate</b>                                        | 18 | 21.5.1 titlesec modifications                     |    |
| <b>13 Interface appendix</b>                                        | 19 | 21.5.2 \sectionformat                             |    |
| <b>14 Interface embedfile: \embedthisfile</b>                       | 19 | 21.5.3 Keys for \sectionformat                    |    |
| <b>15 Interface environ</b>                                         | 19 | 21.5.4 \bottomtitles                              |    |
| <b>16 Interface umrand</b>                                          | 19 | <b>21.6 Interface hyperref</b>                    | 43 |
| <b>17 Interface pgfkeys</b>                                         | 20 | <b>21.7 Interface bookmark</b>                    | 43 |
| 17.1 /.blank                                                        | 21 | 21.7.1 star keys* definitions                     |    |
| 17.2 /.is private                                                   | 21 | 21.7.2 ref and pageref keys                       |    |
| 17.3 /.store in cs, /.estore in cs                                  | 22 | <b>21.8 Interface hypbmsec</b>                    | 46 |
| 17.4 Boolean keys: /.true, /.false, /.on /.off                      | 22 | 21.8.1 interfaces-hypbmsec                        |    |
| 17.5 /.is multiple choice                                           | 22 | <b>21.9 Interface fancyhdr: \pagesetup</b>        | 49 |
| 17.6 /.is enum                                                      | 22 | 21.9.1 \pagesetup                                 |    |
| 17.7 /.store as code                                                | 22 | 21.9.2 Keys for \pagesetup                        |    |
| 17.8 /.store as code in cs                                          | 23 | <b>21.10 marks</b>                                | 62 |
| 17.9 /.store as color                                               | 23 | <b>21.11 Interface tocloft: \tocsetup</b>         | 64 |
| 17.10 /.store as color in cs                                        | 23 | <b>21.12 Interface makecell: \repeatcell</b>      | 70 |
| 17.11 /.store as number                                             | 23 | 21.12.1 countvalues                               |    |
| 17.12 /.store as number in cs                                       | 23 | 21.12.2 \repeatcell                               |    |
| 17.13 /.store as glue                                               | 23 | 21.12.3 colortbl: patch for \CT@cellc             |    |
| 17.14 /.store as dimen                                              | 24 | <b>21.13 Interface enumitem</b>                   | 76 |
| 17.15 /.store as glue in cs                                         | 24 | <b>21.14 Interface truncate</b>                   | 77 |
| 17.16 /.store as dimen in cs                                        | 24 | <b>21.15 Interface appendix</b>                   | 79 |
| 17.17 /.estore as dimen in cs                                       | 24 | <b>21.16 Interface embedfile: \embedthisfile</b>  | 79 |
| <b>18 Interface etoolbox</b>                                        | 24 | <b>21.17 Interface environ</b>                    | 80 |
| 18.1 New macros provided with \providetcommand                      | 24 | <b>21.18 Interface umrand</b>                     | 80 |
| <b>19 Interface scrlfile</b>                                        | 25 | <b>21.19 Interface scrlfile</b>                   | 80 |
| 19.1 Star forms for \BeforeFile and \AfterFile                      | 25 | <b>21.20 Interface pgfkeys</b>                    | 81 |
| 19.2 Allowing \BeforeFile and \AfterFile when<br>\@input is invoked | 25 | <b>21.21 Interface etoolbox</b>                   | 93 |
| <b>20 List of corrected or patched commands</b>                     | 25 | <b>22 History</b>                                 | 95 |
| 20.1 LaTeX                                                          | 25 | [2010/12/07 v2.5]                                 | 95 |
| 20.2 bookmark                                                       | 25 | [2010/11/26 v2.4]                                 | 95 |
| 20.3 tocloft                                                        | 25 | [2010/11/21 v2.3]                                 | 95 |
| 20.4 colortbl                                                       | 26 | [2010/11/16 v2.2]                                 | 95 |
| 20.5 hypbmsec                                                       | 26 | [2010/11/15 v2.1]                                 | 95 |
| 20.6 enumitem                                                       | 27 | [2010/10/31 v2.0]                                 | 95 |
| 20.7 pgfkeys                                                        | 27 | [2010/10/27 v1.2]                                 | 96 |
|                                                                     | 25 | [2010/10/20 v1.0]                                 | 96 |
|                                                                     | 26 | <b>README</b>                                     | 96 |
|                                                                     | 26 | <b>23 Index</b>                                   | 96 |

# 1 Interface LATEX

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

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

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

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

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

# 2 Interface TikZ ([pgfmanual](#))

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

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

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

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

~~~~~ MEMO ~~~~~

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

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

```
\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.

| MEMO | | |
|---------------------------|-------------------------------|---|
| <code>class</code> | 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> |
| <code>shape</code> | 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>\paragrap</code>
leftmargin=left margin
rightmargin=right margin
drop=wears 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 |
| <code>font=format</code> | format for title | applied to the label and the title |
| <code>label</code> | title label | ex: <code>\arabic{section}</code> |
| <code>sep=labelsep</code> | space between label and title | |
| <code>left</code> | left margin | a glue: horizontal space before |
| <code>right</code> | right margin | a glue: horizontal space after |
| <code>top</code> | vertical space before | a glue: vertical space before |
| <code>bottom</code> | vertical space after | a glue: vertical space after |
| <code>mark</code> | macro with 1 parameter | ex <code>\sectionmark</code> : redefines <code>\...mark#1...</code> (the argument is the title) |
| <code>break</code> | parameterless macro | ex <code>\sectionbreak</code> to be executed just before the section command |
| <code>thesection</code> | toc and ref labels | redefines <code>\thesection</code>
<code>\thesubsection</code> etc.. are used to print references and the label in the table of contents |
| <code>before</code> | code before title body | |
| <code>after</code> | code after title body | |

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

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

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

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

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

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

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

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

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

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

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

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

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

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

However, it is always possible to use `\sectionformat` concurrently with `\titleformat` and `\titlespacing`. A better way to use different settings for odd and even pages is the use of `\ifthispageodd` from the KOMA-Script classes, or the features provided by `changepage.sty`, which is almost the same: actually `titlesec` test oddity of the `page` counter (*i.e.* `\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`.

| |
|--|
| <code>\bottomtitles{<glue percentage of \textheight}</code> |
|--|

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` are now independant of `titlesec` and `interfaces-titlesec`.

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

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

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

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

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

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

`interfaces` takes advantage of this and defines:

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

Macros for each sectionning command are defined:

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

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

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

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

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

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

```
\sectionformat\section{
    mark=\marksthe{section}{#1}
    \marksthecs{sectionanchor}{lastsectionanchor}
}

\pagesetup{
    head/right=\hyperref{}{}{\thefirstmarks{sectionanchor}}
                  {\thefirstmarks{section}}
}
```

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...

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

`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).

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

~~~~~ MEMO ~~~~~

---

numbered entries are typeset with this pseudo code:

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

and for not numbered entries:

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

---

\lofsetup and \lotsetup behave the same.

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

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

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

Example:

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

## 8 Interface fancyhdr : \pagestyle{}

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

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

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

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

`foot/left/odd` is the same as `odd/foot/left` etc. but `twoside` option is not checked... Even pages are ignored in `oneside` documents: this is 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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

- `\numexpr`,
- `$\pgfmath$` ( $\sim \usepackage{pgfmath}$  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  $\text{\LaTeX}$  `\@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  $\text{\LaTeX}$  `\@for` (*hint*: you can put `\@empty` before, to avoid expansion).

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

- ☞ If `[<alignment>]` is specified, then each cell is placed into a `\makecell[<alignment>]` command.
- ☞ If both `text/row<i>` and `text/col<i>` are specified for the same cell, then `makecell \diaghead` is used to display the values in a diagonally divided cell.
- ☞ If you want to see (in the `.log` file) the code introduced into the tabular (or array) by `\repeatcell` type `\let\itf@message=\message` before `\repeatcell`.

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

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

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

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

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

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

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

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

10.1 † Conversion to scaled points

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

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

10.2 † Variable length cell

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

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

10.3 † Empty rows

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

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

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

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

10.4 ↗ color palette

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

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

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

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

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

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

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

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

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

10.6 † Double entry array: the Body Mass Index

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

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

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

11 Interface ◉ enumitem : \setlist

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

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

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

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

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

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

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

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

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

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

or

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

\interitem

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

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

```
\item Second item  
\end{enumerate}
```

The example give in real world:

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

And the following follows...

12 Interface ↗ truncate

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

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

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

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

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

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

Hello word!

Hello world!

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

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

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

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

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

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

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

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

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

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

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

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

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

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

→ first quite...second...third (39.05698pt, 33.65436pt, 12.64691pt, total=85.35825pt).

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

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

13 Interface ↗ appendix and the appendices environment

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

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

14 Interface ↗ embedfile : \embedthisfile

```
\embedthisfile  
\embedthisfile*
```

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

`\embedthisfile*` embeds only the main file.

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

15 Interface ↗ environ

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

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

16 Interface umrand

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

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

17 Interface pgfkeys (pgfmanual)

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

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

~~~~~ SUMMARY ~~~~~

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

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

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

17.1 † Handler definition: `./blank`

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

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

The following make blank value behave the same as default :

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

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

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

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

17.2 † Handler definition: `./is private`

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

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

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

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

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

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

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

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

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

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

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

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

17.5 † Handler definition: .is multiple choice

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

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

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

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

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

17.6 † Handler definition: .is enum

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

17.9 † Handler definition: .store as color

When specifying a color, did you say:

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

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

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

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

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

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

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

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

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

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

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

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

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

17.13 † Handler definition: .store as glue

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

Furthermore:

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

the `-` key subtracts: $\backslash\text{glueexpr}\ \text{old value} - \langle\ \text{argument}\ \rangle\ \backslash\text{relax}$
 the `*` key scales: $\backslash\text{glueexpr}\ \text{old value} * \langle\ \text{argument}\ \rangle\ \backslash\text{relax}$

The $\langle\text{argument}\rangle$ is **not expanded** but the content of the key is ready to be used in place of a length. Scaling of two forms are allowed: *fractionnal* and *real*. *Real* scaling is converted to a fraction of 65536:

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

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

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

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

17.14 † New handler: .store as dimen

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

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

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

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

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

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

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

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

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

This handler has 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 $\sim\backslash\text{changefont}$ (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 `\providemode`

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

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

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

If the first occurrence of `{<code>}` into `<hook>` has been removed
 then `\ifin@` is set to `\iftrue`
 otherwise `\ifin@` is set to `\iffalse` (`<hook>` does not contain `{<code>}`).
 this allows loops of the form:
`\in@true`
`\@whilesw\ifin@\fi{\removefrom{<hook>}{<code>}}`

- ☞ `\eremovefrom` and `\xremovefrom` expand `{<code>}` before trying to remove it from `<hook>`. Only `{<code>}` is expanded, the current replacement text of `<hook>` is not.
- ☞ `g` and `x` versions make the assignment to `<hook>` global.
- ☞ `\removefromstart` remove `{<code>}` only if it appears at the very beginning of `<hook>`.

19 Interface `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* `interfaces-titlesec` and `interfaces-bookmark` are loaded.

20.1 `LATEX`

| | Patched command | Type | Importance | Module | ref. page |
|---|--|------|-------------------------------------|-------------------------------|-----------|
| ❶ | <code>\@pkgextension</code>
<code>\@clsextension</code>
<code>\@ifpackageloaded</code>
<code>\@ifclassloaded</code> | | minor
optimisation or correction | <code>interfaces-LaTeX</code> | 80 |

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

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

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

| | Patched command | Type | Importance | Module | ref. page |
|---|-------------------------------|----------------|---------------------------------|--|-----------|
| ❷ | <code>\addcontentsline</code> | <i>prepend</i> | important
but no side effect | <code>interfaces-titlesec</code>
<code>interfaces-bookmark</code> | 42 |

`\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>interfaces-scrfile</code> | 81 |

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

Original:

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

Modified:

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

20.2 † bookmark \\bookmark

| | Patched command | Type | Importance | Module | ref. page |
|---|------------------------|--------------|------------|----------------------------------|-----------|
| ④ | <code>\bookmark</code> | <i>modif</i> | important | <code>interfaces-bookmark</code> | 44 |

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

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

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

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

20.3 † tocloft \\newlistof

| | Patched command | Type | Importance | Module | ref. page |
|---|-------------------------|------|---------------------------------|---------------------------------|-----------|
| ⑤ | <code>\newlistof</code> | | important
but no side effect | <code>interfaces-tocloft</code> | |

20.4 † colortbl

| | Patched command | Type | Importance | Module | ref. page |
|---|------------------------|--------------|---------------------------------|----------------------------------|-----------|
| ⑥ | <code>\CT@cellc</code> | <i>corr.</i> | important
but no side effect | <code>interfaces-makecell</code> | 76 |

`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}%
}
```

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

20.5 † hypbmsec

| | Patched command | Type | Importance | Module | ref. page |
|---|---|--------------|---------------------------------|--|-----------|
| ⑦ | <code>\hbs@do</code>
<code>\hbs@process</code> | <i>redef</i> | important
but no side effect | <code>interfaces-hypbmsec</code>
<code>interfaces-titlesec</code> | 46 |

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

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

20.6 † `enumitem`

| | Patched command | Type | Importance | Module | ref. page |
|---|-----------------------|--------------|-------------------------------------|----------------------------------|-----------|
| ⑧ | <code>\setlist</code> | <i>redef</i> | minor
optimisation or correction | <code>interfaces-enumitem</code> | 76 |

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

20.7 † `pgfkeys.code.tex`

| | Patched command | Type | Importance | Module | ref. page |
|---|--|--------------|------------|---------------------------------|-----------|
| ⑨ | <code>\pgfkeys@unpack</code>
<code>\pgfkeys@case@three</code> | <i>redef</i> | important | <code>interfaces-pgfkeys</code> | 82 |

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

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

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

21 Implementation

21.1 Interfaces

The package namespace is `itf@`.

```

1 <*package>
2 \NeedsTeXFormat{LaTeX2e}[2005/12/01]
3 \ProvidesPackage{interfaces}
4   [2010/12/07 v2.5 - pgfkeys (pgf 2.0 et 2.1 macros for interfaces (FC)]
```

21.1.1 Requirements

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

21.1.2 The package options / list of interfaces

```

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

21.2 Interfaces-base

```

28 <*base>
29 \ProvidesPackage{interfaces-base}
30   [2010/12/07 v2.5 Basic macros for interfaces package (FC)]
31 \RequirePackage{etex,ltxcmds,etoolbox,pgfkeys,scrlfile,infwarerr}
32 \AtEndOfPackage{%
33   \RequirePackage{interfaces-etoolbox}%
34   \RequirePackage{interfaces-scrlfile}%
35   \RequirePackage{interfaces-pgfkeys}%
36   \RequirePackage{interfaces-tikz}%
37   \RequirePackage{interfaces-umrand}% There is a \BeforePackage here !
38 }% AtEndOfPackage
```

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

```

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

```

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

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

```

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

```

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

```

21.2.1 Interfaces patches

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

```

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

```

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

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

```

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

```

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

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

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

```

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

```

```

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

```

21.2.2 Helper macros

```

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

```

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

\itf@strip@dot

```

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

```

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

USAGE: \itf@expandonceifcs\macrotest

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

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

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

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

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

```

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

```

21.2.3 Numbers

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

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

```

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

```

21.2.4 Units and math units

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

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

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

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

```

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

```

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

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

`\ratio` is used.

```

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

```

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

```

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

```

21.2.5 Default color

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

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

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

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

```

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

245 </base>

```

21.3 ⇔ Interface LATEX

```

246 /*LaTeX*/
247 \ProvidesPackage{interfaces-LaTeX}
248   [2010/12/07 v2.5 pgfkeys macros for LaTeX (FC)]
249 \RequirePackage{interfaces-base}
250 \itf@pgfkeys

```

21.3.1 \changefont

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

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

```

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

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

```

```

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

```

294 </TeX>

21.4 Interface TikZ ([pgfmanual](#))

```

296 <*tikz>
297 \ProvidesPackage{interfaces-tikz}
298   [2010/12/07 v2.5 load tikz libraries easily - pgf 2.0 & 2.1 (FC)]
299 \RequirePackage{interfaces-base}
300 \itf@pgfkeys

```

21.4.1 Catcode verifications

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

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

```

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

```

21.4.2 List of pgf/TikZ packages and libraries

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

```

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

```

```
311 pgfpic2e,fp,%           + fp required for fixedpointarithmetic
312 }% \alltikzpackages
```

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

```
313 \newcommand*\allpgfmodules{datavisualization,%
314 decorations,%
315 matrix,%
316 oo,%
317 parser,%
318 plot,%
319 shapes,%
320 snakes,%
321 sorting%
322 }% \allpgfmodules
```

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

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

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

```

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

```

21.4.3 pgfkeys definitions for \usetikz

```

389 \pgfqkeys{/itf/tikz}{%
390   basic/.code={%
391     \usetikzlibrary{calc,%
392                   arrows,%
393                   backgrounds,%
394                   scopes}%
395   }%
396 },%
397 basic/.value forbidden,% \alltikzpackages
398 base/.alias=basic,%
399 all/.code={%
400   \expandafter\RequirePackage\expandafter{\alltikzpackages}%
401   \expandafter\usepgfmodule\expandafter{\allpgfmodules}% see pgfcorescopes.code.tex
402   \expandafter\usetikzlibrary\expandafter{\alltikzlibraries}% from: tikz.code.tex
403 },%
404 .unknown/.code=\itf@usetikzcode\pgfkeyscurrentname
405 }% \pgfkeys for \usetikz

```

\usetikz

```

406 \newrobustcmd*\usetikz[1]{%
407   \RequirePackage{tikz}%
408   \ifblank{#1}%
409     {\pgfqkeys{/itf/tikz}{all}}%
410     {\pgfqkeys{/itf/tikz}{#1}}%
411   \itf@pgfkeys% overwritten by \input primitive otherwise
412 }% \usetikz

```

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

```

413 \newrobustcmd*\itf@usetikzcode[1]{\begingroup
414   \edef\x{%
415     \noexpand\in@{,#1}{,\alltikzpackages,}%
416     \noexpand\ifin@ \noexpand\usepackage{#1}%
417     \noexpand\else
418       \noexpand\in@{,#1}{,\allpgfmodules,}%
419       \noexpand\ifin@ \noexpand\usepgfmodule{#1}%
420       \noexpand\else \noexpand\usetikzlibrary{#1}% \usetikzlibrary by default
421       \noexpand\fi
422     \noexpand\fi
423   }\expandafter\endgroup\x
424 }% \itf@usetikzcode

```

425 ⟨/tikz⟩

21.5 ↗ Interface ↗ titlesec : \sectionformat

```
426 {*titlesec}
427 \ProvidesPackage{interfaces-titlesec}
428   [2010/12/07 v2.5 pgfkeys macros for titlesec (FC)]
429 \RequirePackage{interfaces-base}
430 \RequirePackage{auxhook}% <fix for \ttlp@append (odd/even pages)>
431 \RequirePackage{getttitlestring}
432 \itf@Check@Package{titlesec}
433 \itf@pgfkeys
```

21.5.1 titlesec modifications

```
434 %% t i t l e s e c "p a t c h e s"
435 \robustify\titleclass \robustify\titleformat \robustify\titlespacing
436 \AddLineBeginMainAux{%
437   \string\providetoggle\string\ttlp@append[2]{}}%
438 }
439 \BeforeFile*{ttlkeys.def}{%
440   \expandafter\edef\csname itf@ttlkeys.def\endcsname{%
441     \catcode32 \the\catcode32\relax
442     \catcode37 \the\catcode37\relax
443     \endlinechar\the\endlinechar\relax}%
444   \catcode37 14\relax % percent=comment (inside .dtx files...)
445   \catcode32 9\relax % ignore spaces
446   \endlinechar\m@ne % and line feeds
447 }
448 \AfterFile*{ttlkeys.def}{\csname itf@ttlkeys.def\endcsname}%

```

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

```
449 %%-----
450 %% runinifempty => like hang but no paragraph break (like runin) if title is empty
451 \def\ttlh@runinifempty{%
452   \csname ttlh@\ifx\ttl@savetitle@\empty runin\else hang\fi\endcsname
453 }% \ttlh@runinifempty
```

21.5.2 \sectionformat

```
\sectionformat \sectionformat\subsection* [<shape>]{<key=value>}
454 \newrobustcmd*\sectionformat[1]{\ifstar
455   {\itf@sectionformat{#1}{/*}}
456   {\itf@sectionformat{#1}{}}%
457 }% \sectionformat
458 \def\itf@sectionformat#1#2{@testopt
459   {\expandafter\expandafter\expandafter
460     \itf@sectionformat\rmat\expandafter\gobble\string#1\@nil{#2}}{}%
461 }% \sectionformat
462 \def\itf@sectionformat\rmat#1\@nil#2[#3]#4{\itf@pgfkeys\ifblank{#3}{%
463   \pgfqkeys{/itf/titlesec/#1#2}{%
464     /itf/titlesec/@setup={#1}{#2},#4,/itf/titlesec/@update={#1}{#2}}%
465   \pgfqkeys{/itf/titlesec/#1#2}{%
466     /itf/titlesec/@setup={#1}{#2},#4,shape={#3},/itf/titlesec/@update={#1}{#2}}}}%
467 }% \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 468 % private keys @setup and @update
469 \pgfqkeys{/itf/titlesec}{%
470 @setup/.code 2 args=\itf@ttl@setup{#1}{#2},%
471 @update/.code 2 args=\expandafter\itf@ttl@update

```

472          \csname#1\endcsname{/itf/titlesec/#1}{#2}%
473 }
474 \newrobustcmd*\itf@ttl@setup[2]{%
475   \pgfkeysifdefined{/itf/titlesec/#1#2/shape}{}%
476   {\itf@ttl@newsection{#1}{#2}}%
477   \ifcsundef{tls@#1#2}{}% no previous spacing settings
478   {\expandafter\expandafter\expandafter%
479    \itf@ttl@getspacing\csname tls@#1#2\endcsname{/itf/titlesec/#1#2}}%
480   \ifcsundef{ttlf@#1#2}{}% no previous format settings
481   {\ifttl@explicit
482    \expandafter\expandafter\expandafter%
483    \itf@ttl@getformat\csname ttlf@#1#2\endcsname{/itf/titlesec/#1#2}}%
484   \else
485   \expandafter\expandafter\expandafter%
486   \itf@ttl@getformat\csname ttlf@#1#2\endcsname{/itf/titlesec/#1#2}%
487   \fi}%
488 }% \itf@ttl@setup
489 \def\itf@ttl@setkey#1#2{\begingroup
490   \def\x{\pgfkeysvalueof{#1}}\def\y{#2}%
491   \expandafter\endgroup\ifx\x\y\else
492     \pgfkeyssetvalue{#1}{#2}\fi
493 }% \itf@ttl@setkey
494 \def\itf@ttl@getspacing#1#2#3#4#5#6{%
495   \itf@ttl@setkey{#6/left}{#1}%
496   \itf@ttl@setkey{#6/right}{#2}%
497   \itf@ttl@setkey{#6/top}{#3}%
498   \itf@ttl@setkey{#6/bottom}{#4}%
499   \itf@ttl@setkey{#6/indent first}{#5}%
500 }% \itf@ttl@getspacing
501 \def\itf@ttl@getformat#1#2{%
502   \csname ltx@\ifx\pgfkeysvalueof#1first\else second\fi oftwo\endcsname
503   {\itf@ttl@get@format{}}%
504   {\itf@ttl@get@format{#1}{#2}}%
505 }% \itf@ttl@getformat
506 \def\itf@ttl@get@format#1#2#3#4#5#6#7{%
507   \if\relax\detokenize{#1}\relax\else
508     \pgfkeysEsetvalue{#7/shape}{\expandafter\itf@strip@at\string#1}%
509   \fi
510   \itf@ttl@setkey{#7/font}{#2}%
511   \itf@ttl@setkey{#7/label}{#3}%
512   \itf@ttl@setkey{#7/sep}{#4}%
513   \itf@ttl@setkey{#7/before}{#5}%
514   \itf@ttl@setkey{#7/after}{#6}%
515 }% \itf@ttl@get@format

```

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

```

516 \newrobustcmd*\itf@ttl@update[3]{%
517   \ifblank{#3}
518     {\ttl@getkeys{#1}{titlesec}}
519     {\ttl@getkeys{name=#1, numberless}{titlesec}}%
520   \pgfkeysgetvalue{#2#3/shape}\itf@tp
521   \ifcsundef{tth@}\itf@tp\else
522     {\begingroup \makeatletter
523      \InputIfFileExists{\itf@tp.tss}{}{%
524        \PackageError{titlesec}{Unknown shape}%
525        {Shapes are defined in files with extension tss\MessageBreak
526         Either you have misspelled the shape\MessageBreak
527         or there is no a \itf@tp.tss file}}%
528     \endgroup
529   }%
530   \ifttl@explicit \begingroup
531     \pgfkeysgetvalue{#2#3/before}\pgfkeys@temp
532     \toks@\expandafter{\pgfkeys@temp}%

```

```

533     \edef\x{\endgroup\def\noexpand\itf@tp##1{\the\toks@}}%
534     \x \pgfkeyslet{#2#3/before@cmd}\itf@tp
535     @temptokena{\csdef{ttlf@\ttl@a}##1}%
536     \the@temptokena{%
537         \pgfkeysvalueof{#2#3/shape@cs}%
538         \pgfkeysvalueof{#2#3/font}%
539         \pgfkeysvalueof{#2#3/label}%
540         \pgfkeysvalueof{#2#3/sep}%
541         \pgfkeysvalueof{#2#3/before@cmd}##1}%
542         \pgfkeysvalueof{#2#3/after}{}%
543     \else@temptokena{\csdef{ttlf@\ttl@a}{}%
544     \the@temptokena{%
545         \pgfkeysvalueof{#2#3/shape@cs}%
546         \pgfkeysvalueof{#2#3/font}%
547         \pgfkeysvalueof{#2#3/label}%
548         \pgfkeysvalueof{#2#3/sep}%
549         \pgfkeysvalueof{#2#3/before}{}%
550         \pgfkeysvalueof{#2#3/after}{}%
551     \fi
552     \csname ttl@compat\ttl@a\endcsname
553     \ttl@spacing@ii
554     {\pgfkeysvalueof{#2#3/indent first}{}%
555     \pgfkeysvalueof{#2#3/left}{}%
556     \pgfkeysvalueof{#2#3/top}{}%
557     \pgfkeysvalueof{#2#3/bottom}{}%
558     [\pgfkeysvalueof{#2#3/right}]}%
559 }% \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.

```

560 %-----%
561 %% titleclass
562 \newrobustcmd*\itf@ttl@titleclass{@ifstar
563   {\itf@ttl@titleclass@i*}
564   {\itf@ttl@titleclass@i{}}
565 }% \itf@ttl@titleclass
566 \def\itf@ttl@titleclass@i#1#2[#4]#5@nil{\ifblank{#4}
567   {%
568     \expandafter\titleclass\csname#2\endcsname{#3}%
569   }
570   {%
571     \expandafter\titleclass\csname#2\endcsname{#3}[{#4}]% #4 = sup-level
572     \ifrelax#1\relax % not starred
573       \expandafter\expandafter\expandafter%
574         \itf@ttl@titleclass@i\expandafter\gobble\string#4\relax{#2}%
575     \fi
576   }%
577   \expandafter\providecommand\csname l@#2\endcsname[2]{% no entry in toc
578 }% \itf@ttl@titleclass@i
579 \def\itf@ttl@titleclass@i#1\relax#2{%
580   \ifcsundef{c@#2}
581     {\newcounter{#2}[{#1}]}% creates counter for #1
582     {\@addtoreset{#2}{#1}}%
583   \csxdef{the#2}{%
584     \expandafter\noexpand\csname the#1\endcsname
585     .\noexpand\arabic\expandafter\noexpand\csname c@#2\endcsname
586   }% \the#2
587   \ifcsundef{l@#1}{}{%
588     \expandafter\providecommand\csname l@#2\endcsname[2]{%
589       \csname l@#1\endcsname{##1}##2}%
590     }% \l@#2 = \l@newsection is (by default) the same as \l@sup-level
591   }%

```

```
592 }% \itf@ttl@titleclass@ii
```

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

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

```
593 %%-----  
594 %% Creating new sections  
595 \def\itf@ttl@createsection#1{  
  sectionning command undefined: temporarily defined  
  \expandafter\titleclass\csname#1\endcsname{straight}[*] in straight class  
  \itf@AfterPackage{hypbmsec}{%  
    \AfterPreamble{  
      \csletcs{hbs@#1}{#1}%  
      \csxdef{#1}{\noexpand\hbs@do\expandafter\noexpand\csname hbs@#1\endcsname}{%  
    }%  
  }%  
  \itf@AfterPackage{interfaces-tocloft}{%  
    \itf@cft@newsection{toc}{#1}{%  
  }%  
  \itf@AfterPackage{interfaces-hypbmsec}{%  
    \expandafter\let\csname last#1title\endcsname\relax  
    \expandafter\let\csname last#1bookmark\endcsname\relax  
    \expandafter\let\csname last#1toc\endcsname\relax  
  }%  
} % \itf@ttl@createsection
```

21.5.3 Keys for `\sectionformat`

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

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

```
612 %%-----  
613 %% Defining keys for each section  
614 \def\itf@ttl@newsection#1#2{  
  #1=section, #2=/* or empty  
  \ifcscundef{#1}{  
    sectionning command undefined: temporarily defined in straight class  
    {\itf@ttl@createsection{#1}}  
  }%  
  \ifblank{#2}{%  
    \pgfqkeys{/itf/titlesec/#1}{% these keys do not exist for starred sections  
      class/.code=\itf@ttl@titleclass{#1##1}[@nil,% with \@addtoreset  
        class/.value required,% if sup-level is given  
      class*/.code=\itf@ttl@titleclass*{#1##1}[@nil,% without \@addtoreset  
        class*/.value required,% even if sup-level is given  
      shape/.initial=% initial shapes set by titlesec.sty for not numbered sections.  
      the#1/.store in cs=the#1,%  
      the#1/.default=\pgfkeysvalueof{/itf/titlesec/#1/label},%  
    }%  
    {% starred section: unless titlesec, we do not put the default shape to hang,  
     \pgfkeysifdefined{/itf/titlesec/#1/shape}{% but to the shape of the not starred  
       \pgfkeysgetvalue{/itf/titlesec/#1/shape}\itf@tp% sectionning command  
       {\def\itf@tp{hang}}%  
     \pgfkeyslet{/itf/titlesec/#1##2/shape}\itf@tp%  
    }%  
    \pgfqkeys{/itf/titlesec/#1##2}{%  
      shape/.value required,%  
      shape@cs/.initial=  
        \csname ttlh@\pgfkeysvvalueof{/itf/titlesec/#1##2/shape}\endcsname,% constant  
        shape@cs/.is private=interfaces-titlesec,% constant  
      font/.store as code,%  
        format/.alias*=font,%  
      label/.store as code,%
```

```

642     sep/.estore as glue,%
643     labelsep/.alias*=sep,%
644     before/.store as code,%
645     before@cmd/.is private=interfaces-titlesec,%
646     after/.store as code,%
647     indent first/.true=\pgfkeyssetvalue{/itf/titlesec/#1#2/indent first}{\@ne},%
648     indent first/.false=\pgfkeyssetvalue{/itf/titlesec/#1#2/indent first}{\z@},%
649     identfirst/.alias=indent first,%
650     indent first=true,%
651     left/.estore as glue,%
652     top/.estore as glue,%
653     bottom/.estore as glue,%
654     right/.estore as glue,%
655     break/.code=\ifblank{##1}
656         {\cslet{\#1break}{\empty}}
657         {\protected\csdef{\#1break}{##1}},%
658     break*/.code=\protected\apptocs{\#1break}{##1},%
659     break+/.alias=break*,%
660     mark/.code=\ifblank{##1}
661         {\cslet{\#1mark}{\gobble}}
662         {\protected\long\csdef{\#1mark}####1{##1}},%
663     mark/.default=,
664 }% \pgfkeys
665 \ifblank{##2}{% no default marks for starred (numberless) \section*
666 \itf@AfterPackage{interfaces-marks}{% requires interfaces-marks
667     \pgfkeys{/itf/titlesec/#1/mark/.default=\marksthe{##1}}%
668 }% AfterPackage interfaces-mark
669 \itf@AfterPackage{interfaces-bookmark}{% requires interfaces-bookmark
670     \pgfqkeys{/itf/titlesec/#1}{% does not exist for starred sections
671         bookmark/.is multiple choice,%
672         bookmark/level/.estore in cs=toclevel@#, % depth of the bookmark
673         bookmark/level/.default=\ifcsvvoid{ttll@#1}{99}{\csuse{ttll@#1}},%
674         bookmark/level/.blank=\ifcsvvoid{toclevel@#1}
675             {\ifcsvvoid{ttll@#1}{99}{\csuse{ttll@#1}}{\csuse{toclevel@#1}}},%
676         bookmark/toclevel/.alias=bookmark/level,%
677         bookmark/bold/.code=\ifblank{##1}
678             {\pgfkeyslet{/itf/titlesec/#1/bookmark/bold}{\empty}}
679             {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/bold}{bold=##1},},%
680         bookmark/bold/.default=true,%
681         bookmark/bold*/.code=\ifblank{##1}
682             {\pgfkeyslet{/itf/titlesec/#1/bookmark/bold*}{\empty}}
683             {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/bold*}{bold*=##1},},%
684         bookmark/bold*/.default=true,%
685         bookmark/italic/.code=\ifblank{##1}
686             {\pgfkeyslet{/itf/titlesec/#1/bookmark/italic}{\empty}}
687             {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/italic}{italic=##1},},%
688         bookmark/italic/.default=true,%
689         bookmark/italic*/.code=\ifblank{##1}
690             {\pgfkeyslet{/itf/titlesec/#1/bookmark/italic*}{\empty}}
691             {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/italic*}{italic*=##1},},%
692         bookmark/italic*/.default=true,%
693         bookmark/color/.code=\ifblank{##1}
694             {\pgfkeyslet{/itf/titlesec/#1/bookmark/color}{\empty}}
695             {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/color}{color=##1},},%
696         bookmark/color/.default=,
697         bookmark/color*/.code=\ifblank{##1}
698             {\pgfkeyslet{/itf/titlesec/#1/bookmark/color*}{\empty}}
699             {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/color*}{color*=##1},},%
700         bookmark/color*/.default=,
701         bookmark/on/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on}{disable=false},%
702         bookmark/on/.value forbidden,%
703         bookmark/off/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on}{disable=true},%
704         bookmark/off/.value forbidden,%
705         bookmark/none/.alias=bookmark/off,%
706         bookmark/on*/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on*}{disable*=false},%
    }
```

```

707     bookmark/off*/.code=\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/on*}{disable*=true},%
708     bookmark/none*/.alias=bookmark/off*,%
709     bookmark/disable*/.alias=bookmark/off*,%
710     bookmark/style/.code=\ifblank{##1}%
711         {\pgfkeyslet{/itf/titlesec/#1/bookmark/style}\@empty}
712         {\pgfkeyssetvalue{/itf/titlesec/#1/bookmark/style}{style=##1},}%,%
713     bookmark/style/.default=,%  

714     bookmark/.unknown/.code=\pgfkeysEsetvalue{/itf/titlesec/#1/bookmark/other}%
715                                     {\pgfkeyscurrentname={##1}},%
716     bookmark/bold/.initial=,%  

717     bookmark/bold*/.initial=,%  

718     bookmark/italic/.initial=,%  

719     bookmark/italic*/.initial=,%  

720     bookmark/color/.initial=,%  

721     bookmark/color*/.initial=,%  

722     bookmark/on/.initial=,%  

723     bookmark/on*/.initial=,%  

724     bookmark/style/.initial=,%  

725 }% \AfterPackage*{interfaces-bookmark}  

726 }% not starred (numberless) \section  

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

728 }% \itf@ttl@newsection

```

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

```

729 %% bookmarks styles for sectionning commands:  

730 \@makeother\#  

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

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

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

734         {\endgroup}  

735         {\@PackageInfoNoLine{interfaces-titlesec}  

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

737         {\@PackageWarningNoLine{interfaces-titlesec}  

738             {Patch of \string\addcontentsline\space failed  

739                 \MessageBreak bookmark-style specs for sections  

740                 \MessageBreak will be unavailable}}%  

741 }% AfterPackage interfaces-bookmark  

742 \catcode`\#=6  

743 \def\itf@ttl@BKM#1{%
744     \pgfkeysifdefined{#1/.@cmd}{%
745         {%
746             \edef\itf@tp{\% do nothing if the keys are not set (shortcut)  

747                 \pgfkeysvalueof{#1:bold}\%  

748                 \pgfkeysvalueof{#1:bold*}\%  

749                 \pgfkeysvalueof{#1/italic}\%  

750                 \pgfkeysvalueof{#1/italic*}\%  

751                 \pgfkeysvalueof{#1/color}\%  

752                 \pgfkeysvalueof{#1/color*}\%  

753                 \pgfkeysvalueof{#1/on}\%  

754                 \pgfkeysvalueof{#1/on*}\%  

755                 \pgfkeysvalueof{#1/style}\%}  

756             \ifx\itf@tp\@empty  

757             \else  

758                 \global\pgfkeyslet{#1:bold*}\@empty  

759                 \global\pgfkeyslet{#1/italic*}\@empty  

760                 \global\pgfkeyslet{#1/color*}\@empty  

761                 \global\pgfkeyslet{#1/on*}\@empty  

762                 \expandafter\bookmarksetup\expandafter{\itf@tp}\%  

763                 \pgfkeysifdefined{/itf/bookmark/*}\%  

764                     { \bookmarksetup{color*=} overwrites \sectionformat...{bookmark/color*=} }  

765                     \pgfkeysgetvalue{/itf/bookmark/*}\itf@tp  

766                     \expandafter\bookmarksetup\expandafter{\itf@tp}\%}

```

```

767          }
768          { }%
769          \fi}
770          { }%
771 }% \itf@ttl@BKM

```

21.5.4 `\bottomtitles`

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

```

772 %%-----%
773 % b o t t o m t i t l e s
774 \newrobustcmd*\bottomtitles{@ifnextchar\bgroup
775   \itf@setbottomtitlespace
776   {\afterassignment\itf@set@bottomtitle@space\itf@sk}%
777 }% \bottomtitles
778 \def\itf@setbottomtitlespace#1{%
779   \@defaultunits\itf@sk#1\textheight\relax\@nnil
780   \itf@set@bottomtitle@space
781 }% \itf@setbottomtitlespace
782 \def\itf@set@bottomtitle@space{\edef\bottomtitlespace{\the\dimexpr\itf@sk}%
783 </titlesec>

```

21.6 Interface `hyperref`

```

784 (*hyperref)
785 \ProvidesPackage{interfaces-hyperref}
786   [2010/12/07 v2.5 provides raised hyperdef (FC)]
787 \RequirePackage{interfaces-base}
788 \itf@Check@Package{hyperref}

```

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

```

789 \newrobustcmd\raisedhyperdef[4][]{\begingroup
790   \ifblank{#1}{}
791   { }%
792   \expandafter\def\expandafter\HyperRaiseLinkHook\expandafter{%
793     \expandafter\def\expandafter\HyperRaiseLinkHook
794       \expandafter{\HyperRaiseLinkHook}% restore current definition
795     \HyperRaiseLinkHook
796     \advance\noexpand\HyperRaiseLinkLength by\dimexpr#1\relax\relax}%
797   { }%
798   \ifx \\#2\\def \Hy@AnchorName {#3}\else
799     \def \Hy@AnchorName {#2.#3}%
800   \fi
801   \ltx@leavevmode
802   \Hy@raisedlink{\hyper@anchorstart{\Hy@AnchorName}\hyper@anchorend}%
803   \endgroup
804   #4%
805 }% \raisedhyperdef
806 </hyperref>

```

21.7 `Interface` `bookmark`

```

808 (*bookmark)
809 \ProvidesPackage{interfaces-bookmark}
810   [2010/12/07 v2.5 more control on bookmarks (FC)]
811 \RequirePackage{interfaces-base}
812 \RequirePackage{interfaces-hyperref}
813 \RequirePackage{refcount}
814 \itf@Check@Package{bookmark}
815 \itf@pgfkeys

```

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

```

816 \expandafter\itf@patch\expandafter\patchcmd\csname\string\bookmark\endcsname
817   {\immediate\write\mainaux}{\itf@BKM@write}
818   {\@PackageInfoNoLine{interfaces-bookmark}
819     {Command \expandafter\string\csname\string\bookmark\endcsname\space
820      successfully patched!}}
821   {\@PackageWarningNoLine{interfaces-bookmark}
822     {Command \expandafter\string\csname\string\bookmark\endcsname\space
823      could not be patched\MessageBreak
824      Loading of interfaces-bookmark is canceled}%
825     \endinput}
```

21.7.1 star keys* definitions

```

826 \global\let\itf@BKM@enable@ne 0 = disable*, 1=enable, \@M = disable
827 \pgfqkeys{/itf/bookmark}{%
828   disable/.true=\global\let\itf@BKM@enable@\M,%
829   disable/.false=\global\let\itf@BKM@enable@ne,%
830   disable*/.true=\global\let\itf@BKM@enable\z@,%
831   disable*/.false=\global\let\itf@BKM@enable@ne,%
832   atend/.true=\ifnum\itf@BKM@enable>0\edef\itf@BKM@enable{\number-\itf@BKM@enable}\fi,%
833   atend/.false=\ifnum\itf@BKM@enable<0\edef\itf@BKM@enable{\number-\itf@BKM@enable}\fi,%
834   bold*/.true=\pgfqkeys{/itf/bookmark}{@save={bold=true}},%
835   bold*/.false=\pgfqkeys{/itf/bookmark}{@save={bold=false}},%
836   bold*/@blank/.code=\pgfqkeys{/itf/bookmark/@restore}{bold},%
837   bold*/.blank=@blank,%
838   italic*/.true=\pgfqkeys{/itf/bookmark}{@save={italic=true}},%
839   italic*/.false=\pgfkeys{/itf/bookmark}{@save={italic=false}},%
840   italic*/@blank/.code=\pgfqkeys{/itf/bookmark/@restore}{italic},%
841   italic*/.blank=@blank,%
842   color*/.code={\ifblank{#1}%
843     {\pgfqkeys{/itf/bookmark/@restore}{color}}
844     {\pgfqkeys{/itf/bookmark}{@save={color={#1}}}}%
845   },%
846   @save/.code={%
847     \pgfkeysifdefined{/itf/bookmark/@save/color}{%
848       {\pgfkeyslet{/itf/bookmark/@save/color}\BKM@color}%
849     }%
850     \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{%
851       {\pgfkeyslet{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}%
852     }%
853     \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}{%
854       {\pgfkeysEsetvalue{/itf/bookmark/@save/grouplevel}{\the\currentgrouplevel}}%
855       \pgfkeysaddvalue{/itf/bookmark/*}{}{, #1}%
856       \bookmarksetup{#1}%
857     },%
858     @restore/color/.code=%
859     \pgfkeysifdefined{/itf/bookmark/@save/color}{%
860       {\pgfkeysgetvalue{/itf/bookmark/@save/color}\BKM@color}%
861     },%
862     @restore/FontStyle/.code=%
863     \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{%
864       {\pgfkeysgetvalue{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}%
865     },%
866     @restore/.code={%
867       \pgfkeysifdefined{/itf/bookmark/@save/color}{%
868         {\pgfkeysgetvalue{/itf/bookmark/@save/color}\BKM@color}%
869       },%
870       \pgfkeysifdefined{/itf/bookmark/@save/FontStyle}{%
871         {\pgfkeysgetvalue{/itf/bookmark/@save/FontStyle}\BS@BKM@FontStyle}%
872       },%
873       \pgfkeysifdefined{/itf/bookmark/@save/grouplevel}{%
874         {\% shortcut: \bookmarksetup{bold*} called inside main group (level 0)
875          \ifnum\pgfkeysvalueof{/itf/bookmark/@save/grouplevel}=z@
876            \global\pgfkeyslet{/itf/bookmark/@save/FontStyle}\@undefined
877          }%
878        }%
879      }%
880    }%
881  }%
```

```

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

```

925 \def\itf@BKM@write#1{%
926   \aftergroup\itf@BKM@restore
927   \ifcase\itf@BKM@enable
928     \global\let\itf@BKM@enable\@ne
929   \or
930     \immediate\write\@mainaux{\#1}%
931   \else
932     \ifcase -\itf@BKM@enable % (case -0 = 0 tested before)
933     \or % (case -1 = enable but At End)
934       \let\itf@BKM@enable\@ne
935       \edef\itf@tp{\immediate\write\@mainaux{\#1}}%

```

```

936           \expandafter\BookmarkAtEnd\expandafter{\itf@tp}%
937             \fi
938           \fi
939 }% \itf@BKM@write

```

21.7.2 ref and pageref keys

`\itf@BKM@pageref`

```

940 %% pageref= key for \bookmark (using refcount : \getpagerefnumber)
941 \def\itf@BKM@pageref[#1]{%
942   \expandafter\itf@BKM@@pageref
943   \csname r@#1\endcsname{#1}%
944 }% \itf@BKM@pageref
945 \def\itf@BKM@@pageref#1#2{%
946   \ifx#1\relax
947     \protect\G@refundefinedtrue
948     \@latex@warning{%
949       Hyper reference '#2' on page \thepage \space undefined
950       \MessageBreak for bookmark destination page
951     }%
952   \fi
953   \edef\BKM@page{\getrefbykeydefault{#2}{page}{1}}%
954 }% \itf@BKM@@pageref
955 \define@key{BKM}{pageref}{%
956   \Hy@babelnormalise\itf@BKM@pageref[{-#1}]}

```

`\itf@BKM@ref`

```

957 %% ref= key for \bookmark
958 \def\itf@BKM@ref[#1]{%
959   \expandafter\itf@BKM@@ref
960   \csname r@#1\endcsname{#1}%
961 }% \itf@BKM@ref
962 \def\itf@BKM@@ref#1#2{%
963   \ifx#1\relax
964     \protect\G@refundefinedtrue
965     \@latex@warning{%
966       Hyper reference '#2' on page \thepage \space undefined
967       \MessageBreak for bookmark destination
968     }%
969   \def\BKM@dest{??}%
970   \else
971     \edef\BKM@dest{\expandafter\@fourthoffive#1\empty\empty}%
972   \fi
973 }% \BKM@@ref
974 \define@key{BKM}{ref}{%
975   \Hy@babelnormalise\itf@BKM@ref[{-#1}]}
976 </bookmark>

```

21.8 Interface `hypbmsec`

21.8.1 interfaces-hypbmsec

```

977 <*hypbmsec>
978 \ProvidesPackage{interfaces-hypbmsec}
979   [2010/12/07 v2.5 lastsection (title,anchor, etc. - FC)]
980 \RequirePackage{interfaces-base}
981 \RequirePackage{gettitledstring}
982 \itf@Check@Package{hypbmsec}

\hbs@do (patch)
983 \def\itf@hbs@do#1{%
984   \xdef\itf@hbs@currentname{\expandafter\itf@strip@at\string#1}%

```

```

985   \let\hbs@tocstring\relax % <for starred sections>
986   \let\hbs@bmstring\relax % <for starred sections>
987   \@ifstar
988     {\itf@hbs@ssect#1}%
989     {\let\hbs@seccmd=#1\hbs@checkarg}%
990 }% \itf@hbs@do
991 \let\hbs@do\itf@hbs@do

```

\hbs@process (patch)

```

992 \itf@patch\pretocmd\hbs@process{%
993   \itf@hbs@process{#1}%
994 }%
995 {\@PackageInfoNoLine{interfaces-hypbmsec}
996   {Command \string\hbs@process\space successfully patched}%
997 {\@PackageWarningNoLine{interfaces-hypbmsec}
998   {Command \string\hbs@process\space could not be patched
999     \MessageBreak loading of ‘interfaces-hypbmsec’ aborted
1000    \MessageBreak \string\lastsection\space will not work}%
1001 \endinput}%

```

\itf@hbs@ssect For starred sections:

```

1002 \long\def\itf@hbs@ssect#1#2{%
1003   \itf@Not@titlesec#1
1004   {\expandafter\itf@hbs@last@starred\expandafter{\itf@hbs@currentname}{#2}%
1005   {\let\itf@hbs@last\itf@hbs@last@starred@ttl}%
1006   #1*{#2}%
1007 }% \itf@hbs@ssect

```

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

```

1008 \long\def\itf@hbs@process#1{%
1009   \itf@Not@titlesec \hbs@seccmd
1010   {%
1011     \ifdef\hyper@anchor
1012       \expandafter\itf@hbs@last@nostar@nottl@Hy\expandafter{\itf@hbs@currentname}{#1}%
1013     \else
1014       \expandafter\itf@hbs@last@nostar@nottl\expandafter{\itf@hbs@currentname}{#1}%
1015     \fi
1016   }%
1017   {\let\itf@hbs@last \itf@hbs@last@nostar@ttl}%
1018 }% \itf@hbs@process

```

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

```

1019 \long\def\itf@hbs@last@starred#1#2{%
1020   \GetTitleString{#2}\let\itf@hbs@currenttitle\GetTitleStringResult
1021   \begin{group}
1022     \let\@elt \itf@resetlastsections
1023     \globaldefs \one
1024     \expandafter\let\csname last#1title\endcsname\itf@hbs@currenttitle
1025     \expandafter\let\csname last#1toc\endcsname\empty
1026     \expandafter\let\csname last#1bookmark\endcsname\empty
1027     \expandafter\let\csname last#1anchor\endcsname\relax
1028     \let\lastsectionname \itf@hbs@currentname
1029     \csname cl@#1\endcsname
1030   \end{group}
1031 }% \itf@hbs@last@starred
1032 \def\itf@hbs@last@starred@ttl{%
1033   \global\let\itf@hbs@last \empty
1034   \expandafter\itf@hbs@last@starred
1035   \expandafter\itf@hbs@currentname \expandafter{\ttl@savetitle}%
1036 }% \itf@hbs@last@starred@ttl

```

\itf@hbs@last@nostar Update de values of \lastsectiontitle etc.. for non-starred sections:

```

1037 \long\def\itf@hbs@last@nostar@ttl{%
1038   \GetTitleString{#2}\let\itf@hbs@currenttitle\GetTitleStringResult
1039   \itf@hbs@last@nostar@nottl@Hy

```

```

1039 \begingroup
1040   \let\@elt \itf@resetlastsections
1041   \globaldefs \ne
1042   \expandafter\let\csname last#1title\endcsname\itf@hbs@currenttitle
1043   \expandafter\let\csname last#1toc\expandafter\endcsname
1044     \csname \ifx\hbs@tocstring\relax
1045       \itf@hbs@currenttitle\else \hbs@tocstring\fi \endcsname
1046   \expandafter\let\csname last#1bookmark\expandafter\endcsname
1047     \csname \ifx\hbs@bmstring\relax
1048       \last#1toc\else \hbs@bmstring\fi \endcsname
1049   \expandafter\let\csname last#1anchor\endcsname\HyperLocalCurrentHref
1050   \let\lastsectionname \itf@hbs@currentname
1051   \csname cl@#1\endcsname
1052 \endgroup
1053 }% \itf@hbs@last@starred
1054 \def\itf@hbs@last@nostar@ttl{%
1055   \global\let\itf@hbs@last \empty
1056   \expandafter\itf@hbs@last@nostar
1057     \expandafter\itf@hbs@currentname \expandafter{\ttl@savetitle}%
1058 }% \itf@hbs@last@nostar@ttl
1059 \long\def\itf@hbs@last@nostar@nottl#1#2{%
1060   \expandafter\let\expandafter\itf@hbs@cl@ORI\csname cl@#1\endcsname
1061   \expandafter\def\csname cl@#1\endcsname{%
1062     \expandafter\let\csname cl@#1\endcsname\itf@hbs@cl@ORI
1063     \itf@hbs@cl@ORI
1064     \itf@hbs@last@nostar {#1}{#2}}%
1065 }% \itf@hbs@last@nostar@nottl
1066 \long\def\itf@hbs@last@nostar@nottl@Hy#1#2{%
1067   \begingroup
1068   \toks@\expandafter{%
1069     \expandafter\def\expandafter\hyper@anchorend
1070       \expandafter{\hyper@anchorend}%
1071     \itf@hbs@last@nostar {#1}{#2}}%
1072   \edef\x{\endgroup
1073     \def\noexpand\hyper@anchorend{\the\toks@}%
1074   }\x
1075 }% \itf@hbs@last@nostar@nottl@Hy

```

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

```

1076 \def\itf@Not@titlesec#1{\@firstoftwo}
1077 \AfterPackage*{titlesec}{%
1078 \def\itf@Not@titlesec#1{\begingroup
1079   @onelvel@sanitize#1%
1080   \edef\tempa{\noexpand\in@
1081     {\noexpand\@nil \string\ttl@}
1082     {\noexpand\@nil #1}%
1083   }\tempa
1084   \expandafter\endgroup
1085   \ifin@ \expandafter\@secondoftwo
1086   \else \expandafter\@firstoftwo
1087   \fi
1088 }% \itf@Not@titlesec
1089 }% AfterPackage titlesec

```

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

```

1090 \AfterPackage*{titlesec}{%
1091 \expandafter\def\expandafter\ttl@finmarks\expandafter{%
1092   \expandafter\itf@hbs@last \ttl@finmarks}%
1093 \let\itf@hbs@last \empty
1094 }% AfterPackage

```

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

```

1095 \def\itf@resetlastsections#1{%
1096   \expandafter\let\csname last#1title\endcsname\relax
1097   \expandafter\let\csname last#1toc\endcsname\relax
1098   \expandafter\let\csname last#1bookmark\endcsname\relax
1099   \expandafter\let\csname last#1anchor\endcsname\relax
1100   \csname cl#1\endcsname
1101 }% \itf@resetlastsections

1102 \let\lastsectionname\relax
1103 \@for\itf@tp:=part,section,subsection,subsubsection,paragraph,subparagraph%
1104   \do{\expandafter\let\csname last\itf@tp title\endcsname\relax
1105     \expandafter\let\csname last\itf@tp toc\endcsname\relax
1106     \expandafter\let\csname last\itf@tp bookmark\endcsname\relax
1107     \expandafter\let\csname last\itf@tp anchor\endcsname\relax
1108   }%
1109 \AtBeginDocument{\ifundef\chapter{}%
1110   {\let\lastchaptertitle\relax
1111   \let\lastchaptertos\relax
1112   \let\lastchapterbookmarks\relax
1113   \let\lastchapteranchors\relax}%
1114 }%

1115 
```

21.9 `fancyhdr` : `\pagesetup`

```

1116 {*fancyhdr}
1117 \ProvidesPackage{interfaces-fancyhdr}
1118   [2010/12/07 v2.5 pgfkeys macros for fancyhdr (FC)]
1119 \RequirePackage{interfaces-base}
1120 \itf@Check@Package{fancyhdr}
1121 \itf@pgfkeys
1122 \robustify\fancyhf
1123 \robustify\fancyhfoffset

```

21.9.1 `\pagesetup`

`\pagesetup`

```

1124 \newrobustcmd*\pagesetup{%
1125   \let\fancy@gbl\relax % Just in case...
1126   \@ifstar
1127     {\itf@pagesetup{}}
1128     {\itf@pagesetup{reset,}}%
1129 }% \pagesetup
1130 \def\itf@pagesetup#1{\@testopt
1131   {\itf@p@gesetup{#1}}{%
1132 }% \itf@pagesetup
1133 \def\itf@p@gesetup#1[#2]{\ifblank{#2}{%
1134   % modifies the current page style
1135   \itf@pgfkeys\pgfqkeys{/itf/fancyhdr}{#1#3}\relax\ps@@fancy}%
1136   % defines or modifies a given page style
1137   \ifblank{#1}{%
1138     \csappto% star form: keep existing value (if not overwritten)
1139     \csdef% no star form: overwrite (+ reset at \pagestyle time)
1140     {ps@#2}{\itf@pgfkeys\pgfqkeys{/itf/fancyhdr}{#1#3}\relax\ps@@fancy}%
1141   }%
1142 }% \itf@p@gesetup
1143 \pgfqkeys{/itf/pagesetup}{%
1144   @current/.is private=interfaces-fancyhdr,%
1145   @style/.is private=interfaces-fancyhdr,%
1146 }

```

`\pagestyle`

```

1147 \renewrobustcmd*\pagestyle[1]{\ifcsundef{ps@#1}
1148   {\itf@undefinedpagestyle{#1}}
1149   {\cscname ps@#1\endcscname}%
1150 }% \pagestyle

\thispagestyle

1151 \renewrobustcmd*\thispagestyle[1]{\ifcsundef{ps@#1}
1152   {\itf@undefinedpagestyle{#1}}
1153   {\global\@specialpagetrue\gdef\@specialstyle{#1}}%
1154 }% \thispagestyle
1155 \newcommand*\itf@undefinedpagestyle[1]{%
1156   \@latex@error{Undefined page style: #1}%
1157   \MessageBreak your command will be ignored}\@eha
1158 }% \itf@undefinedpagestyle

```

21.9.2 Keys for \pagesetup

```

1159 \pgfkeys{%
1160   /handlers/.fancy@hf/.code=%
1161   \edef\itf@tp{\pgfkeyscurrentpath}%
1162   \pgfkeysalso{\pgfkeyscurrentpath/.store as code}%
1163   \pgfkeysalso{\pgfkeyscurrentpath/.append code/.expanded=%
1164     \csdef{f@ncy#1}{%
1165       \noexpand\pgfkeysvalueof{\itf@tp/font}%
1166       \noexpand\pgfkeysvalueof{\itf@tp}\noexpand\strut},%
1167 }% handler for fancyhdr interface
1168 \pgfqkeys{/itf/fancyhdr}{% keys for \pagesetup
1169   code/.code={#1},% execution at expansion of \ps@... => \pgfqkeys{...,code={...},...}
1170   page style/.style={code=\pagestyle{#1}},%
1171   page style/.alias=pagestyle,%
1172   load style/.alias=pagestyle,%
1173   head/rule/.store as dimen in cs=headrulewidth,%
1174   head/rule/.default=.4pt,%
1175   head/rule/.blank=\z@,%
1176   head/rulewidth/.alias*=head/rule,%
1177   headrulewidth/.alias*=head/rule,%
1178   head/rule/width/.alias*=head/rule,%
1179   head/rule/font/.store as code,%
1180   head/rule/color/.style={head/rule/font+=\color{#1}},%
1181   headruler/.code=%
1182   \let\headrule\pgfkeyscurrentvalue
1183   \ifx\headrule\empty\else
1184     \preto\headrule{\bgroup\pgfkeysvalueof{/itf/fancyhdr/head/rule/font}}%
1185     \appto\headrule{\egroup}%
1186   \fi,%
1187   headruler/.default=%
1188   \if\fancyplain\let\headrulewidth\plainheadrulewidth\fi
1189   \hrule@height\headrulewidth \width\headwidth
1190   \vskip-\headrulewidth\relax,%
1191   headrule/.alias=headruler,%
1192   headrule,% sets default
1193   foot/rule/.store as dimen in cs=footrulewidth,%
1194   foot/rule/.default=.4pt,%
1195   foot/rule/.blank=\z@,%
1196   foot/rulewidth/.alias*=foot/rule,%
1197   footrulewidth/.alias*=foot/rule,%
1198   foot/rule/width/.alias*=foot/rule,%
1199   foot/rule/font/.store as code,%
1200   foot/rule/color/.style={foot/rule/font+=\color{#1}},%
1201   footruler/.code=%
1202   \let\footrule\pgfkeyscurrentvalue
1203   \ifx\footrule\empty\else
1204     \preto\footrule{\bgroup\pgfkeysvalueof{/itf/fancyhdr/head/rule/font}}%

```

```

1205   \appto\footrule{\egroup}
1206   \fi,%
1207   footruler/.default=%
1208     \if@fancyplain\let\footrulewidth\plainfootrulewidth\fi
1209     \vskip-2\footrulewidth
1210     \hrule@height\footrulewidth \@width\headwidth\vskip\footrulewidth,%
1211   footrule/.alias=footruler,%
1212   footrule,% sets default
1213   rule/.style={head/rule={#1},foot/rule={#1}},%
1214   rule+/.style={head/rule+={#1},foot/rule+={#1}},%
1215   rule-/.style={head/rule-={#1},foot/rule-={#1}},%
1216   rule*/.style={head/rule*={#1},foot/rule*={#1}},%
1217     rules/.alias*=rule,rulewidth/.alias*=rule,%
1218     rule/width/.alias*=rule,rules/width/.alias*=rule,%
1219   norule/.style={rule=\z@},%
1220   norule/.value forbidden,%
1221     norules/.alias=norule,%
1222   rule/font/.style={head/rule/font={#1},foot/rule/font={#1}},%
1223   rule/font+/.style={head/rule/font+={#1},foot/rule/font+={#1}},%
1224     rule/font*/.alias=rule/font+,%
1225     rules/font/.alias=rule/font,
1226     rules/font+/.alias=rule/font+,%
1227     rules/font*/.alias=rule/font+,%
1228   rule/color/.style={head/rule/color={#1},foot/rule/color={#1}},%
1229     rules/color/.alias=rule/color,%
1230   foot/rule/skip/.estore as glue in cs=footruleskip,%
1231     foot/rule/skip/.default=.3\normalbaselineskip,%
1232     foot/rule/skip/.blank=\z@,%
1233     foot/ruleskip/.alias*=foot/rule/skip,%
1234     footruleskip/.alias*=foot/rule/skip,%
1235   font/.style={head/font={#1},foot/font={#1}},%
1236   font+/.style={head/font+={#1},foot/font+={#1}},%
1237     font*/.alias=font+,%
1238   color/.style={head/color={#1},foot/color={#1}},%
1239     head/rule/color={#1},foot/rule/color={#1},%
1240   head/.style={head/left={#1},head/center={#1},head/right={#1}},%
1241   head+/.style={head/left+={#1},head/center+={#1},head/right+={#1}},%
1242   head*/.alias=head+,%
1243     head/font/.style={head/left/font={#1},head/center/font={#1},head/right/font={#1}},%
1244     head/font+/.style={head/left/font+={#1},head/center/font+={#1},head/right/font+={#1}},%
1245     head/font*/.alias=head/font+,%
1246     head/color/.style={head/left/color={#1},head/center/color={#1},head/right/color={#1}},%
1247       head/rule/color={#1},%
1248   nohead/.style={head=,head/font=,head/rule=,head/rule/font=},
1249   foot/.style={foot/left={#1},foot/center={#1},foot/right={#1}},%
1250   foot+/.style={foot/left+={#1},foot/center+={#1},foot/right+={#1}},%
1251   foot*/.alias=foot+,%
1252     foot/font/.style={foot/left/font={#1},foot/center/font={#1},foot/right/font={#1}},%
1253     foot/font+/.style={foot/left/font+={#1},foot/center/font+={#1},foot/right/font+={#1}},%
1254     foot/font*/.alias=foot/font+,%
1255     foot/color/.style={foot/left/color={#1},foot/center/color={#1},foot/right/color={#1}},%
1256       foot/rule/color={#1},%
1257   nofoot/.style={foot=,foot/font=,foot/rule=,foot/rule/font=},
1258   left/.style={head/left={#1},foot/left={#1}},%
1259   left+/.style={head/left+={#1},foot/left+={#1}},%
1260   left*/.alias=left+,%
1261     left/font/.style={head/left/font={#1},foot/left/font={#1}},%
1262     left/font+/.style={head/left/font+={#1},foot/left/font+={#1}},%
1263     left/font*/.alias=left/font+,%
1264     left/color/.style={head/left/color={#1},foot/left/color={#1}},%
1265   center/.style={head/center={#1},foot/center={#1}},%
1266   center+/.style={head/center+={#1},foot/center+={#1}},%
1267   center*/.alias=center+,%
1268     center/font/.style={head/center/font={#1},foot/center/font={#1}},%
1269     center/font+/.style={head/center/font+={#1},foot/center/font+={#1}},%

```

```

1270     center/font*.alias=center/font+,%
1271     center/color/.style={head/center/color={#1},foot/center/color={#1}},%
1272     right/.style={head/right={#1},foot/right={#1}},%
1273     right+.style={head/right+={#1},foot/right+={#1}},%
1274     right*/.alias=right+,%
1275         right/font/.style={head/right/font={#1},foot/right/font={#1}},%
1276         right/font+.style={head/right/font+={#1},foot/right/font+={#1}},%
1277         right/font*/.alias=right/font+,%
1278         right/color/.style={head/right/color={#1},foot/right/color={#1}},%
1279     head/left/.style={head/left/even={#1},head/left/odd={#1}},%
1280     head/left+.style={head/left/even+={#1},head/left/odd+={#1}},%
1281     head/left*/.alias=head/left+,%
1282         head/left/font/.style={head/left/even/font={#1},head/left/odd/font={#1}},%
1283         head/left/font+.style={head/left/even/font+={#1},head/left/odd/font+={#1}},%
1284         head/left/font*/.alias=head/left/font+,%
1285         head/left/color/.style={head/left/even/color={#1},head/left/odd/color={#1}},%
1286             left/head/.alias*=head/left,%
1287             left/head/font/.alias*=head/left/font,%
1288             left/head/color/.alias=head/left/color,%
1289     head/center/.style={head/center/even={#1},head/center/odd={#1}},%
1290     head/center+.style={head/center/even+={#1},head/center/odd+={#1}},%
1291     head/center*/.alias=head/center+,%
1292         head/center/font/.style={head/center/even/font={#1},head/center/odd/font={#1}},%
1293         head/center/font+.style={head/center/even/font+={#1},head/center/odd/font+={#1}},%
1294         head/center/font*/.alias=head/center/font+,%
1295         head/center/color/.style={head/center/even/color={#1},head/center/odd/color={#1}},%
1296             center/head/.alias*=head/center,%
1297             center/head/font/.alias*=head/center/font,%
1298             center/head/color/.alias=head/center/color,%
1299     head/right/.style={head/right/even={#1},head/right/odd={#1}},%
1300     head/right+.style={head/right/even+={#1},head/right/odd+={#1}},%
1301     head/right*/.alias=head/right+,%
1302         head/right/font/.style={head/right/even/font={#1},head/right/odd/font={#1}},%
1303         head/right/font+.style={head/right/even/font+={#1},head/right/odd/font+={#1}},%
1304         head/right/font*/.alias=head/right/font+,%
1305         head/right/color/.style={head/right/even/color={#1},head/right/odd/color={#1}},%
1306             right/head/.alias*=head/right,%
1307             right/head/font/.alias*=head/right/font,%
1308             right/head/color/.alias=head/right/color,%
1309     foot/left/.style={foot/left/even={#1},foot/left/odd={#1}},%
1310     foot/left+.style={foot/left/even+={#1},foot/left/odd+={#1}},%
1311     foot/left*/.alias=foot/left+,%
1312         foot/left/font/.style={foot/left/even/font={#1},foot/left/odd/font={#1}},%
1313         foot/left/font+.style={foot/left/even/font+={#1},foot/left/odd/font+={#1}},%
1314         foot/left/font*/.alias=foot/left/font+,%
1315         foot/left/color/.style={foot/left/even/color={#1},foot/left/odd/color={#1}},%
1316             left/foot/.alias*=foot/left,%
1317             left/foot/font/.alias*=foot/left/font,%
1318             left/foot/color/.alias=foot/left/color,%
1319     foot/center/.style={foot/center/even={#1},foot/center/odd={#1}},%
1320     foot/center+.style={foot/center/even+={#1},foot/center/odd+={#1}},%
1321     foot/center*/.alias=foot/center+,%
1322         foot/center/font/.style={foot/center/even/font={#1},foot/center/odd/font={#1}},%
1323         foot/center/font+.style={foot/center/even/font+={#1},foot/center/odd/font+={#1}},%
1324         foot/center/font*/.alias=foot/center/font+,%
1325         foot/center/color/.style={foot/center/even/color={#1},foot/center/odd/color={#1}},%
1326             center/foot/.alias*=foot/center,%
1327             center/foot/font/.alias*=foot/center/font,%
1328             center/foot/color/.alias=foot/center/color,%
1329     foot/right/.style={foot/right/even={#1},foot/right/odd={#1}},%
1330     foot/right+.style={foot/right/even+={#1},foot/right/odd+={#1}},%
1331     foot/right*/.alias=foot/right+,%
1332         foot/right/font/.style={foot/right/even/font={#1},foot/right/odd/font={#1}},%
1333         foot/right/font+.style={foot/right/even/font+={#1},foot/right/odd/font+={#1}},%
1334         foot/right/font*/.alias=foot/right/font,%

```

```

1335     foot/right/color/.style={foot/right/even/color={#1},foot/right/odd/color={#1}},%
1336         right/foot/.alias*=foot/right,%
1337         right/foot/font/.alias*=foot/right/font,%
1338         right/foot/color/.alias=foot/right/color,%
1339     head/left/odd/.fancy@hf=olh,%
1340     head/left/odd/font/.store as code,%
1341     head/left/odd/color/.style={head/left/odd/font+=\color{#1}},%
1342         head/odd/left/.alias*=head/left/odd,%
1343         odd/head/left/.alias*=head/left/odd,%
1344         odd/left/head/.alias*=head/left/odd,%
1345         left/odd/head/.alias*=head/left/odd,%
1346         left/head/odd/.alias*=head/left/odd,%
1347         head/odd/left/font/.alias*=head/left/odd/font,%
1348         odd/head/left/font/.alias*=head/left/odd/font,%
1349         odd/left/head/font/.alias*=head/left/odd/font,%
1350         left/odd/head/font/.alias*=head/left/odd/font,%
1351         left/head/odd/font/.alias*=head/left/odd/font,%
1352     head/odd/left/color/.alias=head/left/odd/color,%
1353         odd/head/left/color/.alias=head/left/odd/color,%
1354         odd/left/head/color/.alias=head/left/odd/color,%
1355         left/odd/head/color/.alias=head/left/odd/color,%
1356         left/head/odd/color/.alias=head/left/odd/color,%
1357     head/center/odd/.fancy@hf=och,%
1358     head/center/odd/font/.store as code,%
1359     head/center/odd/color/.style={head/center/odd/font+=\color{#1}},%
1360         head/odd/center/.alias*=head/center/odd,%
1361         odd/head/center/.alias*=head/center/odd,%
1362         odd/center/head/.alias*=head/center/odd,%
1363         center/odd/head/.alias*=head/center/odd,%
1364         center/head/odd/.alias*=head/center/odd,%
1365         head/odd/center/font/.alias*=head/center/odd/font,%
1366         odd/head/center/font/.alias*=head/center/odd/font,%
1367         odd/center/head/font/.alias*=head/center/odd/font,%
1368         center/odd/head/font/.alias*=head/center/odd/font,%
1369         center/head/odd/font/.alias*=head/center/odd/font,%
1370     head/odd/center/color/.alias=head/center/odd/color,%
1371         odd/head/center/color/.alias=head/center/odd/color,%
1372         odd/center/head/color/.alias=head/center/odd/color,%
1373         center/odd/head/color/.alias=head/center/odd/color,%
1374         center/head/odd/color/.alias=head/center/odd/color,%
1375     head/right/odd/.fancy@hf=orh,%
1376     head/right/odd/font/.store as code,%
1377     head/right/odd/color/.style={head/right/odd/font+=\color{#1}},%
1378         head/odd/right/.alias*=head/right/odd,%
1379         odd/head/right/.alias*=head/right/odd,%
1380         odd/right/head/.alias*=head/right/odd,%
1381         right/odd/head/.alias*=head/right/odd,%
1382         right/head/odd/.alias*=head/right/odd,%
1383         head/odd/right/font/.alias*=head/right/odd/font,%
1384         odd/head/right/font/.alias*=head/right/odd/font,%
1385         odd/right/head/font/.alias*=head/right/odd/font,%
1386         right/odd/head/font/.alias*=head/right/odd/font,%
1387         right/head/odd/font/.alias*=head/right/odd/font,%
1388     head/odd/right/color/.alias=head/right/odd/color,%
1389         odd/head/right/color/.alias=head/right/odd/color,%
1390         odd/right/head/color/.alias=head/right/odd/color,%
1391         right/odd/head/color/.alias=head/right/odd/color,%
1392         right/head/odd/color/.alias=head/right/odd/color,%
1393     head/left/even/.fancy@hf=elh,%
1394     head/left/even/font/.store as code,%
1395     head/left/even/color/.style={head/left/even/font+=\color{#1}},%
1396         head/even/left/.alias*=head/left/even,%
1397         even/head/left/.alias*=head/left/even,%
1398         even/left/head/.alias*=head/left/even,%
1399         left/even/head/.alias*=head/left/even,%

```

```

1400    left/head/even/.alias*=head/left/even,%
1401    head/even/left/font/.alias*=head/left/even/font,%
1402    even/head/left/font/.alias*=head/left/even/font,%
1403    even/left/head/font/.alias*=head/left/even/font,%
1404    left/even/head/font/.alias*=head/left/even/font,%
1405    left/head/even/font/.alias*=head/left/even/font,%
1406    head/even/left/color/.alias=head/left/even/color,%
1407    even/head/left/color/.alias=head/left/even/color,%
1408    even/left/head/color/.alias=head/left/even/color,%
1409    left/even/head/color/.alias=head/left/even/color,%
1410    left/head/even/color/.alias=head/left/even/color,%
1411    head/center/even/.fancy@hf=ech,%
1412    head/center/even/font/.store as code,%
1413    head/center/even/color/.style={head/center/even/font+\color{#1}},%
1414        head/center/.alias*=head/center/even,%
1415        even/center/.alias*=head/center/even,%
1416        even/center/head/.alias*=head/center/even,%
1417        center/even/head/.alias*=head/center/even,%
1418        center/head/even/.alias*=head/center/even,%
1419        head/even/center/font/.alias*=head/center/even/font,%
1420        even/head/center/font/.alias*=head/center/even/font,%
1421        even/center/head/font/.alias*=head/center/even/font,%
1422        center/even/head/font/.alias*=head/center/even/font,%
1423        center/head/even/font/.alias*=head/center/even/font,%
1424        head/even/center/color/.alias=head/center/even/color,%
1425        even/head/center/color/.alias=head/center/even/color,%
1426        even/center/head/color/.alias=head/center/even/color,%
1427        center/even/head/color/.alias=head/center/even/color,%
1428        center/head/even/color/.alias=head/center/even/color,%
1429    head/right/even/.fancy@hf=erh,%
1430    head/right/even/font/.store as code,%
1431    head/right/even/color/.style={head/right/even/font+\color{#1}},%
1432        head/even/right/.alias*=head/right/even,%
1433        even/head/right/.alias*=head/right/even,%
1434        even/right/head/.alias*=head/right/even,%
1435        right/even/head/.alias*=head/right/even,%
1436        right/head/even/.alias*=head/right/even,%
1437        head/even/right/font/.alias*=head/right/even/font,%
1438        even/head/right/font/.alias*=head/right/even/font,%
1439        even/right/head/font/.alias*=head/right/even/font,%
1440        right/even/head/font/.alias*=head/right/even/font,%
1441        right/head/even/font/.alias*=head/right/even/font,%
1442        head/even/right/color/.alias=head/right/even/color,%
1443        even/head/right/color/.alias=head/right/even/color,%
1444        even/right/head/color/.alias=head/right/even/color,%
1445        right/even/head/color/.alias=head/right/even/color,%
1446        right/head/even/color/.alias=head/right/even/color,%
1447    foot/left/odd/.fancy@hf=olf,%
1448    foot/left/odd/font/.store as code,%
1449    foot/left/odd/color/.style={foot/left/odd/font+\color{#1}},%
1450        foot/odd/left/.alias*=foot/left/odd,%
1451        odd/foot/left/.alias*=foot/left/odd,%
1452        odd/left/foot/.alias*=foot/left/odd,%
1453        left/odd/foot/.alias*=foot/left/odd,%
1454        left/foot/odd/.alias*=foot/left/odd,%
1455        foot/odd/left/font/.alias*=foot/left/odd/font,%
1456        odd/foot/left/font/.alias*=foot/left/odd/font,%
1457        odd/left/foot/font/.alias*=foot/left/odd/font,%
1458        left/odd/foot/font/.alias*=foot/left/odd/font,%
1459        left/foot/odd/font/.alias*=foot/left/odd/font,%
1460        foot/odd/left/color/.alias=foot/left/odd/color,%
1461        odd/foot/left/color/.alias=foot/left/odd/color,%
1462        odd/left/foot/color/.alias=foot/left/odd/color,%
1463        left/odd/foot/color/.alias=foot/left/odd/color,%
1464        left/foot/odd/color/.alias=foot/left/odd/color,%

```

```
1465   foot/center/odd/.fancy@hf=ocf,%
1466   foot/center/odd/font/.store as code,%
1467   foot/center/odd/color/.style={foot/center/odd/font+=\color{#1}},%
1468     foot/odd/center/.alias*=foot/center/odd,%
1469     odd/foot/center/.alias*=foot/center/odd,%
1470     odd/center/foot/.alias*=foot/center/odd,%
1471     center/odd/foot/.alias*=foot/center/odd,%
1472     center/foot/odd/.alias*=foot/center/odd,%
1473     foot/odd/center/font/.alias*=foot/center/odd/font,%
1474     odd/foot/center/font/.alias*=foot/center/odd/font,%
1475     odd/center/foot/font/.alias*=foot/center/odd/font,%
1476     center/odd/foot/font/.alias*=foot/center/odd/font,%
1477     center/foot/odd/font/.alias*=foot/center/odd/font,%
1478     foot/odd/center/color/.alias=foot/center/odd/color,%
1479     odd/foot/center/color/.alias=foot/center/odd/color,%
1480     odd/center/foot/color/.alias=foot/center/odd/color,%
1481     center/odd/foot/color/.alias=foot/center/odd/color,%
1482     center/foot/odd/color/.alias=foot/center/odd/color,%
1483   foot/right/odd/.fancy@hf=orf,%
1484   foot/right/odd/font/.store as code,%
1485   foot/right/odd/color/.style={foot/right/odd/font+=\color{#1}},%
1486     foot/odd/right/.alias*=foot/right/odd,%
1487     odd/foot/right/.alias*=foot/right/odd,%
1488     odd/right/foot/.alias*=foot/right/odd,%
1489     right/odd/foot/.alias*=foot/right/odd,%
1490     right/foot/odd/.alias*=foot/right/odd,%
1491     foot/odd/right/font/.alias*=foot/right/odd/font,%
1492     odd/foot/right/font/.alias*=foot/right/odd/font,%
1493     odd/right/foot/font/.alias*=foot/right/odd/font,%
1494     right/odd/foot/font/.alias*=foot/right/odd/font,%
1495     right/foot/odd/font/.alias*=foot/right/odd/font,%
1496     foot/odd/right/color/.alias=foot/right/odd/color,%
1497     odd/foot/right/color/.alias=foot/right/odd/color,%
1498     odd/right/foot/color/.alias=foot/right/odd/color,%
1499     right/odd/foot/color/.alias=foot/right/odd/color,%
1500     right/foot/odd/color/.alias=foot/right/odd/color,%
1501   foot/left/even/.fancy@hf=elf,%
1502   foot/left/even/font/.store as code,%
1503   foot/left/even/color/.style={foot/left/even/font+=\color{#1}},%
1504     foot/even/left/.alias*=foot/left/even,%
1505     even/foot/left/.alias*=foot/left/even,%
1506     even/left/foot/.alias*=foot/left/even,%
1507     left/even/foot/.alias*=foot/left/even,%
1508     left/foot/even/.alias*=foot/left/even,%
1509     foot/even/left/font/.alias*=foot/left/even/font,%
1510     even/foot/left/font/.alias*=foot/left/even/font,%
1511     even/left/foot/font/.alias*=foot/left/even/font,%
1512     left/even/foot/font/.alias*=foot/left/even/font,%
1513     left/foot/even/font/.alias*=foot/left/even/font,%
1514     foot/even/left/color/.alias=foot/left/even/color,%
1515     even/foot/left/color/.alias=foot/left/even/color,%
1516     even/left/foot/color/.alias=foot/left/even/color,%
1517     left/even/foot/color/.alias=foot/left/even/color,%
1518     left/foot/even/color/.alias=foot/left/even/color,%
1519   foot/center/even/.fancy@hf=ecf,%
1520   foot/center/even/font/.store as code,%
1521   foot/center/even/color/.style={foot/center/even/font+=\color{#1}},%
1522     foot/even/center/.alias*=foot/center/even,%
1523     even/foot/center/.alias*=foot/center/even,%
1524     even/center/foot/.alias*=foot/center/even,%
1525     center/even/foot/.alias*=foot/center/even,%
1526     center/foot/even/.alias*=foot/center/even,%
1527     foot/even/center/font/.alias*=foot/center/even/font,%
1528     even/foot/center/font/.alias*=foot/center/even/font,%
1529     even/center/foot/font/.alias*=foot/center/even/font,%
```

```

1530     center/even/foot/.alias*=foot/center/even/font,%
1531     center/foot/even/font/.alias*=foot/center/even/font,%
1532     foot/center/color/.alias=foot/center/even/color,%
1533     even/foot/center/color/.alias=foot/center/even/color,%
1534     even/center/foot/color/.alias=foot/center/even/color,%
1535     center/foot/color/.alias=foot/center/even/color,%
1536     center/foot/even/color/.alias=foot/center/even/color,%
1537     foot/right/even/.fancyhf=erf,%
1538     foot/right/even/font/.store as code,%
1539     foot/right/even/color/.style={foot/right/even/font+=\color{\#1}},%
1540         foot/even/right/.alias*=foot/right/even,%
1541         even/foot/right/.alias*=foot/right/even,%
1542         even/right/foot/.alias*=foot/right/even,%
1543         right/even/foot/.alias*=foot/right/even,%
1544         right/foot/even/.alias*=foot/right/even,%
1545         foot/even/right/font/.alias*=foot/right/even/font,%
1546         even/foot/right/font/.alias*=foot/right/even/font,%
1547         even/right/font/.alias*=foot/right/even/font,%
1548         right/even/foot/.alias*=foot/right/even/font,%
1549         right/foot/even/font/.alias*=foot/right/even/font,%
1550         foot/even/right/color/.alias=foot/right/even/color,%
1551         even/foot/right/color/.alias=foot/right/even/color,%
1552         even/right/foot/color/.alias=foot/right/even/color,%
1553         right/even/foot/color/.alias=foot/right/even/color,%
1554         right/foot/even/color/.alias=foot/right/even/color,%
1555     foot/inner/.style={foot/right/even=\#1,foot/left/odd=\#1},%
1556     foot/inner+/.style={foot/right/even=\#1,foot/left/odd=\#1},%
1557     foot/inner*/.alias=foot/inner+,%
1558     foot/inner/font/.style={foot/right/even/font=\#1,foot/left/odd/font=\#1},%
1559     foot/inner/color/.style={foot/right/even/color=\#1,foot/left/odd/color=\#1},%
1560     foot/inner/font+/.style={foot/right/even/font=\#1,foot/left/odd/font=\#1},%
1561     foot/inner/font*/.alias=foot/inner/font+,%
1562         inner/foot/.alias*=foot/inner,%
1563         inner/foot/font/.alias*=foot/inner/font,%
1564         inner/foot/color/.alias=foot/inner/color,%
1565     foot/outer/.style={foot/left/even=\#1,foot/right/odd=\#1},%
1566     foot/outer+/.style={foot/left/even=\#1,foot/right/odd=\#1},%
1567     foot/outer*/.alias=foot/outer+,%
1568     foot/outer/font/.style={foot/left/even/font=\#1,foot/right/odd/font=\#1},%
1569     foot/outer/color/.style={foot/left/even/color=\#1,foot/right/odd/color=\#1},%
1570     foot/outer/font+/.style={foot/left/even/font=\#1,foot/right/odd/font=\#1},%
1571     foot/outer/font*/.alias=foot/outer/font+,%
1572         outer/foot/.alias*=foot/outer,%
1573         outer/foot/font/.alias*=foot/outer/font,%
1574         outer/foot/color/.alias=foot/outer/color,%
1575     head/inner/.style={head/right/even=\#1,head/left/odd=\#1},%
1576     head/inner+/.style={head/right/even=\#1,head/left/odd=\#1},%
1577     head/inner*/.alias=head/inner+,%
1578     head/inner/font/.style={head/right/even/font=\#1,head/left/odd/font=\#1},%
1579     head/inner/color/.style={head/right/even/color=\#1,head/left/odd/color=\#1},%
1580     head/inner/font+/.style={head/right/even/font=\#1,head/left/odd/font=\#1},%
1581     head/inner/font*/.alias=head/inner/font+,%
1582         inner/head/.alias*=head/inner,%
1583         inner/head/font/.alias*=head/inner/font,%
1584         inner/head/color/.alias=head/inner/color,%
1585     head/outer/.style={head/left/even=\#1,head/right/odd=\#1},%
1586     head/outer+/.style={head/left/even=\#1,head/right/odd=\#1},%
1587     head/outer*/.alias=head/outer+,%
1588     head/outer/font/.style={head/left/even/font=\#1,head/right/odd/font=\#1},%
1589     head/outer/color/.style={head/left/even/color=\#1,head/right/odd/color=\#1},%
1590     head/outer/font+/.style={head/left/even/font=\#1,head/right/odd/font=\#1},%
1591     head/outer/font*/.alias=head/outer/font+,%
1592         outer/head/.alias*=head/outer,%
1593         outer/head/font/.alias*=head/outer/font,%
1594         outer/head/color/.alias=head/outer/color,%

```

```

1595 inner/.style={foot/inner={#1},head/inner={#1}},%
1596 inner+.style={foot/inner+=#1,head/inner+=#1},%
1597 inner*/.alias=inner+,%
1598 inner/font/.style={foot/inner/font={#1},head/inner/font={#1}},%
1599 inner/font+/.style={foot/inner/font+=#1,head/inner/font+=#1},%
1600 inner/font*/.alias=inner/font+,%
1601 inner/color/.style={foot/inner/color={#1},head/inner/color={#1}},%
1602 outer/.style={foot/outer={#1},head/outer={#1}},%
1603 outer+/.style={foot/outer+=#1,head/outer+=#1},%
1604 outer*/.alias=outer+,%
1605 outer/font/.style={foot/outer/font={#1},head/outer/font={#1}},%
1606 outer/font+/.style={foot/outer/font+=#1,head/outer/font+=#1},%
1607 outer/font*/.alias=outer/font+,%
1608 outer/color/.style={foot/outer/color={#1},head/outer/color={#1}},%
1609 offset/.style={head/offset={#1},foot/offset={#1}},%
1610     offset+.style={head/offset+=#1,foot/offset+=#1},%
1611     offset-.style={head/offset-=#1,foot/offset-=#1},%
1612     offset*/.style={head/offset*={#1},foot/offset*={#1}},%
1613 head/offset/.style={head/left/offset={#1},head/right/offset={#1}},%
1614     head/offset+/.style={head/left/offset+=#1,head/right/offset+=#1},%
1615     head/offset-/.style={head/left/offset-=#1,head/right/offset-=#1},%
1616     head/offset*/.style={head/left/offset*={#1},head/right/offset*={#1}},%
1617         offset/head/.alias*=head/offset,%
1618 foot/offset/.style={foot/left/offset={#1},foot/right/offset={#1}},%
1619     foot/offset+/.style={foot/left/offset+=#1,foot/right/offset+=#1},%
1620     foot/offset-/.style={foot/left/offset-=#1,foot/right/offset-=#1},%
1621     foot/offset*/.style={foot/left/offset*={#1},foot/right/offset*={#1}},%
1622         offset/foot/.alias*=foot/offset,%
1623 odd/offset/.style={left/odd/offset={#1},right/odd/offset={#1}},%
1624     odd/offset+/.style={left/odd/offset+=#1,right/odd/offset+=#1},%
1625     odd/offset-/.style={left/odd/offset-=#1,right/odd/offset-=#1},%
1626         offset/odd/.alias*=odd/offset,%
1627 even/offset/.style={left/even/offset={#1},right/even/offset={#1}},%
1628     even/offset+/.style={left/even/offset+=#1,right/even/offset+=#1},%
1629     even/offset-/.style={left/even/offset-=#1,right/even/offset-=#1},%
1630         offset/even/.alias*=even/offset,%
1631 left/offset/.style={foot/left/offset={#1},head/left/offset={#1}},%
1632     left/offset+/.style={foot/left/offset+=#1,head/left/offset+=#1},%
1633     left/offset-/.style={foot/left/offset-=#1,head/left/offset-=#1},%
1634     left/offset*/.style={foot/left/offset*={#1},head/left/offset*={#1}},%
1635         offset/left/.alias*=left/offset,%
1636 left/even/offset/.style={foot/left/even/offset={#1},head/left/even/offset={#1}},%
1637     left/even/offset+/.style={foot/left/even/offset+=#1,head/left/even/offset+=#1},%
1638     left/even/offset-/.style={foot/left/even/offset-=#1,head/left/even/offset-=#1},%
1639     left/even/offset*/.style={foot/left/even/offset*={#1},head/left/even/offset*={#1}},%
1640         left/offset/even/.alias*=left/even/offset,%
1641         even/left/offset/.alias*=left/even/offset,%
1642             even/offset/left/.alias*=left/even/offset,%
1643                 offset/even/left/.alias*=left/even/offset,%
1644                 offset/left/even/.alias*=left/even/offset,%
1645 left/odd/offset/.style={foot/left/odd/offset={#1},head/left/odd/offset={#1}},%
1646     left/odd/offset+/.style={foot/left/odd/offset+=#1,head/left/odd/offset+=#1},%
1647     left/odd/offset-/.style={foot/left/odd/offset-=#1,head/left/odd/offset-=#1},%
1648     left/odd/offset*/.style={foot/left/odd/offset*={#1},head/left/odd/offset*={#1}},%
1649         left/offset/odd/.alias*=left/odd/offset,%
1650         odd/left/offset/.alias*=left/odd/offset,%
1651             odd/offset/left/.alias*=left/odd/offset,%
1652                 offset/odd/left/.alias*=left/odd/offset,%
1653                 offset/left/odd/.alias*=left/odd/offset,%
1654 right/offset/.style={foot/right/offset={#1},head/right/offset={#1}},%
1655     right/offset+/.style={foot/right/offset+=#1,head/right/offset+=#1},%
1656     right/offset-/.style={foot/right/offset-=#1,head/right/offset-=#1},%
1657     right/offset*/.style={foot/right/offset*={#1},head/right/offset*={#1}},%
1658         offset/right/.alias*=right/offset,%
1659 right/even/offset/.style={foot/right/even/offset={#1},head/right/even/offset={#1}},%

```

```

1660     right/even/offset+/.style={foot/right/even/offset+={#1},head/right/even/offset+={#1}},%
1661     right/even/offset-/.style={foot/right/even/offset-={#1},head/right/even/offset-={#1}},%
1662     right/even/offset*/.style={foot/right/even/offset*={#1},head/right/even/offset*={#1}},%
1663         right/offset/even/.alias*=right/even/offset,%
1664         even/right/offset/.alias*=right/even/offset,%
1665         even/offset/right/.alias*=right/even/offset,%
1666         offset/even/right/.alias*=right/even/offset,%
1667         offset/right/even/.alias*=right/even/offset,%
1668     right/odd/offset/.style={foot/right/odd/offset={#1},head/right/odd/offset={#1}},%
1669         right/odd/offset+/.style={foot/right/odd/offset+={#1},head/right/odd/offset+={#1}},%
1670         right/odd/offset-/.style={foot/right/odd/offset-={#1},head/right/odd/offset-={#1}},%
1671         right/odd/offset*/.style={foot/right/odd/offset*={#1},head/right/odd/offset*={#1}},%
1672             right/offset/odd/.alias*=right/odd/offset,%
1673             odd/right/offset/.alias*=right/odd/offset,%
1674             odd/offset/right/.alias*=right/odd/offset,%
1675             offset/odd/right/.alias*=right/odd/offset,%
1676             offset/right/odd/.alias*=right/odd/offset,%
1677     head/left/offset/.style={head/left/even/offset={#1},head/left/odd/offset={#1}},%
1678         head/left/offset+/.style={head/left/even/offset+={#1},head/left/odd/offset+={#1}},%
1679         head/left/offset-/.style={head/left/even/offset-={#1},head/left/odd/offset-={#1}},%
1680         head/left/offset*/.style={head/left/even/offset*={#1},head/left/odd/offset*={#1}},%
1681             head/offset/left/.alias*=head/left/offset,%
1682             left/head/offset/.alias*=head/left/offset,%
1683             left/offset/head/.alias*=head/left/offset,%
1684             offset/head/left/.alias*=head/left/offset,%
1685             offset/left/head/.alias*=head/left/offset,%
1686     head/right/offset/.style={head/right/even/offset={#1},head/right/odd/offset={#1}},%
1687         head/right/offset+/.style={head/right/even/offset+={#1},head/right/odd/offset+={#1}},%
1688         head/right/offset-/.style={head/right/even/offset-={#1},head/right/odd/offset-={#1}},%
1689         head/right/offset*/.style={head/right/even/offset*={#1},head/right/odd/offset*={#1}},%
1690             head/offset/right/.alias*=head/right/offset,%
1691             right/head/offset/.alias*=head/right/offset,%
1692             right/offset/head/.alias*=head/right/offset,%
1693             offset/head/right/.alias*=head/right/offset,%
1694             offset/right/head/.alias*=head/right/offset,%
1695     foot/left/offset/.style={foot/left/even/offset={#1},foot/left/odd/offset={#1}},%
1696         foot/left/offset+/.style={foot/left/even/offset+={#1},foot/left/odd/offset+={#1}},%
1697         foot/left/offset-/.style={foot/left/even/offset-={#1},foot/left/odd/offset-={#1}},%
1698         foot/left/offset*/.style={foot/left/even/offset*={#1},foot/left/odd/offset*={#1}},%
1699             foot/offset/left/.alias*=foot/left/offset,%
1700             left/foot/offset/.alias*=foot/left/offset,%
1701             left/offset/foot/.alias*=foot/left/offset,%
1702             offset/foot/left/.alias*=foot/left/offset,%
1703             offset/left/foot/.alias*=foot/left/offset,%
1704     foot/right/offset/.style={foot/right/even/offset={#1},foot/right/odd/offset={#1}},%
1705         foot/right/offset+/.style={foot/right/even/offset+={#1},foot/right/odd/offset+={#1}},%
1706         foot/right/offset-/.style={foot/right/even/offset-={#1},foot/right/odd/offset-={#1}},%
1707         foot/right/offset*/.style={foot/right/even/offset*={#1},foot/right/odd/offset*={#1}},%
1708             foot/offset/right/.alias*=foot/right/offset,%
1709             right/foot/offset/.alias*=foot/right/offset,%
1710             right/offset/foot/.alias*=foot/right/offset,%
1711             offset/foot/right/.alias*=foot/right/offset,%
1712             offset/right/foot/.alias*=foot/right/offset,%
1713     head/left/even/offset/.store in skip cs=f@ncy0@elh,%
1714         head/left/even/offset/.append code=\fancy@setoffs,%
1715             head/even/offset/left/.alias*=head/left/even/offset,%
1716             head/even/left/offset/.alias*=head/left/even/offset,%
1717             head/offset/even/left/.alias*=head/left/even/offset,%
1718             head/offset/left/even/.alias*=head/left/even/offset,%
1719             head/left/offset/even/.alias*=head/left/even/offset,%
1720             left/head/offset/even/.alias*=head/left/even/offset,%
1721             left/head/even/offset/.alias*=head/left/even/offset,%
1722             left/offset/head/even/.alias*=head/left/even/offset,%
1723             left/offset/even/head/.alias*=head/left/even/offset,%
1724             left/even/offset/head/.alias*=head/left/even/offset,%

```

```

1725   left/even/head/offset/.alias*=head/left/even/offset,%
1726   even/head/offset/left/.alias*=head/left/even/offset,%
1727   even/head/left/offset/.alias*=head/left/even/offset,%
1728   even/offset/head/left/.alias*=head/left/even/offset,%
1729   even/offset/left/head/.alias*=head/left/even/offset,%
1730   even/left/offset/head/.alias*=head/left/even/offset,%
1731   even/left/head/offset/.alias*=head/left/even/offset,%
1732   offset/head/left/even/.alias*=head/left/even/offset,%
1733   offset/head/even/left/.alias*=head/left/even/offset,%
1734   offset/left/head/even/.alias*=head/left/even/offset,%
1735   offset/left/even/head/.alias*=head/left/even/offset,%
1736   offset/even/left/head/.alias*=head/left/even/offset,%
1737   offset/even/head/left/.alias*=head/left/even/offset,%
1738 head/left/odd/offset/.store in skip cs=f@ncy0@olh,%
1739   head/left/odd/offset/.append code=\fancy@setoffs,%
1740   head/odd/offset/left/.alias*=head/left/odd/offset,%
1741   head/odd/left/offset/.alias*=head/left/odd/offset,%
1742   head/offset/odd/left/.alias*=head/left/odd/offset,%
1743   head/offset/left/odd/.alias*=head/left/odd/offset,%
1744   head/left/offset/odd/.alias*=head/left/odd/offset,%
1745   left/head/offset/odd/.alias*=head/left/odd/offset,%
1746   left/head/odd/offset/.alias*=head/left/odd/offset,%
1747   left/offset/head/odd/.alias*=head/left/odd/offset,%
1748   left/offset/odd/head/.alias*=head/left/odd/offset,%
1749   left/odd/offset/head/.alias*=head/left/odd/offset,%
1750   left/odd/head/offset/.alias*=head/left/odd/offset,%
1751   odd/head/offset/left/.alias*=head/left/odd/offset,%
1752   odd/head/left/offset/.alias*=head/left/odd/offset,%
1753   odd/offset/head/left/.alias*=head/left/odd/offset,%
1754   odd/offset/left/head/.alias*=head/left/odd/offset,%
1755   odd/left/offset/head/.alias*=head/left/odd/offset,%
1756   odd/left/head/offset/.alias*=head/left/odd/offset,%
1757   offset/head/left/odd/.alias*=head/left/odd/offset,%
1758   offset/left/head/odd/.alias*=head/left/odd/offset,%
1759   offset/left/odd/head/.alias*=head/left/odd/offset,%
1760   offset/odd/left/head/.alias*=head/left/odd/offset,%
1761   offset/odd/head/left/.alias*=head/left/odd/offset,%
1762 head/right/even/offset/.store in skip cs=f@ncy0@erh,%
1763   head/right/even/offset/.append code=\fancy@setoffs,%
1764   head/even/offset/right/.alias*=head/right/even/offset,%
1765   head/even/right/offset/.alias*=head/right/even/offset,%
1766   head/offset/even/right/.alias*=head/right/even/offset,%
1767   head/offset/right/even/.alias*=head/right/even/offset,%
1768   head/right/offset/even/.alias*=head/right/even/offset,%
1769   right/head/offset/even/.alias*=head/right/even/offset,%
1770   right/head/even/offset/.alias*=head/right/even/offset,%
1771   right/offset/head/even/.alias*=head/right/even/offset,%
1772   right/offset/even/head/.alias*=head/right/even/offset,%
1773   right/even/offset/head/.alias*=head/right/even/offset,%
1774   right/even/head/offset/.alias*=head/right/even/offset,%
1775   even/head/offset/right/.alias*=head/right/even/offset,%
1776   even/head/right/offset/.alias*=head/right/even/offset,%
1777   even/offset/head/right/.alias*=head/right/even/offset,%
1778   even/offset/right/head/.alias*=head/right/even/offset,%
1779   even/right/offset/head/.alias*=head/right/even/offset,%
1780   even/right/head/offset/.alias*=head/right/even/offset,%
1781   offset/head/right/even/.alias*=head/right/even/offset,%
1782   offset/head/even/right/.alias*=head/right/even/offset,%
1783   offset/right/head/even/.alias*=head/right/even/offset,%
1784   offset/right/even/head/.alias*=head/right/even/offset,%
1785   offset/even/right/head/.alias*=head/right/even/offset,%
1786   offset/even/head/right/.alias*=head/right/even/offset,%
1787 head/right/odd/offset/.store in skip cs=f@ncy0@orh,%
1788   head/right/odd/offset/.append code=\fancy@setoffs,%
1789

```

```
1790     head/odd/offset/right/.alias*=head/right/odd/offset,%
1791     head/odd/right/offset/.alias*=head/right/odd/offset,%
1792     head/offset/odd/right/.alias*=head/right/odd/offset,%
1793     head/offset/right/odd/.alias*=head/right/odd/offset,%
1794     head/right/offset/odd/.alias*=head/right/odd/offset,%
1795     right/head/offset/odd/.alias*=head/right/odd/offset,%
1796     right/head/odd/offset/.alias*=head/right/odd/offset,%
1797     right/offset/head/odd/.alias*=head/right/odd/offset,%
1798     right/offset/odd/head/.alias*=head/right/odd/offset,%
1799     right/odd/offset/head/.alias*=head/right/odd/offset,%
1800     right/odd/head/offset/.alias*=head/right/odd/offset,%
1801     odd/head/offset/right/.alias*=head/right/odd/offset,%
1802     odd/head/right/offset/.alias*=head/right/odd/offset,%
1803     odd/offset/head/right/.alias*=head/right/odd/offset,%
1804     odd/offset/right/head/.alias*=head/right/odd/offset,%
1805     odd/right/offset/head/.alias*=head/right/odd/offset,%
1806     odd/right/head/offset/.alias*=head/right/odd/offset,%
1807     offset/head/right/odd/.alias*=head/right/odd/offset,%
1808     offset/head/odd/right/.alias*=head/right/odd/offset,%
1809     offset/right/head/odd/.alias*=head/right/odd/offset,%
1810     offset/right/odd/head/.alias*=head/right/odd/offset,%
1811     offset/odd/right/head/.alias*=head/right/odd/offset,%
1812     offset/odd/head/right/.alias*=head/right/odd/offset,%
1813 foot/left/even/offset/.store in skip cs=f@ncy0@elf,%
1814         foot/left/even/offset/.append code=\fancy@setoffs,%
1815         foot/even/offset/left/.alias*=foot/left/even/offset,%
1816         foot/even/left/offset/.alias*=foot/left/even/offset,%
1817         foot/offset/even/left/.alias*=foot/left/even/offset,%
1818         foot/offset/left/even/.alias*=foot/left/even/offset,%
1819         foot/left/offset/even/.alias*=foot/left/even/offset,%
1820         left/foot/offset/even/.alias*=foot/left/even/offset,%
1821         left/foot/even/offset/.alias*=foot/left/even/offset,%
1822         left/offset/foot/even/.alias*=foot/left/even/offset,%
1823         left/offset/even/foot/.alias*=foot/left/even/offset,%
1824         left/even/offset/foot/.alias*=foot/left/even/offset,%
1825         left/even/foot/offset/.alias*=foot/left/even/offset,%
1826         even/foot/offset/left/.alias*=foot/left/even/offset,%
1827         even/foot/left/offset/.alias*=foot/left/even/offset,%
1828         even/offset/foot/left/.alias*=foot/left/even/offset,%
1829         even/offset/left/foot/.alias*=foot/left/even/offset,%
1830         even/left/offset/foot/.alias*=foot/left/even/offset,%
1831         even/left/foot/offset/.alias*=foot/left/even/offset,%
1832         offset/foot/left/even/.alias*=foot/left/even/offset,%
1833         offset/foot/even/left/.alias*=foot/left/even/offset,%
1834         offset/left/foot/even/.alias*=foot/left/even/offset,%
1835         offset/left/even/foot/.alias*=foot/left/even/offset,%
1836         offset/even/left/foot/.alias*=foot/left/even/offset,%
1837         offset/even/foot/left/.alias*=foot/left/even/offset,%
1838 foot/left/odd/offset/.store in skip cs=f@ncy0@olf,%
1839         foot/left/odd/offset/.append code=\fancy@setoffs,%
1840         foot/odd/offset/left/.alias*=foot/left/odd/offset,%
1841         foot/odd/left/offset/.alias*=foot/left/odd/offset,%
1842         foot/offset/odd/left/.alias*=foot/left/odd/offset,%
1843         foot/offset/left/odd/.alias*=foot/left/odd/offset,%
1844         foot/left/offset/odd/.alias*=foot/left/odd/offset,%
1845         left/foot/offset/odd/.alias*=foot/left/odd/offset,%
1846         left/foot/odd/offset/.alias*=foot/left/odd/offset,%
1847         left/offset/foot/odd/.alias*=foot/left/odd/offset,%
1848         left/offset/odd/foot/.alias*=foot/left/odd/offset,%
1849         left/odd/offset/foot/.alias*=foot/left/odd/offset,%
1850         left/odd/foot/offset/.alias*=foot/left/odd/offset,%
1851         odd/foot/offset/left/.alias*=foot/left/odd/offset,%
1852         odd/foot/left/offset/.alias*=foot/left/odd/offset,%
1853         odd/offset/foot/left/.alias*=foot/left/odd/offset,%
1854         odd/offset/left/foot/.alias*=foot/left/odd/offset,%
```

```

1855     odd/left/offset/.alias*=foot/left/odd/offset,%
1856     odd/left/foot/offset/.alias*=foot/left/odd/offset,%
1857     offset/foot/left/odd/.alias*=foot/left/odd/offset,%
1858     offset/foot/odd/left/.alias*=foot/left/odd/offset,%
1859     offset/left/foot/odd/.alias*=foot/left/odd/offset,%
1860     offset/left/odd/foot/.alias*=foot/left/odd/offset,%
1861     offset/odd/left/foot/.alias*=foot/left/odd/offset,%
1862     offset/odd/foot/left/.alias*=foot/left/odd/offset,%
1863 foot/right/even/offset/.store in skip cs=f@ncy0@erf,%
1864     foot/right/even/offset/.append code=\fancy@setoffs,%
1865     foot/even/offset/right/.alias*=foot/right/even/offset,%
1866     foot/even/right/offset/.alias*=foot/right/even/offset,%
1867     foot/offset/even/right/.alias*=foot/right/even/offset,%
1868     foot/offset/right/even/.alias*=foot/right/even/offset,%
1869     foot/right/offset/even/.alias*=foot/right/even/offset,%
1870     right/foot/offset/even/.alias*=foot/right/even/offset,%
1871     right/foot/even/offset/.alias*=foot/right/even/offset,%
1872     right/offset/foot/even/.alias*=foot/right/even/offset,%
1873     right/offset/even/foot/.alias*=foot/right/even/offset,%
1874     right/even/offset/foot/.alias*=foot/right/even/offset,%
1875     right/even/foot/offset/.alias*=foot/right/even/offset,%
1876     even/foot/offset/right/.alias*=foot/right/even/offset,%
1877     even/foot/right/offset/.alias*=foot/right/even/offset,%
1878     even/offset/foot/right/.alias*=foot/right/even/offset,%
1879     even/offset/right/foot/.alias*=foot/right/even/offset,%
1880     even/right/offset/foot/.alias*=foot/right/even/offset,%
1881     even/right/foot/offset/.alias*=foot/right/even/offset,%
1882     offset/foot/right/even/.alias*=foot/right/even/offset,%
1883     offset/foot/even/right/.alias*=foot/right/even/offset,%
1884     offset/right/foot/even/.alias*=foot/right/even/offset,%
1885     offset/right/even/foot/.alias*=foot/right/even/offset,%
1886     offset/even/right/foot/.alias*=foot/right/even/offset,%
1887     offset/even/foot/right/.alias*=foot/right/even/offset,%
1888 foot/right/odd/offset/.store in skip cs=f@ncy0@orf,%
1889     foot/right/odd/offset/.append code=\fancy@setoffs,%
1890     foot/odd/offset/right/.alias*=foot/right/odd/offset,%
1891     foot/odd/right/offset/.alias*=foot/right/odd/offset,%
1892     foot/offset/odd/right/.alias*=foot/right/odd/offset,%
1893     foot/offset/right/odd/.alias*=foot/right/odd/offset,%
1894     foot/right/offset/odd/.alias*=foot/right/odd/offset,%
1895     right/foot/offset/odd/.alias*=foot/right/odd/offset,%
1896     right/foot/odd/offset/.alias*=foot/right/odd/offset,%
1897     right/offset/foot/odd/.alias*=foot/right/odd/offset,%
1898     right/offset/odd/foot/.alias*=foot/right/odd/offset,%
1899     right/odd/offset/foot/.alias*=foot/right/odd/offset,%
1900     right/odd/foot/offset/.alias*=foot/right/odd/offset,%
1901     odd/foot/offset/right/.alias*=foot/right/odd/offset,%
1902     odd/foot/right/offset/.alias*=foot/right/odd/offset,%
1903     odd/offset/foot/right/.alias*=foot/right/odd/offset,%
1904     odd/offset/right/foot/.alias*=foot/right/odd/offset,%
1905     odd/right/offset/foot/.alias*=foot/right/odd/offset,%
1906     odd/right/foot/offset/.alias*=foot/right/odd/offset,%
1907     offset/foot/right/odd/.alias*=foot/right/odd/offset,%
1908     offset/foot/odd/right/.alias*=foot/right/odd/offset,%
1909     offset/right/foot/odd/.alias*=foot/right/odd/offset,%
1910     offset/right/odd/foot/.alias*=foot/right/odd/offset,%
1911     offset/odd/right/foot/.alias*=foot/right/odd/offset,%
1912     offset/odd/foot/right/.alias*=foot/right/odd/offset,%
1913 head/inner/offset/.style={head/right/even/offset={#1},head/left/odd/offset={#1}},%
1914 head/inner/offset+/.style={head/right/even/offset+={#1},head/left/odd/offset+={#1}},%
1915 head/inner/offset-/.style={head/right/even/offset-={#1},head/left/odd/offset-={#1}},%
1916 head/inner/offset*/.style={head/right/even/offset*={#1},head/left/odd/offset*={#1}},%
1917     head/offset/inner/.alias*=head/inner/offset,%
1918     offset/head/inner/.alias*=head/inner/offset,%
1919     offset/inner/head/.alias*=head/inner/offset,%

```

```

1920     inner/head/offset/.alias*={head/inner/offset},%
1921     inner/offset/head/.alias*={head/inner/offset},%
1922 head/outer/offset/.style={head/left/even/offset={#1},head/right/odd/offset={#1}},%
1923 head/outer/offset+/.style={head/left/even/offset+={#1},head/right/odd/offset+={#1}},%
1924 head/outer/offset-/.style={head/left/even/offset-={#1},head/right/odd/offset-={#1}},%
1925 head/outer/offset*/.style={head/left/even/offset*={#1},head/right/odd/offset*={#1}},%
1926     head/offset/outer/.alias*={head/outer/offset},%
1927     offset/head/outer/.alias*={head/outer/offset},%
1928     offset/outer/head/.alias*={head/outer/offset},%
1929     outer/head/offset/.alias*={head/outer/offset},%
1930     outer/offset/head/.alias*={head/outer/offset},%
1931 foot/inner/offset/.style={foot/right/even/offset={#1},foot/left/odd/offset={#1}},%
1932 foot/inner/offset+/.style={foot/right/even/offset+={#1},foot/left/odd/offset+={#1}},%
1933 foot/inner/offset-/.style={foot/right/even/offset-={#1},foot/left/odd/offset-={#1}},%
1934 foot/inner/offset*/.style={foot/right/even/offset*={#1},foot/left/odd/offset*={#1}},%
1935     foot/offset/inner/.alias*={foot/inner/offset},%
1936     offset/foot/inner/.alias*={foot/inner/offset},%
1937     offset/inner/foot/.alias*={foot/inner/offset},%
1938     inner/foot/offset/.alias*={foot/inner/offset},%
1939     inner/offset/foot/.alias*={foot/inner/offset},%
1940 foot/outer/offset/.style={foot/left/even/offset={#1},foot/right/odd/offset={#1}},%
1941 foot/outer/offset+/.style={foot/left/even/offset+={#1},foot/right/odd/offset+={#1}},%
1942 foot/outer/offset-/.style={foot/left/even/offset-={#1},foot/right/odd/offset-={#1}},%
1943 foot/outer/offset*/.style={foot/left/even/offset*={#1},foot/right/odd/offset*={#1}},%
1944     foot/offset/outer/.alias*={foot/outer/offset},%
1945     offset/foot/outer/.alias*={foot/outer/offset},%
1946     offset/outer/foot/.alias*={foot/outer/offset},%
1947     outer/foot/offset/.alias*={foot/outer/offset},%
1948     outer/offset/foot/.alias*={foot/outer/offset},%
1949 inner/offset/.style={foot/inner/offset={#1},head/inner/offset={#1}},%
1950 inner/offset+/.style={foot/inner/offset+={#1},head/inner/offset+={#1}},%
1951 inner/offset-/.style={foot/inner/offset-={#1},head/inner/offset-={#1}},%
1952 inner/offset*/.style={foot/inner/offset*={#1},head/inner/offset*={#1}},%
1953     offset/inner/.alias*={inner/offset},%
1954 outer/offset/.style={foot/outer/offset={#1},head/outer/offset={#1}},%
1955 outer/offset+/.style={foot/outer/offset+={#1},head/outer/offset+={#1}},%
1956 outer/offset-/.style={foot/outer/offset-={#1},head/outer/offset-={#1}},%
1957 outer/offset*/.style={foot/outer/offset*={#1},head/outer/offset*={#1}},%
1958     offset/outer/.alias*={outer/offset},%
1959 reset/.style={norule,head=,foot=,offset=\z@,foot/rule/skip,rule/font=,font=},%
1960     reset/.value forbidden,%
1961 }% keys for \pagesetup (stored into token \itf@tk)
1962 \AtBeginDocument{%
1963 \ifdim\headwidth<\z@
1964 %% This catches the case that \headwidth hasn't been initialized and the
1965 %% case that the user added something to \headwidth in the expectation that
1966 %% it was initialized to \textwidth. We compensate this now. This loses if
1967 %% the user intended to multiply it by a factor. But that case is more
1968 %% likely done by saying something like \headwidth=1.2\textwidth.
1969 %% The doc says you have to change \headwidth after the first call to
1970 %% \pagestyle{fancy}. This code is just to catch the most common cases were
1971 %% that requirement is violated.
1972     \global\advance\headwidth123456789sp\global\advance\headwidth\textwidth
1973 \fi}
1974 fancyhdr>

```

21.10 marks

```

1975 (*marks)
1976 \ProvidesPackage{interfaces-marks}
1977     [2010/12/07 v2.5 easier use of e-TeX marks (FC)]
1978 \RequirePackage{interfaces-base}

```

\marksthe

```
1979 \newrobustcmd{\marksthe}[2]{%
```

```

1980  \ifcsname marks@#1\endcsname \else \itf@marks@new{#1}\fi
1981  {\toks@{#2}\expandafter\marks\csname marks@#1\endcsname{\the\toks@}}%
1982 }% \marksthecs

\marksthecs

1983 \newrobustcmd*\marksthecs[2]{%
1984  \ifcsname marks@#1\endcsname \else \itf@marks@new{#1}\fi
1985  \ifcsundef{#2}
1986  {}
1987  {{\toks@\expandafter\expandafter\expandafter{%
1988   \csname #2\endcsname}%
1989   \expandafter\marks\csname marks@#1\endcsname{\the\toks@}}}}%
1990 }% \marksthecs

\itf@marks@new

1991 \def\itf@marks@new#1{%
1992  \expandafter\newmarks\csname marks@#1\endcsname
1993  \PackageInfo{interfaces-marks}{marks '#1' defined}%
1994 }% \itf@marks@new

\thefirstmarks Purely expandable in exactly one step of expansion.

\thebotmarks
\thetopmarks
1995 \newcommand*\thefirstmarks[1]{\romannumeral
1996  \ifcsname marks@#1\endcsname \expandafter\z@
1997   \firstmarks\csname marks@#1\expandafter\endcsname
1998  \else \expandafter\z@
1999  \fi
2000 }% \thefirstmarks
2001 \newcommand*\thebotmarks[1]{\romannumeral
2002  \ifcsname marks@#1\endcsname \expandafter\z@
2003   \botmarks\csname marks@#1\expandafter\endcsname
2004  \else \expandafter\z@
2005  \fi
2006 }% \thebotmarks
2007 \newcommand*\thetopmarks[1]{\romannumeral
2008  \ifcsname marks@#1\endcsname \expandafter\z@
2009   \topmarks\csname marks@#1\expandafter\endcsname
2010  \else \expandafter\z@
2011  \fi
2012 }% \thetopmarks

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

2013 %% \getthemarks\firstmarks{section}\macro
2014 \newrobustcmd*\getthemarks[3]{%
2015  \ifcsname marks@#2\endcsname
2016   \edef#3{\unexpanded\expandafter{#1\csname marks@#2\endcsname}}%
2017  \else \let#3\undefined
2018  \fi
2019 }% \getthemarks

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

\getthebotmarks
\getthetopmarks
2020 \newrobustcmd*\getthefirstmarks{\getthemarks\firstmarks}
2021 \newrobustcmd*\getthebotmarks{\getthemarks\botmarks}
2022 \newrobustcmd*\getthetopmarks{\getthemarks\topmarks}

\ifmarksequal

2023 %% \ifmarksequal\firstmarks\botmarks{subsection}{ true }{ false }
2024 \newrobustcmd*\ifmarksequal[3]{%
2025  \begin{group}
2026    \getthemarks{#1}{#3}\x
2027    \getthemarks{#2}{#3}\y
2028  \expandafter\endgroup

```

```

2029   \csname ltx@\ifdefined\x\ifdefined\y
2030     \ifx\x\y first\else
2031       second\fi\else second\fi\else second\fi
2032   oftwo\endcsname
2033 }% \ifmarksequal
2034 
```

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

```

2035 {*tocloft}
2036 \ProvidesPackage{interfaces-tocloft}
2037 [2010/12/07 v2.5 pgfkeys macros for tocloft (FC)]
2038 \RequirePackage{interfaces-base}
2039 \itf@Check@Package{tocloft}
2040 \itf@pgfkeys

```

\tocsetup The user commands.

```

\lofsetup 2041 \newrobustcmd\tocsetup[1]{%
\lotsetup 2042   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/toc}{#1}}%
\listofsetup 2043 }% \tocsetup
2044 \newrobustcmd\lofsetup[1]{%
2045   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/lof}{#1}}%
2046 }% \lofsetup
2047 \newrobustcmd\lotsetup[1]{%
2048   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/lot}{#1}}%
2049 }% \lotsetup
2050 \newrobustcmd\listofsetup[2]{%
2051   \AfterPreamble{\itf@pgfkeys\pgfqkeys{/itf/tocloft/#1}{#2}}%
2052 }% \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`.

```

2053 \def\itf@cft@newlistof#1#2#3{%
  #1=toc/lof (ext.), #2=entry (\newlistof only), #3=title (\newl
2054   \pgfqkeys{/itf/tocloft/#1}{%
2055     depth/.store in counter cs=c@#1depth,%
2056     depth/.append code=\expandafter\global\csname c@#1depth\endcsname
2057                                         \csname c@#1depth\endcsname
2058     \addtocontents{#1}{\csname c@#1depth\endcsname
2059                               \the\csname c@#1depth\endcsname\relax},%
2060     #1depth/.alias*=depth,%
2061     pagestyle/.code=\tocloftpagestyle{##1},%
2062     parskip/.store in skip cs=cftparskip,%
2063     parsep/.alias=parskip,%
2064     mark/.store in cs=cftmark#1,%
2065     marks/.alias=mark,%
2066     mark=,%
2067     label/.store as code,%
2068     ref/.store as code,%
2069     ref/.initial=\csname cft#1title\endcsname,% overwritten (toc,lof,lot)
2070     ref/.default=\csname cft#1title\endcsname,% overwritten (toc,lof,lot)
2071     ref/.blank=\pgfkeysnovalue,%
2072     title/.store as code in cs=cft#1title,% overwritten below for toc, lof and lot
2073     title/top/.store in skip cs=cftbefore#1titleskip,%
2074     beforetitleskip/.alias*=title/top,%
2075     title/bottom/.store in skip cs=cftafter#1titleskip,%
2076     aftertitleskip/.alias*=title/bottom,%
2077     title/font/.store as code in cs=cft#1titlefont,%
2078     title/format/.alias*=title/font,%
2079     title/color/.store as color,%
2080     title/color/.append code=\pgfkeysalso{[/itf/tocloft/#1/title/font+={%
2081       \color{\pgfkeysvalueof{/itf/tocloft/#1/title/color}}}},%
2082     title/font/color/.alias=title/color,%
2083     title/format/color/.alias=title/color,%

```

```

2084     title/after/.store as code in cs=cftafter#1title,%
2085     title/command/.code=\csdef{cftmake#1title}####1{##1},%
2086     dotsep/.code=\itf@units@\itf@musk{##1}%      converts to mu
2087             \edef\cftdotsep{\itf@strip@\itf@musk},%
2088             dotsep/.default=4.5mu,%
2089             dotsep/.blank=\cftnodots,%
2090     dot/.store in=\cftdot,%
2091     before/.store as code,%
2092     after/.store as code,%
2093     columns/before@hook/.is private=interfaces-tocloft,%
2094     columns/after@hook/.is private=interfaces-tocloft,%
2095     columns/before@hook/.initial=,%
2096     columns/after@hook/.initial=,%
2097     columns/.code={\ltx@ifpackageloaded{multicol}{%
2098         \pgfkeysEsetvalue\pgfkeyscurrentkey{\number##1}%
2099         \ifnum\number##1>\@ne
2100             \pgfkeysgetvalue{/itf/tocloft/#1/columns/before@hook}\itf@tp
2101             \ifx\itf@tp\@empty
2102                 \pgfkeyssetvalue{/itf/tocloft/#1/columns/before@hook}{%
2103                     \begingroup
2104                     \multicols\pgfkeysvalueof{/itf/tocloft/#1/multicolsep}\relax
2105                     \begin{multicols}{\pgfkeysvalueof{/itf/tocloft/#1/columns}}%
2106                     \columnseprule\pgfkeysvalueof{/itf/tocloft/#1/columns/rule}\relax
2107                     \columnsep\pgfkeysvalueof{/itf/tocloft/#1/columns/sep}\relax
2108                     \pgfkeysgetvalue{/itf/tocloft/#1/columns/rule/color}\columnseprulecolor
2109                     \pgfkeysvalueof{/itf/tocloft/#1/multicols/begin}%
2110                 }%
2111                 \pgfkeyssetvalue{/itf/tocloft/#1/columns/after@hook}{%
2112                     \pgfkeysvalueof{/itf/tocloft/#1/multicols/end}%
2113                     \end{multicols}
2114                     \endgroup
2115                 }%
2116             \fi
2117         \else
2118             \pgfkeyslet\pgfkeyscurrentkey\@ne
2119             \pgfkeyslet{/itf/tocloft/#1/columns/before@hook}\@empty
2120             \pgfkeyslet{/itf/tocloft/#1/columns/after@hook}\@empty
2121         \fi}
2122         {\PackageError{interfaces-tocloft}
2123             {Key ‘columns’: package multicol is required}\@eha}%
2124     },%
2125     columns/.default=2,columns/.blank=1,%
2126     multicol/.alias=columns,%
2127     multicols/.alias=columns,%
2128     twocolumns/.true=\pgfkeysalso{columns=2},%
2129     twocolumns/.false=\pgfkeysalso{columns=1},%
2130     columns/rule/.estore as dimen,%
2131     columns/rule/.default=\columnseprule,%
2132     columns/rule,% assign default
2133         columnseprule/.alias*=columns/rule,%
2134         columns/rule/width/.alias*=columns/rule,%
2135     columns/sep/.estore as dimen,%
2136     columns/sep/.default=\columnsep,%
2137     columns/sep/.blank=10\p@,%
2138     columns/sep,% assign default
2139         columnsep/.alias*=columns/sep,%
2140     columns/rule/color/.store as color command,%
2141     columns/rule/color,% assign default (\normalcolor)
2142         columnseprulecolor/.alias*=columns/rule/color,%
2143         columns/rulecolor/.alias*=columns/rule/color,%
2144     multicolssep/.estore as glue,%
2145         multicolssep/.default=0pt plus4pt minus3pt,%
2146         multicolssep,% sets default
2147     multicols/begin/.initial=,multicols/begin/.default=,%
2148     multicols/before/.alias=multicols/begin,%

```

```

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

```

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

```

2203 \itf@cft@newlistof{toc}{}{}
2204 \itf@cft@newlistof{lof}{}{}
2205 \itf@cft@newlistof{lot}{}{}

2206 %% Specific to toc only -----
2207 \pgfqkeys{/itf/tocloft/toc}{}%
2208     title/.store as code in cs=contentsname,%
2209     contentsname/.alias=title,%

```

```

2210   ref/.initial=\contentsname,%
2211     ref/.default=\contentsname,%
2212   pagenumbers/.style={% boolean key
2213     part/pagenumbers={#1},chapter/pagenumbers={#1},section/pagenumbers={#1},%
2214     subsubsection/pagenumbers={#1},subsubsubsection/pagenumbers={#1},%
2215     paragraph/pagenumbers={#1},subparagraph/pagenumbers={#1},%
2216     page/pagenumbers/.alias=pagenumbers,%
2217 }%
2218 %% Specific to lof and lot -----
2219 \pgfqkeys{/itf/tocloft/lof}{%
2220   title/.store as code in cs=listfigurename,%
2221     listfigurename/.alias=title,%
2222   ref/.initial=\listfigurename,%
2223     ref/.default=\listfigurename,%
2224 }%
2225 \pgfqkeys{/itf/tocloft/lot}{%
2226   title/.store as code in cs=listtablename,%
2227     listtablename/.alias=title,%
2228   ref/.initial=\listtablename,%
2229     ref/.default=\listtablename,%
2230 }%
2231 \if@cftsubfigopt
2232 \pgfqkeys{/itf/tocloft}{%
2233   lof/pagenumbers/.style={figure/pagenumbers={#1},subfigure/pagenumbers={#1}},%
2234     lof/pagenumber/.alias=lof/pagenumbers,%
2235   lot/pagenumbers/.style={figure/pagenumbers={#1},subfigure/pagenumbers={#1}},%
2236     lot/pagenumber/.alias=lot/pagenumbers,%
2237 }%
2238 \else
2239 \pgfqkeys{/itf/tocloft}{%
2240   lof/pagenumbers/.style={figure/pagenumbers={#1}},%
2241     lof/pagenumber/.alias=lof/pagenumbers,%
2242   lot/pagenumbers/.style={figure/pagenumbers={#1}},%
2243     lot/pagenumber/.alias=lot/pagenumbers,%
2244 }%
2245 \fi

```

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

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

```

2246 \def\itf@cft@name#1.#2.#3@nil{cft#1\itf@cft@secname{#2}#3}
2247 \def\itf@cft@secname#1{%
2248   \expandafter\itf@cft@sec@name\csname#1\endcsname{#1}%
2249 }% \itf@cft@secname
2250 \def\itf@cft@sec@name#1#2{%
2251   \ifx#1\relax #2\else
2252   \ifx#1\section sec\else
2253     \ifx#1\subsection subsec\else
2254       \ifx#1\subsubsection subsubsec\else
2255         \ifx#1\paragraph para\else
2256           \ifx#1\subparagraph subpara\else
2257             \ifx#1\figure fig\else
2258               \ifx#1\subsubfigure subfig\else
2259                 \ifx#1\table tab\else
2260                   \ifx#1\subtable subtab\else
2261                     \ifx#1\chapter chap\else
2262                       #2\fi
2263             \fi\fi\fi\fi\fi\fi\fi
2264 }% \itf@cft@section@name

2265 %% tocloft setup for tables sections :
2266 \pgfkeys{%
2267 /handlers/.cft@def/.style={%
2268   \pgfkeyscurrentpath/.store in cs=\itf@cft@name #1@nil},%

```

```

2269 /handlers/.cft@code/.style=%
2270   \pgfkeyscurrentpath/.store as code in cs=\itf@cft@name #1@nil},%
2271 /handlers/.cft@length/.style=%
2272   \pgfkeyscurrentpath/.store in skip cs=\itf@cft@name #1@nil},%
2273 }
```

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

```

2274 \def\itf@cft@newsection{\itf@cft@new@section{/itf/tocloft}}
2275 \def\itf@cft@new@section#1#2#3{%
2276   \pgfqkeys{#1/#2/#3}{%
2277     skip/.cft@length=before.#3.skip,%
2278       beforeskip/.alias*=skip,%
2279       top/.alias*=skip,%
2280     indent/.cft@length=.#3.indent,%
2281     font/.cft@code=.#3.font,%
2282     font/.append code=\pgfkeysalso{/itf/tocloft/#2/#3/pagenumber/font={##1}},%
2283       format/.alias*=font,%
2284     color/.store as color,%
2285       color/.append code=\pgfkeysalso{/itf/tocloft/#2/#3/font+=%
2286         \color{\pgfkeysvalueof{/itf/tocloft/#2/#3/color}}%
2287         \ifdefined\hypersetup \hypersetup{%
2288           linkcolor=\pgfkeysvalueof{/itf/tocloft/#2/#3/color}}%
2289         \fi},%
2290       font/color/.alias=color,%
2291       format/color/.alias=color,%
2292     number/width/.cft@length=.#3.numwidth,%
2293       numwidth/.alias*=number/width,%
2294     presnum/.cft@code=.#3.presnum,%
2295       number/before/.alias*=presnum,%
2296     aftersnum/.cft@code=.#3.aftersnum,%
2297       number/after/.alias*=aftersnum,%
2298     aftersnumb/.cft@code=.#3.aftersnumb,%
2299       title/before/.alias*=aftersnumb,%
2300     leader/.cft@def=.#3.leader,%
2301       leader/.default=\cftdotfill{\csname\itf@cft@name.#3.dotsep@nil\endcsname},%
2302       leaders/.alias=leader,%
2303     dotsep/.code={\ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
2304       \csdef{\itf@cft@name.#3.dotsep@nil}{\cftdotsep}%
2305       \else
2306         \itf@units@mu\itf@musk{##1}%
2307         \csedef{\itf@cft@name.#3.dotsep@nil}{\itf@strip@mu\itf@musk}%
2308       \fi},%
2309     dotsep/.blank=\cftnodots,%
2310     nodots/.code=\csdef{\itf@cft@name.#3.dotsep@nil}{\cftnodots},%
2311       nodots/.value forbidden,%
2312     pagefont/.cft@code=.#3.pagefont,%
2313       page/font/.alias*=pagefont,%
2314       pagename/font/.alias*=pagefont,%
2315       pagenumbers/font/.alias*=pagefont,%
2316     page/color/.store as color,%
2317       page/color/.append code=\pgfkeysalso{/itf/tocloft/#2/#3/pagefont+=%
2318         \color{\pgfkeysvalueof{/itf/tocloft/#2/#3/page/color}}%
2319         \ifdefined\hypersetup \hypersetup{%
2320           linkcolor=\pgfkeysvalueof{/itf/tocloft/#2/#3/page/color}}%
2321         \fi},%
2322       pagename/color/.alias=page/color,%
2323       pagenumbers/color/.alias=page/color,%
2324       page/font/color/.alias=page/color,%
2325       pagename/font/color/.alias=page/color,%
2326       pagenumbers/font/color/.alias=page/color,%
2327     afterpnum/.cft@code=.#3.afterpnum,%
2328       page/after/.alias*=afterpnum,%
2329       pagename/after/.alias*=afterpnum,%
2330     pagenumbers/.on=@cftpnumon{\itf@cft@secname{#3}},%
```

```

2331      pagenumbers/.off= \@cftpnumoff{\itf@cft@secname{#3}},%
2332          pagenumber/.alias=pagenumbers,%
2333      }% \pgfqkeys
2334 }% \itf@cft@new@section

2335 \@for\itf@tp:=part, chap, chapter, sec, section, subsec, subsection,%
2336             subsubsec, subsubsection, para, paragraph, subpara, subparagraph%
2337   \do{\edef\itf@tp{\noexpand\itf@cft@newsection{toc}{\itf@tp}}\itf@tp}%
2338 \if@cftsubfigopt
2339   \@for\itf@tp:=fig, figure, subfig, subfigure%
2340     \do{\edef\itf@tp{\noexpand\itf@cft@newsection{lof}{\itf@tp}}\itf@tp}%
2341   \@for\itf@tp:=tab, table, subtab, subtable%
2342     \do{\edef\itf@tp{\noexpand\itf@cft@newsection{lot}{\itf@tp}}\itf@tp}%
2343 \else
2344   \@for\itf@tp:=fig, figure%
2345     \do{\edef\itf@tp{\noexpand\itf@cft@newsection{lof}{\itf@tp}}\itf@tp}%
2346   \@for\itf@tp:=tab, table%
2347     \do{\edef\itf@tp{\noexpand\itf@cft@newsection{lot}{\itf@tp}}\itf@tp}%
2348 \fi

\@cftmaketoctitle (patch)
\@cftmakeloftitle (patch)
2349 \itf@patch\pretocmd\@cftmaketoctitle
\@cftmakelottitle (patch) {\itf@tocloft\contentsname{toc}{tableofcontents}}
\\newlistof (patch) {\@PackageInfoNoLine{interfaces-tocloft}
2351   {Command \string@\cftmaketoctitle\space successfully augmented}}
2352   {\@PackageWarningNoLine{interfaces-tocloft}
2353     {Command \string@\cftmaketoctitle\space could not be patched
2354      \MessageBreak label and bookmark keys for \string\tocsetup\space won't work}}%
2355 \itf@patch\pretocmd\@cftmakeloftitle
2356 {\itf@tocloft\listfigurename{lof}{listoffigures}}
2357 {\@PackageInfoNoLine{interfaces-tocloft}
2358   {Command \string@\cftmakeloftitle\space successfully augmented}}
2359   {\@PackageWarningNoLine{interfaces-tocloft}
2360     {Command \string@\cftmakeloftitle\space could not be patched
2361      \MessageBreak label and bookmark keys for \string\lofsetup\space won't work}}%
2362 \itf@patch\pretocmd\@cftmakelottitle
2363 {\itf@tocloft\listtablename{lot}{listoftables}}
2364 {\@PackageInfoNoLine{interfaces-tocloft}
2365   {Command \string@\cftmakelottitle\space successfully augmented}}
2366   {\@PackageWarningNoLine{interfaces-tocloft}
2367     {Command \string@\cftmakelottitle\space could not be patched
2368      \MessageBreak label and bookmark keys for \string\lotsetup\space won't work}}%
2369 \expandafter\itf@patch\expandafter\patchcmd\csname\string\newlistof\endcsname
2370 {\#4}\{\csname cft#3title\endcsname}
2371 {\@PackageInfoNoLine{interfaces-tocloft}
2372   {Command \string\newlistof\space successfully patched}}
2373   {\@PackageWarningNoLine{interfaces-tocloft}
2374     {Command \string\newlistof\space could not be patched
2375      \MessageBreak \string\listofsetup\space will not work}}%
2376 \expandafter\itf@patch\expandafter\apptocmd\csname\string\newlistof\endcsname
2377 {\itf@cft@newlistof{\#3}{\#2}{\#4}}
2378 {\@PackageInfoNoLine{interfaces-tocloft}
2379   {Command \string\newlistof\space successfully augmented}}
2380   {\@PackageWarningNoLine{interfaces-tocloft}
2381     {Command \string\newlistof\space could not be patched
2382      \MessageBreak \string\listofsetup\space will not work}}%
2383 \expandafter\itf@patch\expandafter\apptocmd\csname\string\newlistentry\endcsname
2384 {\itf@cft@newsection{\#3}{\#2}}
2385 {\@PackageInfoNoLine{interfaces-tocloft}
2386   {Command \string\newlistentry\space successfully augmented}}
2387   {\@PackageWarningNoLine{interfaces-tocloft}
2388     {Command \string\newlistentry\space could not be patched
2389      \MessageBreak \string\listofsetup\space will not work}}%

```

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

2391 \def\itf@tocloft#1#2#3{%
2392   \expandafter\def\expandafter#1\expandafter{%
2393     \expandafter\itf@cft@label
2394       \expandafter#1\expandafter{#1}{#2}{#3}}%
2395 }% \itf@tocloft
2396 \long\def\itf@cft@label#1#2#3#4{%
2397   \def#1{#2}% restore original
2398   \begin{group}
2399     \pgfkeysifdefined{/itf/tocloft/#3/bookmark}
2400     {%
2401       \raisedhyperdef{interfaces}{#4}{#2}%
2402       \pgfkeysgetvalue{/itf/tocloft/#3/bookmark/text}\itf@tp
2403       \ifx\itf@tp\@empty\else
2404         \pgfkeysgetvalue{/itf/tocloft/#3/bookmark}\@tempa
2405         \expandafter\bookmark\expandafter[%
2406           \expandafter{\@tempa,dest=interfaces.#4}]%
2407           {\pgfkeysvalueof{/itf/tocloft/#3/bookmark/text}}%
2408         \global\BKM@startatroottrue
2409       \fi
2410     {#2}%
2411     \pgfkeysgetvalue{/itf/tocloft/#3/label}\itf@tp
2412     \ifx\itf@tp\@empty\else
2413       \def@\currentHref{interfaces.#4}%
2414       \pgfkeysgetvalue{/itf/tocloft/#3/ref}@currentlabel
2415       \def@\currentlabelname{#2}%
2416       \label{\pgfkeysvalueof{/itf/tocloft/#3/label}}%
2417     \fi
2418   \end{group}
2419   \gdef\itf@tp{\def#1{#2}\let\itf@tp\@undefined}% restore original
2420   \aftergroup\itf@tp % simple group introduced by tocloft to print \contentsname
2421 }% \itf@cft@label

2422 </tocloft>
```

21.12 Interface `makecell` : `\repeatcell`

```
2423 <*makecell>
2424 \ProvidesPackage{interfaces-makecell}
2425   [2010/12/07 v2.5 repeatcell to fill in tabulars (FC)]
2426 \RequirePackage{interfaces-base}
2427 \itf@Check@Package{makecell}
2428 \edef\itf@AtEnd{\catcode126 \the\catcode126}% 126 = ~
2429 \AtEndOfPackage{\itf@AtEnd}
2430 \catcode126 \active % 126 = ~
2431 \itf@pgfkeys
```

21.12.1 `countvalues`

```
\countvalues

2432 \def\countvalues#1{\number\numexpr
2433   \expandafter\itf@countvalues\expandafter\#1,\countvalues\@nnil
2434 }% \countvalues
2435 \def\itf@countvalues#1#2,#3\@nnil{%
2436   \ifx\countvalues#3%
2437     +\#1\expandafter\remove@to\@nnil
2438   \else
2439     +1\expandafter\itf@countvalues
2440   \fi 1#3\@nnil
2441 }% \itf@countvalues

\itf@getvalue

2442 \def\itf@getvalue#1#2{\romannumeral-\`q
```

```

2443   \expandafter\itf@getv@lue\expandafter#1,\itf@getvalue@\nil{#2-1}\@nnil
2444 }% \itf@getvalue
2445 \def\itf@getv@lue#1#2,#3@\nil#4@\nnil{%
2446   \ifnum\numexpr#1=\numexpr#4\relax
2447     #2\expandafter\remove@to@nnil
2448   \else\ifx\itf@getvalue#3%
2449     \expandafter\expandafter\expandafter\remove@to@nnil
2450   \else
2451     \expandafter\expandafter\expandafter\itf@getv@lue
2452   \fi\fi{#1+1}#3@\nil#4@\nnil
2453 }% \itf@getv@lue

```

21.12.2 \repeatcell

\repeatcell

```

2454 \newif\ifitf@celltranspose
2455 \newrobustcmd*\repeatcell{\begingroup \iffalse{\fi \ifnum0='}\fi \itf@repeatcell}
2456 \newrobustcmd*\itf@repeatcell[3][]{%
2457   \ifblank{#1}{% alignment option ?
2458     {\let\itf@makecell@\firstofone}%
2459     {\def\itf@makecell##1{\noexpand\makecell[{#1}]{{#1}}}}%
2460   \countdef\itf@col=100% local
2461   \countdef\itf@row=101% local
2462   \countdef\itf@cnt=102% local
2463   \toksdef\itf@cell=1% local
2464   \itf@pgfkeys
2465   \pgfqkeys{/itf/makecell/repeatcell}{@reset={#2},#3}%
2466   \toks@{\ifnum0='{\fi \iffalse}\fi \endgroup}% local
2467   \ifnum\itf@row>\z@ % number
2468     \edef\itf@rowloop{\noexpand\@whilenum\itf@row<\the\itf@row}%
2469     \def\itf@cell@init{\def\r{\row}\let\R\empty}%
2470   \else % list
2471     \def\itf@rowloop{@for\r:=\itf@rows}%
2472     \expandafter\def\expandafter\itf@cell@init\expandafter{%
2473       \expandafter\def\expandafter\itf@cell@init\R\expandafter{%
2474         \expandafter\itf@getvalue\expandafter{\expandafter\@empty\itf@rows}}}%
2475   \fi
2476   \ifnum\itf@col>\z@ % number
2477     \edef\itf@colloop{\noexpand\@whilenum\itf@col<\the\itf@col}%
2478     \expandafter\def\expandafter\itf@cell@init\expandafter{%
2479       \itf@cell@init\def\c{\column}\let\C\empty}%
2480   \else % list
2481     \def\itf@colloop{@for\c:=\itf@cols}%
2482     \expandafter\expandafter\expandafter\def
2483       \expandafter\expandafter\expandafter\itf@cell@init
2484         \expandafter\expandafter\expandafter\expandafter\expandafter{%
2485           \expandafter\itf@cell@init
2486             \expandafter\def\expandafter\itf@cell@init\expandafter{%
2487               \expandafter\itf@getvalue\expandafter{\expandafter\@empty\itf@cols}}}%
2488   \fi
2489   \itf@cell@init % in case they are used in formulas
2490   \ifitf@celltranspose
2491     \itf@col\z@
2492     \itf@colloop\do{%
2493       \ifnum\itf@col>\z@
2494         \pgfkeysifdefined{/itf/makecell/repeatcell/cr/row\the\itf@col}%
2495           {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr/row\the\itf@col}\itf@thecr}%
2496           {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr}\itf@thecr}%
2497         \toks@\expandafter{\the\expandafter\toks@\itf@thecr}%
2498       \fi
2499       \advance\itf@col\@ne
2500       \itf@row\z@
2501       \itf@rowloop\do{%

```

```

2502          \advance\itf@row\@ne
2503          \ifnum\itf@row>\@ne
2504              \toks@\expandafter{\the\toks@@}\%
2505          \fi
2506          \toks@\expandafter{\the\expandafter\toks@ \itf@cell@init}\%
2507          \itf@cell@insert}\% rows loop
2508      }\% cols loop
2509  \else
2510  \itf@row\z@
2511  \itf@rowloop\do{%
2512      \ifnum\itf@row>\z@
2513          \pgfkeysifdefined{/itf/makecell/repeatcell/cr/row\the\itf@row}\%
2514              {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr/row\the\itf@row}\itf@thecr}
2515              {\pgfkeysgetvalue{/itf/makecell/repeatcell/cr}\itf@thecr}\%
2516          \toks@\expandafter{\the\expandafter\toks@ \itf@thecr}\%
2517      \fi
2518      \advance\itf@row\@ne
2519      \itf@col\z@
2520      \itf@colloop\do{%
2521          \advance\itf@col\@ne
2522          \ifnum\itf@col>\@ne
2523              \toks@\expandafter{\the\toks@@}\%
2524          \fi
2525          \toks@\expandafter{\the\expandafter\toks@ \itf@cell@init}\%
2526          \itf@cell@insert}\% cols loop
2527      }\% rows loop
2528  \fi
2529  \itf@message{Code introduced by repeatcell in the \currenvir:^^]
2530          \the\toks@}\% (to see the code...)
2531  \the\toks@ \ignorespaces
2532 }\% \itf@repeatcell

```

`\itf@message`

```
2533 \let\itf@message@gobble
```

Now the keys for `\repeatcell`:

```

2534 \def\itf@cell@rowcolkey{%
2535     \removefromstart\pgfkeyscurrentname{row}\%
2536     \ifin@\else
2537         \removefromstart\pgfkeyscurrentname{col}\%
2538     \fi
2539     \ifin@
2540         \itf@number\itf@cnt\pgfkeyscurrentname\relax\@nnil
2541             {\pgfkeyslet\pgfkeyscurrentkey\pgfkeyscurrentvalue}
2542             \in@false
2543     \fi
2544     \ifin@\else
2545         \pgfkeysvalueof{/errors/unknown key/.@cmd}\pgfkeyscurrentkey
2546                         \pgfkeyscurrentvalue
2547                         \pgfeov
2548     \fi
2549 }\% \itf@cell@rowcolkey
2550 \pgfqkeys{/itf/makecell/repeatcell}{%
2551     font/.store in=\itf@cellfont,%
2552     font/.unknown/.code=\itf@cell@rowcolkey,%
2553     text/.store in=\itf@celltext,%
2554         text/.default=\ifdefined\@C\@column\fi,%
2555     text/.unknown/.code=\itf@cell@rowcolkey,%
2556     cr/.initial=\@,,%
2557         cr/.default=\crrcr,cr/.blank=\@,,end/.alias=cr,%
2558     cr/.unknown/.code=\itf@cell@rowcolkey,%
2559     rows/.code=\itf@number\itf@row#1\relax\@nnil
2560             {\ifnum\itf@row<\z@\itf@row\@ne\fi}
2561             {\itf@row\z@}

```

```

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

```

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

```

2592 \def\itf@cell@getformula#1{%
2593   \itf@cell@getf@rmula #1$$$$\itf@cell@getformula$\relax\@nnil
2594   \expandafter\ifx\expandafter\toks@\the\toks@
2595     \itf@cell@formulaerror{\#1}\thr@@
2596   \fi
2597 }% \itf@cell@getformula
2598 \def\itf@cell@getf@rmula #1##2##3##4##5##6{%
2599   \toks@\{\toks@}%
2600   \ifx\itf@cell@getformula #4%
2601     \toks@\{\itf@cell@numcalc{\#1}\}%
2602   \else\ifx $#4%
2603     \ifx\itf@cell@getformula #6%
2604       \ifdef\pgfmathparse\else\itf@cell@formulaerror{\#2}4\fi
2605       \toks@\{\itf@cell@pgfcalc{\#2}\}%
2606     \else\ifx $#6%
2607       \ifdef\FPeval\else\itf@cell@formulaerror{\#3}5\fi
2608       \toks@\{\itf@cell@FPcalc{\#3}\}%
2609     \fi\fi
2610   \fi\fi\remove@to@nnil
2611 }% \itf@cell@getf@rmula
2612 \def\itf@cell@formulaerror#1#2{%
2613   \def\itf@tp{\#1}\@onelvel@sanitize\itf@tp
2614   \ifcase#2\@%
2615     \PackageError{interfaces-makecell}{%
2616       {Missing formula for key \pgfkeyscurrentname!}\@ehd
2617     \or %
2618     \PackageError{interfaces-makecell}{%
2619       {Computed key \pgfkeyscurrentname\space was left blank!}\@ehd
2620     \or %
2621     \PackageError{interfaces-makecell}{%
2622       {Duplicate definition for computed key \pgfkeyscurrentname\space will redefine the key}\@ehd
2623     }%
2624   }%
2625 }%

```

```
2624  \or %3
2625  \PackageError{interfaces-makecell}
2626    {Incorrect formula in \string\repeatcell
2627      \MessageBreak Check: \itf@tp
2628      \MessageBreak Take care of commas and equal signs...}@\ehd
2629 \or %4
2630 \PackageError{interfaces-makecell}
2631   {Formula: \itf@tp
2632     \MessageBreak requires \string\pgfmathparse:\space\string\usetikz{pgfmath}}@\ehd
2633 \or %5
2634 \PackageError{interfaces-makecell}
2635   {Formula: \itf@tp
2636     \MessageBreak requires \string\FPeval:\space\string\usepackage{fp}}@\ehd
2637 \else
2638 \PackageError{interfaces-makecell}
2639   {Impossible value!}@\ehd
2640 \fi
2641 }% \itf@cell@formulaerror
```

```

\itf@cell@calc
\itf@cell@numcalc 2642 \protected\def\itf@cell@calc#1#2{%
\itf@cell@pgfcalc 2643   \edef#1{\def\noexpand#1{#2}}%
\itf@cell@FPCalc 2644   \toks@\expandafter{\the\expandafter\toks@ #1}%
2645   #1%
2646 }% \itf@cell@calc
2647 \protected\def\itf@cell@numcalc#1#2{%
2648   \itf@cell@calc#2{\number\numexpr#1}%
2649 }% \itf@cell@numcalc
2650 \protected\def\itf@cell@pgfcalc#1#2{%
2651   \pgfmathparse{#1}%
2652   \itf@cell@calc#2\pgfmathresult
2653 }% \itf@cell@pgfcalc
2654 \protected\def\itf@cell@FPCalc#1#2{%
2655   \begingroup
2656   \setbox\z@\vbox{\%FPeval introduces some weird code, hence the calculation in a \vbox
2657     \FPeval\FPresult{#1}%
2658     \global\let\FPresult\FPresult}%
2659   \endgroup
2660   \itf@cell@calc#2\FPresult
2661 }% \itf@cell@FPCalc

```

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

```

2662 \def\itf@cell@getifdefined#1#2{%
2663   \pgfkeysifdefined{/itf/makecell/repeatcell/#1}%
2664     {\pgfkeysgetvalue{/itf/makecell/repeatcell/#1}#2}%
2665     {\let#2\empty}%
2666 }%
2667 \def\itf@cell@insert{%
2668   \pgfkeysvalueof{/itf/makecell/repeatcell/@calc}%
2669   \ifx\R\empty\else
2670     \toks@\expandafter{\the\expandafter\toks@
2671       \expandafter\def\expandafter\r\expandafter{\r}}%
2672   \fi
2673   \ifx\C\empty\else
2674     \toks@\expandafter{\the\expandafter\toks@
2675       \expandafter\def\expandafter\c\expandafter{\c}}%
2676   \fi
2677   \itf@cell@getifdefined{font/row\the\itf@row}\itf@cellfontrow
2678   \itf@cell@getifdefined{font/col\the\itf@col}\itf@cellfontcol
2679   \itf@cell@getifdefined{text/row\the\itf@row}\itf@cellrow
2680   \itf@cell@getifdefined{text/col\the\itf@col}\itf@cellcol
2681   \toks@\expandafter{\the\expandafter\toks@ \itf@cellfont}%
2682   \ifx\itf@cellrow\empty
2683     \ifx\itf@cellcol\empty
2684       \itf@cell\expandafter\expandafter\expandafter{%
2685         \expandafter\itf@cellfontcol \itf@cellfontrow}%
2686       \let\itf@tp\itf@celltext
2687     \else
2688       \itf@cell\expandafter{\itf@cellfontcol}%
2689       \let\itf@tp\itf@cellcol
2690     \fi
2691   \else
2692     \ifx\itf@cellcol\empty
2693       \itf@cell\expandafter{\itf@cellfontrow}%
2694       \let\itf@tp\itf@cellrow
2695     \else
2696       \itf@cell{}%
2697       \ifitf@celltranspose
2698         \edef\itf@tp{\noexpand\itf@diaghead(-2,1)
2699           {\expandonce\itf@cellfontrow\expandonce\itf@cellrow}
2700           {\expandonce\itf@cellfontcol\expandonce\itf@cellcol}}%
2701       \else

```

```

2702     \edef\itf@tp{\noexpand\itf@diaghead(-2,1)
2703         {\expandonce\itf@cellfontcol\expandonce\itf@cellcol}
2704         {\expandonce\itf@cellfontrow\expandonce\itf@cellrow}}%
2705     \fi
2706     \fi
2707     \fi
2708     \itf@cell\expandafter{\the\expandafter\itf@cell \itf@tp}%
2709     \edef\itf@tp{\itf@makecell{\the\itf@cell}}%
2710     \toks@\expandafter{\the\expandafter\toks@ \itf@tp}%
2711 }% \itf@cell@insert

```

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

```

2712 \def\itf@diaghead(#1)#2#3{%
2713   \begingroup
2714   \edef\itf@tp{%
2715     \noexpand\diaghead({#1}){\unexpanded{#2 #3}}%
2716         {\unexpanded{#2}}%
2717         {\unexpanded{#3}}}}%
2718   \expandafter\endgroup\itf@tp
2719 }% \itf@diaghead

```

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

`\CT@cellc`

```

2720 \AfterPackage*{colortbl}{%
2721 \def\CT@cellc#1[#2]#3{%
2722   #1 is either \@firstofone or \@gobble, [#2]#3 = color spec
2723   \xdef\CT@cell@color{\noexpand\CT@color#1[{{#2}}]}{#3}%
2724   \global\let\noexpand\CT@cell@color\relax}%
2724 }%

```

2725

21.13 † Interface ☈ `enumitem` : `\setlist`

```

2726 /*enumitem*/
2727 \ProvidesPackage{interfaces-enumitem}
2728   [2010/12/07 v2.5 more control on lists (FC)]
2729 \RequirePackage{interfaces-base}
2730 \itf@Check@Package{enumitem}
2731 \itf@pgfkeys

```

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

```

2732 \providerobustcmd*\interitem{%
2733   \item[]\hskip-\leftmargin
2734 }% \interitem

```

`\setlist` (extension, star form)

```

2735 \renewrobustcmd*\setlist{\begingroup \@ifstar
2736   {\let\do\csgappto \itf@setlist}
2737   {\let\do\csgdef \itf@setlist}%
2738 }% \setlist
2739 \newrobustcmd*\itf@setlist[2][0]{%
2740   \let\enit@a@\empty
2741   \let\enit@b@\empty
2742   @for\x:=#1\do{%
2743     \ifx\x@\empty\else
2744       \itf@number\count@\x\relax\@nnil
2745       {\edef\enit@b{\enit@b,\the\count@}%
2746       \edef\enit@a{\enit@a,\x}}%
2747     \fi}%
2748   \ifx\enit@a@\empty\def\enit@a{list}\fi
2749   \ifx\enit@b@\empty\def\enit@b{0}\fi
2750   @for\aa:=\enit@a\do{%

```

```

2751     \ifx\@empty\else
2752     \@for\b:=\enit@\b\do{%
2753         \ifx\b\@empty\else
2754             \ifcsundef{\end\@a}{%
2755                 \expandafter\itf@setlist@error\expandafter{\@a}%
2756                 {\do{\enit@@\@a\romannumeral\b}{, #2}}%
2757             }%
2758         \fi}%
2759     \endgroup
2760 }% \itf@setlist
2761 \def\itf@setlist@error#1{%
2762     \PackageError{interfaces-enumitem}{%
2763         Environment '#1' is undefined
2764         MessageBreak Use \string\newlist\space before \string\setlist
2765         MessageBreak Your command will be ignored}@\eha
2766 }% \itf@setlist@error
2767 </enumitem>

```

21.14 Interface `truncate`

```

2768 <*truncate>
2769 \ProvidesPackage{interfaces-truncate}

```

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

```

2770 \newrobustcmd{\trunc}[3]{\TruncateMarker}{% do nothing if width of #3 is less than #2
2771     \begingroup
2772         \hbadness\maxdimen\hfuzz\vsiz
2773         \setbox\z@=\hbox to\dimexpr#2\relax\bgrou\color@begingroup
2774                                         \ignorespaces#3\unskip
2775                                         \color@endgroup\egroup
2776         \ifnum\badness>\@M \truncate[{\#1}]{\#2}{\#3}%
2777     }%
2778         \def\next{\truncate[{\#1}]{\#2}{\#3}}%
2779     \else #3%\def\next{\#3}%
2780     \fi
2781 \endgroup
2781 }% \trunc{text}

```

\htruncadjust

```

2782 \newrobustcmd{\htruncadjust}[1]{\@testopt
2783     {\itf@htruncadjust{\#1}}{}%
2784 }% \htruncadjust
2785 \long\def\itf@htruncadjust[#2]#3{%
2786     %#1=total width, #2=[x_1..x_n], #3=text_1..text_n
2787     \begingroup
2788         \let\itf@trunc@width\empty
2789         \let\itf@istruncated@gobble % to remove the first comma
2790         \let\the@dimen\relax
2791         \count@\z@ \dimen@\z@
2792         \ifblank{\#2}{%
2793             no optional [x_1,...,x_n] parameters
2794             %
2795             \@for\x:=\#3\do{%
2796                 \advance\count@\ne
2797                 \setbox\z@\hbox\bgrou\color@begingroup
2798                     \ignorespaces\x\unskip
2799                     \color@endgroup\egroup
2800                     \dimen\count@ = \wd\z@
2801                     \toks\count@\expandafter{\the\dimen\count@}%
2802                     \advance\dimen@\dimen\count@}%
2803             }
2804             %
2805             \@for\x:=\#2\do{%
2806                 \advance\count@\ne
2807                 \toks\count@\expandafter{\x\p@}%
2808                 \advance\dimen@\x\p@
2809                 \dimen\count@ -\ne\p@}%

```

```

2808      }%
2809      \advance\count@\@ne \dimendef\itf@freespace=\count@
2810      \advance\count@\@ne \dimendef\itf@tempdim=\count@
2811      \advance\count@\@ne \dimendef\itf@sum=\count@
2812      \count@\z@
2813      \itf@freespace\z@ % free space left when w_i < x_i/sum(x_i) * total
2814      \@for\x:=#3\do{%
2815          \advance \count@\ @ne
2816          \ifdim\dimen@=\z@ \else
2817              \ifdim \dimen\count@<\z@ % not measured before
2818                  \setbox\z@\hbox\bgroup\color@begingroup
2819                      \ignorespaces\x\unskip
2820                      \color@endgroup\egroup
2821                  \dimen\count@ = \wd\z@
2822          \fi
2823          \itf@tempdim = \itf@scaledimen{\#1}*\the\toks\count@\dimen@\relax
2824          \ifdim \dimen\count@ > \itf@tempdim
2825              \dimen\count@ = \itf@tempdim
2826              \edef\itf@istruncated{\itf@istruncated,\the\count@}%
2827              \advance\itf@sum by\the\toks\count@
2828          \else
2829              \advance\itf@freespace \itf@tempdim
2830              \advance\itf@freespace -\dimen\count@
2831          \fi
2832      \fi
2833      \edef\itf@trunc@width{\itf@trunc@width\the\count@{\the@dimen\the\count@}}%
2834  }%
2835 %% Free space=\the\itf@freespace\par
2836 \ifdim\itf@freespace > \z@ % reinsert free space proportionally
2837     \ifx\itf@istruncated\gobble\else
2838         \@for\x:=\itf@istruncated\do{%
2839             \advance\dimen\x by\itf@scaledimen\itf@freespace*\the\toks\x\itf@sum\relax
2840         }%
2841     \fi\fi
2842     \def\the@dimen{\the\dimen}%
2843     \edef\itf@trunc@width{\itf@trunc@width}%
2844     \expandafter\endgroup
2845         \expandafter\def
2846             \expandafter\itf@trunc@width
2847                 \expandafter{\itf@trunc@width}%
2848 %% total=\the\dimexpr#1\relax
2849 }% \itf@htruncadjust
2850 \def\itf@scalefactor#1#2{\strip@pt\dimexpr#1*65536/#2\relax}
2851 \def\itf@scaledimen#1#2#3\relax{%
2852     \dimexpr\itf@scalefactor{#2}{#3}\dimexpr#1\relax\relax
2853 }% \itf@scaledimen

\htruncwidth

2854 \newcommand*\htruncwidth[1]{% purely expandable
2855     \expandafter\itf@htruncwidth\itf@trunc@width\relax\relax@nnil{#1}%
2856 }% \htruncwidth
2857 \def\itf@htruncwidth#1#2#3\@nnil#4{%
2858     \ifx\relax#1\itf@trunc@nosuchwidth{#4}%
2859     \else\ifnum#1=#4
2860         #2%
2861     \else
2862         \itf@htruncwidth#3\@nnil{#4}%
2863     \fi\fi
2864 }% \itf@htruncwidth
2865 \def\itf@trunc@nosuchwidth#1{%
2866     \PackageError{interfaces-truncate}{%
2867         {You requested the width for text nr#1}%
2868         \MessageBreak but there is no such text
2869         \MessageBreak Therefore no such width!}\@ehd

```

```
2870 }% \itf@trunc@nosuchwidth
```

```
2871 </truncate>
```

21.15 † Interface ↗ appendix and the appendices environment

```
2872 {*appendix}
```

```
2873 \ProvidesPackage{interfaces-appendix}
```

```
2874 [2010/12/07 v2.5 easy control of appendices environment (FC)]
```

```
2875 \RequirePackage{interfaces-base}
```

```
2876 \itf@Check@Package{appendix}
```

```
2877 \itf@pgfkeys
```

```
\appendices
```

```
2878 \let\itf@appendices@pp\appendices
```

```
2879 \let\itf@endappendices@pp\endappendices
```

```
2880 \let\itf@appendices@pp\appendices
```

```
2881 \let\itf@endappendices@pp\endappendices
```

```
2882 \renewenvironment{appendices}[1][]{%
```

```
2883   \pgfqkeys{/itf/appendix/appendices}{#1} %
```

```
2884   \itf@appendices@pp
```

```
2885 }
```

```
2886 {\itf@endappendices@pp} %
```

```
2887 \pgfqkeys{/itf/appendix/appendices}{%
```

```
2888   title/.store as code in cs=appendixpagename,%
```

```
2889     title/.append code=\ifx\appendixpagename\empty \@dopage@ppfalse
```

```
2890       \else \@dopage@pptrue\fi,%
```

```
2891     title/.value required,%
```

```
2892   intoc/.is if=@dotoc@pp,%
```

```
2893   intoc/title/.store as code in cs=appendixtocname,%
```

```
2894     intoc/title/.append code=\ifx\appendixtocname\empty \@dotoc@ppfalse
```

```
2895       \else \@dotoc@pptrue\fi,%
```

```
2896     intoc/title/.default=\pgfkeysvalueof{/itf/appendix/appendices/title},%
```

```
2897   section/.is enum={section,chapter},%
```

```
2898   section/.append code={\@chapter@ppfalse
```

```
2899     \ifnum\pgfkeysvalueof{/itf/appendix/appendices/section/.number}=1\z@
```

```
2900       \else \ifundefined{chapter}
```

```
2901         {\PackageWarning{interfaces-appendix}}
```

```
2902           {You asked chapter-style appendices but your
```

```
2903             \MessageBreak current class does not provide chapters!}}
```

```
2904           \@chapter@pptrue
```

```
2905         \fi},%
```

```
2906   chapter/.style={section=chapter},%
```

```
2907     chapter/.value forbidden,%
```

```
2908 }% keys for appendices environment
```

```
2909 </appendix>
```

21.16 † Interface ↗ embedfile : \embedthisfile

```
2910 {*embedfile}
```

```
2911 \ProvidesPackage{interfaces-embedfile}
```

```
2912 [2010/12/07 v2.5 easy macros to embed files into pdf (FC)]
```

```
2913 \RequirePackage{interfaces-base}
```

```
2914 \itf@Check@Package{embedfile}
```

```
2915 \newrobustcmd*\embedthisfile{\@ifstar
```

```
2916   \embedthisfile@main
```

```
2917   \embedthisfile@all
```

```
2918 }% \embedthisfile
```

```
2919 \@onlypreamble\embedthisfile
```

```
2920 \def\embedthisfile@main{\embedfile{\jobname.tex}}
```

```
2921 \def\embedthisfile@all{%
```

```
2922   \embedthisfile@main
```

```
2923   \AfterPreamble{%
```

```

2924     \ltx@ifundefined{@listfiles}
2925         {\let\@listfiles\@empty} to avoid: \global\let\@filelist\relax in \document
2926         {}%
2927     }%
2928     \gpreto\@enddocumenthook{%
2929         \let\do\itf@EmbedOne
2930         \expandafter\docs vlist\expandafter{\@filelist}%
2931     }%
2932 }% \embedthisfile@all
2933 \def\itf@EmbedOne#1{\itf@Embed@ne#1.tex\@nil}
2934 \def\itf@Embed@ne#1.tex#2\@nil{%
2935     \ltx@ifundefined{cp#1}
2936     {}
2937     {\ifblank{#2}{}
2938         {\embedfile{#1.tex}}}}%
2939 }% \itf@Embed@ne
2940 </embedfile>

```

21.17 † Interface ⇔ environ

```

2942 <*environ>
2943 \ProvidesPackage{interfaces-environ}
2944     [2010/12/07 v2.5 ReNewEnviron = RenewEnviron (FC)]
2945 \RequirePackage{interfaces-base}
2946 \AfterPackage{environ}{%
2947     \ifundef\ReNewEnviron
2948         {\let\ReNewEnviron\RenewEnviron}
2949         {}%
2950 }%
2951 </environ>

```

21.18 † Interface umrand

```

2952 <*umrand>
2953 \ProvidesPackage{interfaces-umrand}
2954     [2010/12/07 v2.5 patch for umrand.sty ifnotempty (FC)]
2955 \RequirePackage{interfaces-base}
2956 \BeforePackage{umrand}{%
2957     \let\itf@newif\newif
2958     \def\itf@tp{\ifnotempty}%
2959     \def\newif#1{\def\x{#1}\ifx\x\itf@tp \expandafter@gobble
2960         \else\expandafter\ltx@newif\fi{#1}}%
2961 }%
2962 \AfterPackage{umrand}{%
2963     \let\newif\itf@newif
2964     \let\itf@newif\@undefined
2965 }
2966 </umrand>

```

21.19 † Interface ⇔ scrlfile

```

2967 <*scrlfile>
2968 \ProvidesPackage{interfaces-scrlfile}
2969     [2010/12/07 v2.5 scrlfile hooks improved (FC)]
2970 \RequirePackage{interfaces-base}

```

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

\@clsextension

```

2971 \AfterEndPreamble{%
2972     \let\@pkgextension\ltx@pkgextension
2973     \let\@clsextension\ltx@clsextension
2974     \let\@ifpackageloaded\ltx@ifpackageloaded
2975     \let\@ifclassloaded\ltx@ifclassloaded
2976 }

```

\scr@load@hook

```

2977 \renewrobustcmd*\scr@load@hook[2]{%
2978   \csuse{\#2-\#1}%
2979   \ifcsname\#2-\#1-\string\keep\endcsname\else\global\csundef{\#2-\#1}\fi
2980 }% \scr@load@hook

\scr@set@load@hook (optimisation)

2981 \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 2982 \renewrobustcmd*\BeforeFile{\@ifstar
2983   {\itf@s@LoadFile{before}}%
2984   {\itf@LoadFile{before}}%
2985 }% \BeforeFile
2986 \renewrobustcmd*\AfterFile{\@ifstar
2987   {\itf@s@LoadFile{after}}%
2988   {\itf@LoadFile{after}}%
2989 }% \AfterFile

2990 \def\itf@s@LoadFile#1#2{%
2991   \global\cslet{\#2-\#1-\string\keep}{\empty}
2992   \scr@set@load@hook{\#1}{\#2}%
2993 }% \itf@s@LoadFile
2994 \def\itf@LoadFile#1#2{%
2995   \global\csundef{\#2-\#1-\string\keep}%
2996   \scr@set@load@hook{\#1}{\#2}%
2997 }% \itf@LoadFile

```

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

```

2998 \renewcommand*\@input[1]{\IfFileExists{\#1}%
2999   {\scr@load@hook{before}{\#1}%
3000     \@@input\@filef@und
3001     \scr@load@hook{after}{\#1}}%
3002   {\typeout{No file \#1.}}%
3003 }% \@input

3004 </scrfile>

```

21.20 ⇨ Interface pgfkeys ([⇨ pgfmanual](#))

```

3005 <*pgfkeys>
3006 \ProvidesPackage{interfaces-pgfkeys}
3007   [2010/12/07 v2.5 pgfkeys handlers definitions (FC)]
3008 \RequirePackage{interfaces-base}
3009 \ifdefined\pgfkeys@ifexecutehandler % pgf version 2.1
3010 \def\itf@pgfkeys{%
3011   \global\let\pgfkeys@unpack \itf@pgfkeys@unpack
3012   \global\let\pgfkeys@case@three \itf@pgfkeys@case@three
3013   \global\let\pgfkeys@case@three@handleall\itf@pgfkeys@case@three@handleall
3014 }% \itf@pgfkeys
3015 \else
3016 \def\itf@pgfkeys{%
3017   \global\let\pgfkeys@unpack \itf@pgfkeys@unpack
3018   \global\let\pgfkeys@case@three \itf@pgfkeys@case@three
3019   \global\let\pgfkeys@addvalue \itf@pgfkeys@addvalue
3020   \global\let\pgfkeys@exp@call \itf@pgfkeys@exp@call
3021 }% \itf@pgfkeys
3022 \fi
3023 \AtBeginDocument{\itf@pgfkeys \let\itf@pgfkeys \relax}

```

Some new pgfkeys macros

\pgfkeysmeaning

```
3024 \providecommand*\pgfkeysmeaning[1]{\pgfkeysifdefined{\#1}}
```

```

3025   {\expandafter\meaning\csname pgfk@#1\endcsname}%
3026   {\meaning\@undefined}%
3027 }% \pgfkeysmeaning

\pgfkeysvalueof@unexpanded

3028 \def\pgfkeysvalueof@unexpanded#1{%
3029   \unexpanded\expandafter\expandafter\expandafter{%
3030     \csname pgfk@#1\endcsname}%
3031 }% \pgfkeysvalueof@unexpanded

```

Some pgfkeys macros redefinition/corrections

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

```

3032 \ifdefined\pgfkeys@ifexecutehandler \else % pgf v2.1: bug fixed
3033 \def\itf@pgfkeysaddvalue#1#2#3{%
3034   \edef\pgfkeys@global@temp{%
3035     \unexpanded{#2}%
3036     \ifcsname pgfk@#1\endcsname
3037       \unexpanded\expandafter\expandafter\expandafter{%
3038         \csname pgfk@#1\endcsname}%
3039     \fi
3040     \unexpanded{#3}}%
3041   \pgfkeyslet{#1}\pgfkeys@global@temp
3042 }% \itf@pgfkeysaddvalue
3043 \fi

```

`\pgfkeysEsetvalue`

`\pgfkeysEaddvalue`

```

3044 \ifdefined\pgfkeys@ifexecutehandler \long \fi % pgf version 2.1
3045 \def\pgfkeysEsetvalue#1#2{%
3046   \expandafter\edef\csname pgfk@#1\endcsname{#2}%
3047 }% \pgfkeysEsetvalue
3048 \ifdefined\pgfkeys@ifexecutehandler \long \fi % pgf v2.1
3049 \def\pgfkeysEaddvalue#1#2#3{%
3050   \edef\pgfkeys@global@temp{%
3051     #2%
3052     \ifcsname pgfk@#1\endcsname
3053       \unexpanded\expandafter\expandafter\expandafter{%
3054         \csname pgfk@#1\endcsname}%
3055     \fi
3056     #3}%
3057   \pgfkeyslet{#1}\pgfkeys@global@temp
3058 }% \pgfkeysEaddvalue

```

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

```

3059 %% missed a pair of braces around the argument in pgfkeys.code.tex (pgf v2.0)
3060 %% ( the code could break if it contains = or , )
3061 \ifdefined\pgfkeys@ifexecutehandler\else % (bug corrected in pgf v2.1)
3062   \def\itf@pgfkeys@exp@call#1{\pgfkeysalso{\pgfkeyscurrentpath={#1}}}{}
3063 \fi

```

Handlers /blank

```

3064 %% The value is now checked by \itf@pgfkeys@default@blank@required:
3065 \ifdefined\pgfkeys@ifexecutehandler \long \fi % pgf v2.1
3066 \def\itf@pgfkeys@unpack#1=#2#3\pgfkeys@stop{%
3067   \pgfkeys@spdef\pgfkeyscurrentkey{#1}%
3068   \edef\pgfkeyscurrentkey{\pgfkeyscurrentkey}%
3069   \ifx\pgfkeyscurrentkey\pgfkeys@empty
3070     % Skip
3071   \else
3072     \pgfkeys@add@path@as@needed
3073     \pgfkeys@spdef\pgfkeyscurrentvalue{#2}%

```

```

3074     \itf@pgfkeys@default@blank@required % <FC>
3075             \pgfkeyscurrentkey % <FC>
3076             \pgfkeys@case@one % <FC>
3077     \fi
3078 }% \itf@pgfkeys@unpack

\itf@pgfkeys@default@blank@required                               Handle the case: ./def → ./blank → ./def ...
3079 \def\itf@pgfkeys@default@blank@required#1#2{%
3080     \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text% Hmm... no value
3081         \pgfkeysifdefined{#1/.@def}%
3082             {\pgfkeysgetvalue{#1/.@def}\pgfkeyscurrentvalue}
3083             {}% no default, so leave it
3084     \fi
3085     \ifx\pgfkeyscurrentvalue\empty
3086         \pgfkeysifdefined{#1/.@blank}
3087             {\pgfkeysgetvalue{#1/.@blank}\pgfkeyscurrentvalue}
3088             {}% no /.blank specified
3089     \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text% in case /.blank -> \pgfkeysnovalue
3090         \pgfkeysifdefined{#1/.@def}%
3091             {\pgfkeysgetvalue{#1/.@def}\pgfkeyscurrentvalue}
3092             {}% no default, so leave it
3093     \fi\fi
3094     \ifx\pgfkeyscurrentvalue\pgfkeysvaluerequired
3095         \pgfkeysvalueof{/errors/value required/.@cmd}\pgfkeyscurrentkey
3096                                         \pgfkeyscurrentvalue
3097                                         \pgfeov
3098     \else
3099         #2%
3100     \fi
3101 }% \itf@pgfkeys@default@blank@required

```

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

```

3102 %% To handle the cases of: /handler/.default= /handler/.blank= /handler/.value required,
3103 %% it is necessary to redefine \pgfkeys@case@three
3104 \def\itf@pgfkeys@case@three{%
3105     \pgfkeys@split@path
3106     \pgfkeysifdefined{/handlers/\pgfkeyscurrentname/.@cmd}%
3107         {%
3108             \itf@pgfkeys@default@blank@required{/handlers/\pgfkeyscurrentname}%
3109             {\pgfkeysgetvalue{/handlers/\pgfkeyscurrentname/.@cmd}\pgfkeys@code
3110                 \expandafter\pgfkeys@code\pgfkeyscurrentvalue\pgfeov}%
3111         }
3112         \pgfkeys@unknown % no handler, thus the key is unknown
3113 }% \itf@pgfkeys@case@three
3114 \ifdef{\pgfkeys@ifexecutehandler}
3115     \let\itf@pgfkeys@case@three@handleall \itf@pgfkeys@case@three
3116 \fi

```

Loading interfaces modifications for pgfkeys

```
3117 \itf@pgfkeys
```

Handler definition: `/alias`, .blank

```

3118 \pgfkeys{%
3119     /handlers/.alias/.code=\itf@pgfkeys@ifaliasexists{#1}{%
3120         \pgfkeyscedef\pgfkeyscurrentpath{%
3121             \noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}},%
3122             /handlers/.alias/.value required,%
3123             /handlers/.blank/.code=\pgfkeyssetvalue{\pgfkeyscurrentpath/.@blank}{#1},%
3124             /handlers/.blank/.default=\pgfkeysnovalue,%

```

```

3125 }% /.alias and /.blank
3126 \def\itf@pgfkeys@ifaliasexists#1#2{%
3127   \pgfkeysifdefined{\pgfkeysdefaultpath#1/.@cmd}{#2}%
3128   {\pgfkeysifdefined{\pgfkeysdefaultpath#1}%
3129     {#2}%
3130     {\itf@pgfkeys@aliaserror{#1}}}%
3131 }% \itf@pgfkeys@ifaliasexists
3132 \def\itf@pgfkeys@aliaserror#1{%
3133   \PackageError{interfaces-pgfkeys}%
3134   {You want the key \pgfkeyscurrentpath\space%
3135   \MessageBreak to be an alias for \pgfkeysdefaultpath#1%
3136   \MessageBreak but the key \pgfkeysdefaultpath#1%
3137   \MessageBreak is not defined!}\@eha\@gobble
3138 }% \itf@pgfkeys@aliaserror

```

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

Then, the `/.alias*` handler is a facility to create aliases for starred `*`, `+`, `-` keys in a batch (as long as the alias exists, of course).

```

3139 \pgfkeys{%
3140   /handlers/.alias*/.code=\itf@pgfkeys@ifaliasexists{#1}{%
3141     \pgfkeyscurrentpath{\noexpand\pgfkeysalso{\pgfkeysdefaultpath#1={##1}}}}%
3142     @tfor\x:=+-*\do{%
3143       \pgfkeysifdefined{\pgfkeysdefaultpath#1\x/.@cmd}%
3144         {\pgfkeyscurrentpath\x}{%
3145           \noexpand\pgfkeysalso{\pgfkeysdefaultpath#1\x={##1}}}}%
3146     {}%
3147   },%
3148   /handlers/.alias*/.value required,%
3149 }% /.alias*

```

Handler definition: `/.is private`

```

3150 \pgfkeys{%
3151   /handlers/.is private/.code=\pgfkeyscurrentpath{%
3152     \PackageError{#1}{Key \pgfkeyscurrentkey\space is private%
3153     \MessageBreak and should not be modified}\@ehd},%
3154   /handlers/.is private to class/.code=\pgfkeyscurrentpath{%
3155     \ClassError{#1}{Key \pgfkeyscurrentkey\space is private%
3156     \MessageBreak and should not be modified}\@ehd}%
3157 }% /.is private

```

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

```

3158 \def\itf@pgfkeys@choice[#1]#2#3{%
3159   \pgfkeys{\pgfkeyscurrentpath/.cd,%
3160     .is choice,%%
3161     #2/.code={#3},%
3162     .default/.expanded=@firstoftwo#1}%
3163 }% \itf@pgfkeys@choice
3164 \pgfkeys{%
3165   /handlers/.true/.code=\itf@pgfkeys@choice[{\true}{\false}]{\true}{#1},%
3166   /handlers/.false/.code=\itf@pgfkeys@choice[{\true}{\false}]{\false}{#1},%
3167   /handlers/.on/.code=\itf@pgfkeys@choice[{\on}{\off}]{\on}{#1},%
3168   /handlers/.off/.code=\itf@pgfkeys@choice[{\on}{\off}]{\off}{#1},%
3169 }% /.true/.false

```

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

```

3170 \pgfkeys{%
3171   /handlers/.store in cs/.code=\pgfkeysdef\pgfkeyscurrentpath{\csdef{\#1}{##1}},%
3172   /handlers/.estore in cs/.code=\pgfkeysdef\pgfkeyscurrentpath{\csedef{\#1}{##1}},%
3173 }% /.store in cs

```

Handler definition: `/.store as code`

```

3174 %% /.store as code handlers
3175 \pgfkeys{%
3176   /handlers/.store as code/.code={%
3177     \pgfkeyscurrentpath{%
3178       \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
3179       \pgfkeyslet{\pgfkeyscurrentpath}@empty
3180       \pgfkeyslet{\pgfkeyscurrentpath/.@def}@empty
3181       @tfor\x:=#1\do{%
3182         \pgfkeyscurrentpath\x}{%
3183           \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
3184             \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
3185             \noexpand\unexpanded{##1}}}}%
3186         \pgfkeyslet{\pgfkeyscurrentpath\x/.@def}@empty
3187       }},%
3188   /handlers/.store as code/.default=*,%
3189 }% /.store as code

```

Handler definition: `/.estore as code`

```

3190 \pgfkeys{%
3191   /handlers/.estore as code/.code={%
3192     \pgfkeyscurrentpath{%
3193       \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{##1}}%
3194       \pgfkeyslet{\pgfkeyscurrentpath}@empty
3195       @tfor\x:=#1\do{%
3196         \pgfkeyscurrentpath\x}{%
3197           \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
3198             \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
3199             \noexpand\unexpanded{##1}}}}%
3200       }},%
3201   /handlers/.estore as code/.default=*,%
3202 }% /.estore as code

```

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

```

3203 \pgfkeys{%
3204   /handlers/.store as code in cs/.code={%
3205     \pgfkeyscurrentpath{%
3206       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3207       \else\csdef{\#1}{##1}\fi}%
3208       @tfor\x:=*+\do{%
3209         \pgfkeyscurrentpath\x}{%
3210           \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
3211             \noexpand\ifcsname#1\endcsname
3212               \noexpand\expandafter\noexpand\expandonce
3213                 \noexpand\csname#1\endcsname
3214               \noexpand\fi
3215               \noexpand\unexpanded{##1}}}}%
3216         \pgfkeyslet{\pgfkeyscurrentpath\x/.@def}@empty
3217       }},%
3218   /handlers/.store as code in cs/.value required,%
3219 }% /.store as code in cs
3220 \pgfkeys{%
3221   /handlers/.estore as code in cs/.code={%

```

```

3222     \pgfkeysdef{\pgfkeyscurrentpath}{%
3223         \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3224             \else\csedef{#1}{##1}\fi}%
3225     \atfor\x:=*+\do{%
3226         \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3227             \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
3228                 \noexpand\ifcsname#1\endcsname
3229                     \noexpand\expandafter\noexpand\expandonce
3230                         \noexpand\csname#1\endcsname
3231                     \noexpand\fi
3232                     \noexpand\unexpanded{##1}}}}%
3233     },% default if \pgfkeysnovalue which expands to \@empty
3234     /handlers/.estore as code in cs/.value required,%
3235 }% /.estore as code in cs

```

Handler definition: `./store as color`

```

3236 \protected\def\itf@pgfk@color#1#2@nil{%
3237     \ifx\@let@token\color
3238         \def\pgfkeyscurrentvalue{#2}%
3239     \else\ifx\@let@token\normalcolor
3240         \def\pgfkeyscurrentvalue{default}%
3241     \else\ifx\@let@token\pgfkeysnovalue
3242         \def\pgfkeyscurrentvalue{default}%
3243     \fi\fi\fi
3244 }% \itf@pgfk@color
3245 \pgfkeys{%
3246     /handlers/.store as color/.code=%
3247         \pgfkeysedef{\pgfkeyscurrentpath}{%
3248             \futurelet\noexpand\@let@token\itf@pgfk@color
3249                 ##1\noexpand\pgfkeysnovalue\noexpand\@nil
3250             \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
3251 }% /.store as color

```

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

```

3252 \pgfkeys{%
3253     /handlers/.store as color in cs/.code=%
3254         \pgfkeysedef{\pgfkeyscurrentpath}{%
3255             \futurelet\noexpand\@let@token\itf@pgfk@color
3256                 ##1\noexpand\pgfkeysnovalue\noexpand\@nil
3257             \cslet{#1}\noexpand\pgfkeyscurrentvalue}%
3258 }% /.store as color in cs

```

Handler definition: `./store as color command`

```

3259 \protected\def\itf@pgfk@colorcmd#1@nil{%
3260     \ifcat @\noexpand\@let@token
3261         \def\pgfkeyscurrentvalue{\color{#1}}%
3262     \else\ifcat @\noexpand\@let@token
3263         \def\pgfkeyscurrentvalue{\color{#1}}%
3264     \else\ifx\@let@token\@nil
3265         \let\pgfkeyscurrentvalue\normalcolor
3266     \fi\fi\fi
3267 }% \itf@pgfk@colorcmd
3268 \pgfkeys{%
3269     /handlers/.store as color command/.code=%
3270         \pgfkeysedef{\pgfkeyscurrentpath}{%
3271             \futurelet\noexpand\@let@token\itf@pgfk@colorcmd
3272                 ##1\noexpand\@nil
3273             \noexpand\pgfkeyslet{\pgfkeyscurrentpath}\noexpand\pgfkeyscurrentvalue}%
3274             \pgfkeyssetvalue{\pgfkeyscurrentpath/.@def}\normalcolor
3275 }% /.store as color command

```

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

```

3276 \pgfkeys{%
3277   /handlers/.store as color command in cs/.code=%
3278   \pgfkeysedef\pgfkeyscurrentpath{%
3279     \futurelet\noexpand\@let@token\itf@pgfk@colorcmd
3280     ##1\noexpand\@nil
3281     \cslet{#1}\noexpand\pgfkeyscurrentvalue}%
3282   \pgfkeyssetvalue{\pgfkeyscurrentpath/.@def}\normalcolor
3283 }% /.store as color command in cs

```

Handler definition `/.store as number`

```

3284 \pgfkeys{%
3285   /handlers/.store as number/.code=%
3286   \pgfkeysedef\pgfkeyscurrentpath{%
3287     \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\numexpr##1\relax}%
3288   \pgfkeyslet\pgfkeyscurrentpath\z@
3289   \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
3290   \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3291   @tfor\x:=#1\do{%
3292     \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3293       \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
3294         \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
3295         \x
3296         \numexpr\unexpanded{##1}\relax}%
3297       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@\fi}%
3298       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@\fi}%
3299     }%,%
3300   /handlers/.store as number/.default=+-*,%
3301 }% /.store as number

```

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

```

3302 \pgfkeys{%
3303   /handlers/.store as number in cs/.code=%
3304   \pgfkeysdef\pgfkeyscurrentpath{%
3305     \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3306     \else\csdef{#1}{\numexpr##1\relax}\fi}%
3307   \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3308   @tfor\x:=+-*\do{%
3309     \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3310       \let\cs\noexpand\pgfkeys@temp{%
3311         \noexpand\ifcsname#1\endcsname
3312           #1\noexpand\else\if*\x @ne\else z@\fi\noexpand\fi}%
3313       \edef\cs\noexpand\pgfkeys@temp{%
3314         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3315           \noexpand\expandonce\noexpand\pgfkeys@temp
3316           \x
3317           \numexpr\noexpand\unexpanded{##1}\relax}%
3318         }\noexpand\pgfkeys@temp}%
3319       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@\fi}%
3320       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@\fi}%
3321     }%,%
3322   /handlers/.store as number in cs/.value required,%
3323 }% /.store as number in cs

```

Handler definition: `./estore as number`

```

3324 \pgfkeys{%
3325   /handlers/.estore as number/.code={%
3326     \pgfkeysedef\pgfkeyscurrentpath{%
3327       \noexpand\pgfkeysetvalue{\pgfkeyscurrentpath}{%
3328         \noexpand\number\numexpr##1\relax}%
3329       \pgfkeyslet\pgfkeyscurrentpath\z@%
3330       \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@%
3331       \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@%
3332     @tfor\x:=#1\do{%
3333       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3334         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3335           \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}%
3336           \x
3337           \numexpr##1\relax}%
3338     },%
3339   /handlers/.estore as number/.default=+-*,%
3340 }% /.estore as number

```

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

```

3341 \pgfkeys{%
3342   /handlers/.estore as number in cs/.code={%
3343     \pgfkeysedef\pgfkeyscurrentpath{%
3344       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3345         \else\csedef{\#1}{\number\numexpr##1\relax}\fi}%
3346     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3347     @tfor\x:=-*\do{%
3348       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3349         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3350           \noexpand\csname\noexpand\ifcsname#1\endcsname
3351             #1\noexpand\else\if*\x @ne\else z@\fi\noexpand\fi\endcsname
3352             \x
3353             \numexpr##1\relax}%
3354       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else z@\fi}%
3355       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else z@\fi}%
3356     },%
3357   /handlers/.estore as number in cs/.value required,%
3358 }% /.estore as number in cs

```

Handler definition: `./store in counter cs`

```

3359 \pgfkeys{%
3360   /handlers/.store in counter cs/.code={%
3361     \pgfkeysedef\pgfkeyscurrentpath{%
3362       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3363         \else\csname#1\endcsname\numexpr##1\relax\fi}%
3364     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3365     @tfor\x:=-*\do{%
3366       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3367         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3368           \noexpand\csname\noexpand\ifcsname#1\endcsname
3369             #1\noexpand\else\if*\x @ne\else z@\fi\noexpand\fi\endcsname
3370             \x
3371             \numexpr##1\relax}%
3372       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else z@\fi}%
3373       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else z@\fi}%
3374     },%
3375   /handlers/.store in counter cs/.value required,%
3376 }% /.store in counter cs

```

Handlers definition: `/store as glue`

```

3377 \pgfkeys{%
3378   /handlers/.store as glue/.code={%
3379     \pgfkeysedef{\pgfkeyscurrentpath}{%
3380       \noexpand\pgfkeysetvalue{\pgfkeyscurrentpath}{\glueexpr##1\relax}}%
3381     \pgfkeyslet{\pgfkeyscurrentpath}{z@skip}
3382     \pgfkeyslet{\pgfkeyscurrentpath/.@def}{z@skip}
3383     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}{z@skip}
3384     @tfor\x:=#1\do{%
3385       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3386         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded=%
3387           \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
3388           \x
3389           \if*\x\noexpand\itf@units@scaleNoCalc##1!
3390           \else\glueexpr\noexpand\unexpanded{##1}\relax\fi}}%
3391       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
3392       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
3393     }},%
3394   /handlers/.store as glue/.default=+-*,%
3395 }% /.store as glue

```

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

```

3396 \pgfkeys{%
3397   /handlers/.store as glue in cs/.code={%
3398     \pgfkeysedef{\pgfkeyscurrentpath}{%
3399       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3400       \else\csdef{#1}{\glueexpr##1\relax}\fi}%
3401     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}{z@skip}
3402     @tfor\x:=-*\do{%
3403       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3404         \letcs\noexpand\pgfkeys@temp{%
3405           \noexpand\ifcsname#1\endcsname
3406             #1\noexpand\else z@skip\noexpand\fi}%
3407           \edef\noexpand\pgfkeys@temp{%
3408             \noexpand\noexpand\noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3409               \noexpand\expandonce\noexpand\pgfkeys@temp
3410               \x
3411               \if*\x\noexpand\itf@units@scaleNoCalc##1!
3412               \else\glueexpr\noexpand\unexpanded{##1}\relax\fi}}%
3413             }\noexpand\pgfkeys@temp}%
3414       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@skip\fi}%
3415       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@skip\fi}%
3416     }},%
3417   /handlers/.store as glue in cs/.value required,%
3418 }% /.store as glue in cs

```

Handlers definition: `/estore as glue`

```

3419 \pgfkeys{%
3420   /handlers/.estore as glue/.code={%
3421     \pgfkeysedef{\pgfkeyscurrentpath}{%
3422       \itf@setlength\itf@sk{##1}%
3423       \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{%
3424         \noexpand\the\itf@sk}}%
3425     \pgfkeyslet{\pgfkeyscurrentpath}{z@skip}
3426     \pgfkeyslet{\pgfkeyscurrentpath/.@def}{z@skip}
3427     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}{z@skip}
3428     @tfor\x:=#1\do{%
3429       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3430         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3431           \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\x
3432           \if*\x\noexpand\itf@units@scale##1!
```

```

3433          \else\glueexpr##1\relax\fi}}%
3434      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@skip\fi}%
3435      \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@skip\fi}%
3436  },%
3437 /handlers/.estore as glue/.default=+-*,%
3438 }% /.estore as glue

```

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

```

3439 \pgfkeys{%
3440   /handlers/.estore as glue in cs/.code={%
3441     \pgfkeysdef\pgfkeyscurrentpath{%
3442       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3443       \else\itf@setlength\itf@sk{##1}%
3444       \csedef{#1}{\the\itf@sk}\fi}%
3445     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
3446     @tfor\x:=+-*\do{%
3447       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3448         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3449           \noexpand\csname\noexpand\ifcsname#1\endcsname
3450             #1\noexpand\else z@skip\noexpand\fi\endcsname
3451             \x
3452             \if*\x\noexpand\itf@units@scale##1!
3453             \else\glueexpr##1\relax\fi}}%
3454     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@skip\fi}%
3455     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@skip\fi}%
3456   },%
3457   /handlers/.etore as glue in cs/.value required,%
3458 }% /.estore as glue in cs

```

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

```

3459 \pgfkeys{%
3460   /handlers/.store in skip cs/.code={%
3461     \pgfkeysdef\pgfkeyscurrentpath{%
3462       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3463       \else\expandafter\itf@setlength\csname#1\endcsname{##1}\fi}%
3464     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@skip
3465     @tfor\x:=+-*\do{%
3466       \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3467         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3468           \csname#1\endcsname % should have been defined as a skip (not expandable)
3469           \x
3470           \if*\x\noexpand\itf@units@scale##1!
3471           \else\glueexpr##1\relax\fi}}%
3472     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@\ne\else\z@skip\fi}%
3473     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@\ne\else\z@skip\fi}%
3474   },%
3475   /handlers/.store in skip cs/.value required,%
3476 }% /.store in skip cs

```

Handlers definition: `.store as dimen`

```

3477 \pgfkeys{%
3478   /handlers/.store as dimen/.code={%
3479     \pgfkeysedef\pgfkeyscurrentpath{%
3480       \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\dimexpr##1\relax}}%
3481     \pgfkeyslet\pgfkeyscurrentpath\z@
3482     \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
3483     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3484     @tfor\x:=#1\do{%

```

```

3485     \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3486         \noexpand\pgfkeysalso{\pgfkeyscurrentpath/.expanded={%
3487             \noexpand\pgfkeysvalueof{unexpanded{\pgfkeyscurrentpath}}%
3488             \x
3489             \if*\x \noexpand\itf@units@scaleNoCalc##1!
3490             \else\dimexpr\noexpand\unexpanded{##1}\relax\fi}}}}%
3491     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
3492     \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
3493 },%
3494 /handlers/.store as dimen/.default=+-*,%
3495 }% /.store as dimen

```

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

```

3496 \pgfkeys{%
3497     /handlers/.store as dimen in cs/.code={%
3498         \pgfkeysedef\pgfkeyscurrentpath{%
3499             \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3500             \else\csdef{#1}{\dimexpr##1\relax}\fi}%
3501             \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3502             \@tfor\x:=+-*\do{%
3503                 \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3504                     \letcs\noexpand\pgfkeys@temp{%
3505                         \noexpand\ifcsname#1\endcsname
3506                         #1\noexpand\else\z@\noexpand\fi}%
3507                         \edef\noexpand\pgfkeys@temp{%
3508                             \noexpand\noexpand\noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3509                                 \noexpand\expandonce\noexpand\pgfkeys@temp
3510                                 \x
3511                                 \if*\x\noexpand\itf@units@scaleNoCalc##1!
3512                                 \else\dimexpr\noexpand\unexpanded{##1}\relax\fi}}%
3513                         }\noexpand\pgfkeys@temp}%
3514             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
3515             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
3516 },%
3517     /handlers/.store as dimen in cs/.value required,%
3518 }% /.store as dimen in cs {csname}{+-*}

```

Handlers definition: `./estore as dimen`

```

3519 \pgfkeys{%
3520     /handlers/.estore as dimen/.code={%
3521         \pgfkeysedef\pgfkeyscurrentpath{%
3522             \itf@setlength\itf@sk##1}%
3523             \noexpand\pgfkeysEsetvalue{\pgfkeyscurrentpath}{%
3524                 \noexpand\the\dimexpr\itf@sk}%
3525             \pgfkeyslet\pgfkeyscurrentpath\z@
3526             \pgfkeyslet{\pgfkeyscurrentpath/.@def}\z@
3527             \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3528             \@tfor\x:=#1\do{%
3529                 \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3530                     \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3531                         \noexpand\pgfkeysvalueof{\pgfkeyscurrentpath}\x
3532                         \if*\x \noexpand\itf@units@scale##1!
3533                         \else \dimexpr##1\relax\fi}}%
3534             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
3535             \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
3536 },%
3537     /handlers/.estore as dimen/.default=+-*,%
3538 }% /.estore as dimen

```

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

```

3539 \pgfkeys{%
3540   /handlers/.estore as dimen in cs/.code={%
3541     \pgfkeysdef\pgfkeyscurrentpath{%
3542       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3543       \else \itf@setlength\itf@sk{\#1}%
3544         \csedef{\#1}{\the\dimexpr\itf@sk}\fi}%
3545     \pgfkeyslet{\pgfkeyscurrentpath/.@blank}\z@
3546   @tfor\x:=+-*\do{%
3547     \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3548       \noexpand\ifcsname#1\endcsname
3549         \noexpand@\defaultunits \itf@sk
3550           \noexpand\csname#1\endcsname pt\relax\noexpand@nnil
3551         \noexpand\else \itf@sk \z@ \noexpand\fi
3552         \if*\x
3553           \itf@setlength\itf@sk {\itf@sk*\noexpand\itf@units@scale##1!}%
3554         \else
3555           \noexpand@\defaultunits@tempdima ##1pt \relax\noexpand@nnil
3556           \advance\itf@sk\x \atempdima
3557         \fi
3558         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=\itf@sk}%
3559       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
3560       \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
3561     },%
3562   /handlers/.etore as dimen in cs/.value required,%
3563 }% /.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.

```

3564 \pgfkeys{%
3565   /handlers/.store in dimen cs/.code={%
3566     \pgfkeysdef\pgfkeyscurrentpath{%
3567       \ifx\pgfkeyscurrentvalue\pgfkeysnovalue@text
3568       \else\expandafter\itf@setlength\csname#1\endcsname{\#1}\fi}%
3569     \pgfkeyslet{/pgfkeyscurrentpath/.@blank}\z@
3570   @tfor\x:=+-*\do{%
3571     \pgfkeysedef{\pgfkeyscurrentpath\x}{%
3572       \if*\x \else
3573         \noexpand@\defaultunits\itf@sk ##1pt \relax\noexpand@nnil
3574       \fi
3575         \noexpand\pgfkeysalso{\pgfkeyscurrentpath=%
3576           \csname#1\endcsname % has been defined as a dimen (not expandable)
3577           \x
3578           \if*\x \noexpand\itf@units@scale##1!
3579           \else\itf@sk\fi}%
3580         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@def}{\if*\x@ne\else\z@\fi}%
3581         \pgfkeysEsetvalue{\pgfkeyscurrentpath\x/.@blank}{\if*\x@ne\else\z@\fi}%
3582     },%
3583   /handlers/.store in dimen cs/.value required,%
3584 }% /.store in dimen cs

```

Handlers definition: `./is multiple choice`

```

3585 \pgfkeys{%
3586   /handlers/.is multiple choice/.code=%
3587     \pgfkeys{\pgfkeyscurrentpath/.cd,%
3588       .unknown/.style={%
3589         /errors/unknown choice value=\pgfkeyscurrentkey\pgfkeyscurrentvalue},%
3590         .code=\expandafter\pgfkeys\expandafter{\pgfkeyscurrentkey/.cd,###1}%
3591       },%
3592   /handlers/.is multiple choice/.value forbidden,%
3593 }% /.is multiple choice

```

Handlers definition: `/is enum`

```

3594 \pgfkeys{%
3595   /handlers/.is enum/.code={%
3596     \itf@cnt\z@
3597     \@for\x:=\pgfkeyscurrentvalue\do{%
3598       \pgfkeysedef{\pgfkeyscurrentpath/\x}{%
3599         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath}{\x}%
3600         \noexpand\pgfkeyssetvalue{\pgfkeyscurrentpath/.number}{\the\itf@cnt}%
3601       \ifnum\itf@cnt=\z@
3602         \pgfkeyslet{\pgfkeyscurrentpath/.def}\x
3603         \pgfkeyslet{\pgfkeyscurrentpath/.blank}\x
3604       \fi
3605       \advance\itf@cnt\@ne
3606     }%
3607     \pgfkeysalso{\pgfkeyscurrentpath/.is choice},%
3608   /handlers/.is enum/.value required,%
3609 }% /.is enum
3610 
```

3610 </pgfkeys>

21.21 Interface etoolbox - etoolbox-like commands

```

3611 (*etoolbox)
3612 \ProvidesPackage{interfaces-etoolbox}
3613   [2010/12/07 v2.5 some more etoolbox-like commands (FC)]
3614 \RequirePackage{interfaces-base}
3615 \begingroup\catcode`\&=3
3616   \long\gdef\etb@doitem#1&{\do{#1}}
3617 \endgroup

\preappto
\epreappto
\gpreappto 3618 \providerobustcmd\preappto[3]{%
3619   \edef#1{\unexpanded{#2}\ifdefined#1\expandonce#1\fi\unexpanded{#3}}%
3620 }% \preappto
3621 \providerobustcmd\epreappto[3]{%
3622   \edef#1{#2\ifdefined#1\expandonce#1\fi#3}%
3623 }% \epreappto
3624 \providerobustcmd*\gpreappto{\global\preappto}
3625 \providerobustcmd*\xpreappto{\global\epreappto}

\cspreappto
\csepreatto
\csgpreappto 3626 \providerobustcmd*\cspreappto[1]{%
3627   \csname itf@\ifcsname#1\endcsname
3628     preappto\else defpreapp\fi\expandafter\endcsname
3629     \csname#1\endcsname
3630 }% \cspreappto
3631 \long\def\itf@preappto#1#2#3{\edef#1{#2\expandonce#1#3}}
3632 \long\def\itf@defpreapp#1#2#3{\edef#1{#2#3}}
3633 \providerobustcmd*\csepreatto[1]{%
3634   \csname itf@\ifcsname#1\endcsname
3635     preappto\else defpreapp\fi\expandafter\endcsname
3636     \csname#1\endcsname
3637 }% \csepreatto
3638 \providerobustcmd*\csgpreappto{\global\cspreappto}
3639 \providerobustcmd*\csxpreappto{\global\csepreatto}

\removefrom
\eremovefrom
\gremovefrom
\xremovefrom 3640 \providerobustcmd\removefrom[2]{\in@false
3641   \long\def\itf@tp##1\itf@tp{%
3642     \in@true
3643     \edef#1{\unexpanded{##1}}%
3644     }\expandafter\itf@removefrom\expandafter{#1}{#2}%
3645 }% \removefrom

```

```
3646 \providerobustcmd\eremovefrom[2]{\begingroup
3647   \edef\x{\endgroup\removefrom\noexpand#1{#2}%
3648   }\x
3649 }% \eremovefrom
3650 \providerobustcmd\gremovefrom[2]{%
3651   \removefrom{#1}{#2}%
3652   \global\let#1=#1%
3653 }% \gremovefrom
3654 \providerobustcmd\xremovefrom[2]{%
3655   \eremovefrom{#1}{#2}%
3656   \global\let#1=#1%
3657 }% \xremovefrom

\itf@removefrom

3658 \long\def\itf@removefrom#1#2{%
  removes #2 (code) from #1 (hook)
3659   \def\in@##1#2##2\in@@{\itf@tp##1##2\in@@@}%
3660   \def\in@@@##1\in@@##2\in@@@{%
3661     \ifx\in@@##2\else##1\fi
3662   }\expandafter\in@@@\\in@@#1\itf@tp\in@@@#2\in@\in@@
3663 }% \itf@removefrom

\removefromstart
\eremovefromstart
\gremovefromstart
\xremovefromstart
3664 \newrobustcmd\removefromstart[2]{\in@false
3665   \long\def\itf@tp##1\itf@tp{%
3666     \in@true
3667     \edef#1{\unexpanded{##1}}%
3668   }\expandafter\itf@removefromstart\expandafter{#1}{#2}%
3669 }% \removefromstart
3670 \providerobustcmd\eremovefromstart[2]{\begingroup
3671   \edef\x{\endgroup\removefromstart\noexpand#1{#2}%
3672   }\x
3673 }% \eremovefromstart
3674 \providerobustcmd\gremovefromstart[2]{%
3675   \removefromstart{#1}{#2}%
3676   \global\let#1=#1%
3677 }% \gremovefromstart
3678 \providerobustcmd\xremovefromstart[2]{%
3679   \eremovefromstart{#1}{#2}%
3680   \global\let#1=#1%
3681 }% \xremovefromstart

\itf@removefromstart

3682 \long\def\itf@removefromstart#1#2{%
  removes #2 (code) from #1 (hook)
3683   \long\def\in@##1#2##2\in@@{##1\in@\itf@tp##2\in@@}%
3684   \long\def\in@@@##1\in@@##2\in@@@{%
3685     \ifx\in@@##2\fi
3686   }\expandafter\in@@@\\in@@#1\itf@tp\in@@@#2\in@@
3687 }% \itf@removefromstart

3688 </etoolbox>
```

22 History

[2010/12/07 v2.5]

- $\sim\backslash\text{sectionformat}$ now works with the `explicit` option of package `titlesec`. (in [interfaces-titlesec](#))
- Compatibility with `calc` package as been implemented for `pgfkeys` handlers:

| | | |
|-------------------------------|-------------------------------------|---------------------------------|
| <code>.estore as dimen</code> | <code>.estore as dimen in cs</code> | <code>.store in dimen cs</code> |
| <code>.estore as glue</code> | <code>.estore as glue in cs</code> | <code>.store in skip cs</code> |

As a result, it is now possible to write:

```
\sectionformat\section{labelsep=\widthof{\ some text }}  
\tocsetup{ section/pagenumbers/width=\widthof{ 00.00\enspace } }
```

For scaling operations, `interfaces` keeps the shrink and stretch components of glues, while `calc` discard them.

- $\sim\backslash\text{listofsetup}$ now works properly ([interfaces-tocloft](#)).
- Some `+` and `*` keys added to [interfaces-fancyhdr](#).
- Fix a bug of `titlesec` when inside `.dtx` files: `\input{ttlkeys.def}` shall be done with `%` as a comment character (`\BeforeFile... \AfterFile` does the trick).

[2010/11/26 v2.4]

- [interfaces-fancyhdr](#) : complete key set for header, footer and offset settings
Now keys like `foot/inner` or `head/outer/offset` are available (ouf!).
- [interfaces-hypbmsec](#) has been completely rewritten and is now independant of `titlesec` and `interfaces-titlesec`.
`\lastsectiontitle` etc. are defined by the mean of `\GetTitleString` (from H.O. eponym package).
`interfaces-hypbmsec` now provides `\lastsectionanchor` when used with `hyperref`.
- [interfaces-marks](#) revisited: expandable macros `\thefirstmarks`, `\thebotmarks` and `\thetopmarks` are now expandable in exactly one step of expansion.
`\marksthe` and `\marksthecs` does not use `\unexpanded` (better for the `.log` file to see the marks).
Optimization for `\getthemarks` with incidence on $\sim\backslash\text{ifmarksequal}$.
- $\sim\backslash\text{repeatcell}$: modification of the code. & alignment tab character is not activated and the “dirty trick” (according to D. Knuth) `\iffalse{\fi \ifnum0='}\fi` is used instead.
- $\sim\backslash\text{htruncadjust}$ produced an error when used with empty text arguments – fixed.
- Modification in `\bottomtitles`: `\bottomtitlespace` should be stored as a dimen, not a skip.
- $\sim\backslash\text{removefromstart}$ is now a `\long` macro.

[2010/11/21 v2.3]

- [interfaces-fancyhdr](#) : new keys and new aliases for header, footer and offset settings.

[2010/11/16 v2.2]

- Fix a bug in [interfaces-tocloft](#) (key `section/dotsep`).

[2010/11/15 v2.1]

- Improved default and blank values for bookmark levels, based on `\tt11@<section-name>`
- Correction of a bug in `\setlist`: missed a comma
- Correction of a bug in [interfaces-fancyhdr](#) (fixed in `./fancy@hf` handler).

[2010/10/31 v2.0]

- Comptability of [interfaces-pgfkeys](#) with `pgf` version 2.1
- Modification of the key `depth` for $\sim\backslash\text{tocsetup}$ to be able to change the t.o.c. depth anywhere inside the document (just like bookmarks depths).
- handler `./defcounter` has been replaced by more powerful `./store in counter cs`

[2010/10/27 v1.2]

- `interfaces-hypbmsec` needed some corrections.
- The `offset` keys for `fancyhdr` didn't work. Fixed.
- Correction of conversion `pt` to `mu` (catcodes...)
- Added `microtype` to the documentation...

[2010/10/20 v1.0]

- First version.

Appendix: README (for CTAN)

The `interfaces` package provides a small number of macros to modify the settings of a large number of parameters provided by Most Frequently Used Package (MFUP) for typesetting beautiful document with L^AT_EX.

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

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

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

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

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

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

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

`interfaces` is mainly based on `scrlfile.sty` (provided in the Koma-Script bundle, but Koma-Script classes are not required), on `pgfkeys` and `etoolbox`.

`interfaces` defines some `pgfkeys` handlers designed specially to make key-value interface for other packages. Using those handlers allows to make an interface in a few minutes...

23 Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; plain numbers refer to the code lines where the entry is used.

| Symbols | |
|--|--|
| <code>\#</code> | 81, 89, 98, 730, 742 |
| <code>\&</code> | 3615 |
| <code>\@input</code> | 3000 |
| <code>\@PackageInfoNoLine</code> | 735,
818, 995, 2194, 2351, 2358, 2365, 2372, 2379, 2386 |
| <code>\@PackageWarningNoLine</code> | 69, 737,
821, 997, 2197, 2353, 2360, 2367, 2374, 2381, 2388 |
| <code>\@addtoreset</code> | 574, 582, 620, 622 |
| <code>\@break@tfor</code> | 210 |
| <code>\@cftmakeloftitle_(patch)</code> | 2349 |
| <code>\@cftmakelottitle_(patch)</code> | 2349 |
| <code>\@cftmaketoctitle_(patch)</code> | 2349 |
| <code>\@cftpnumoff</code> | 2187, 2331 |
| <code>\@cftpnumon</code> | 2186, 2330 |
| <code>\@clsextension</code> | 2971 |
| <code>\@defaultunits</code> | 133,
134, 216, 231, 278, 280, 282, 779, 3549, 3555, 3573 |
| <code>\@elt</code> | 1022, 1040 |
| <code>\@expandtwoargs</code> | 210, 225, 226 |
| <code>\@filef@und</code> | 3000 |
| <code>\@filelist</code> | 2925, 2930 |
| <code>\@firstofone</code> | 2458, 2721 |
| <code>\@fourthoffive</code> | 971 |
| <code>\@ifclassloaded</code> | 2975 |
| <code>\@ifpackageloaded</code> | 2974 |
| <code>\@input_(patch)</code> | 2998 |
| <code>\@mainaux</code> | 817, 930, 935 |
| <code>\@onelevel@sanitize</code> | 1079, 2613 |
| <code>\@onlypreamble</code> | 51, 2919 |
| <code>\@pkgextension</code> | 2971 |

| | | |
|------------------------|--|------------|
| \@tfor | 208, 3142, 3181, 3195,
3208, 3225, 3291, 3308, 3332, 3347, 3365, 3384,
3402, 3428, 3446, 3465, 3484, 3502, 3528, 3546, 3570 | 3049, 3626 |
| \@whilenum | 2468, 2477 | 2736, 2981 |
| \bookmarks@patch | 816 | 3626 |
| \newlistof@patch | 2349 | 3626 |
| A | | |
| \active | 2430 | 3626 |
| \addcontentsline@patch | 729 | 3626 |
| \AddLineBeginMainAux | 436 | 3626 |
| \addtocontents | 2058 | 3626 |
| \AfterFile | 448, 2178, 2982 | 3626 |
| \aftergroup | 885, 926, 2420 | 3626 |
| \AfterPackage | 54,
67, 176, 254, 302, 725, 1077, 1090, 2720, 2946, 2962 | 3626 |
| \allpgfmodules | 313, 401, 418 | 3626 |
| \alltikzlibraries | 323, 402 | 3626 |
| \alltikzpackages | 308, 397, 400, 415 | 3626 |
| \appendices | 2878 | 3626 |
| \apptocmd | 76, 96, 2377, 2384 | 3626 |
| B | | |
| \BeforeFile | 439, 2174, 2982 | 3626 |
| \BeforePackage | 37, 2956 | 3626 |
| \BKM@ref | 973 | 3626 |
| \BKM@color | 848, 858, 866, 878 | 3626 |
| \BKM@dest | 969, 971 | 3626 |
| \BKM@page | 953 | 3626 |
| \bookmark | 6, 816, 819, 822, 940, 957, 2405 | 3626 |
| \BookmarkAtEnd | 936 | 3626 |
| \bookmarkget | 894, 896, 904, 906 | 3626 |
| \bookmarksetup | 6, 762, 764, 766, 854, 872, 899, 909 | 3626 |
| \botmarks | 2003, 2021, 2023 | 3626 |
| \bottomtitles | 5, 772 | 3626 |
| \bottomtitlespace | 782 | 3626 |
| \BS@BKM@FontStyle | 850, 862, 869, 879, 892, 902 | 3626 |
| C | | |
| \c | 204, 2479, 2486, 2554, 2673 | 3626 |
| \c@ | 2479, 2481, 2675 | 3626 |
| \calc@Adimen | 163 | 3626 |
| \calc@B | 173, 174 | 3626 |
| \calc@Bdimen | 163, 173, 174 | 3626 |
| \calc@multiply@by@real | 162 | 3626 |
| \cftdot | 2090 | 3626 |
| \cftdotfill | 2301 | 3626 |
| \cftdotsep | 2087, 2304 | 3626 |
| \cftnodots | 2089, 2309, 2310 | 3626 |
| \changefont | 3, 251, 286 | 3626 |
| \chapter | 1109, 2261, 2900 | 3626 |
| \color | 240, 241, 289, 293, 1180, 1200, 1341,
1359, 1377, 1395, 1413, 1431, 1449, 1467, 1485,
1503, 1521, 1539, 2081, 2286, 2318, 3237, 3261, 3263 | 3626 |
| \color@begingroup | 2773, 2795, 2818 | 3626 |
| \color@endgroup | 2775, 2797, 2820 | 3626 |
| \colorlet | 237 | 3626 |
| \column | 2479, 2554, 2585 | 3626 |
| \columnsep | 2107, 2136 | 3626 |
| \columnseprule | 2106, 2131 | 3626 |
| \columnseprulecolor | 2108 | 3626 |
| \contentsname | 2165, 2210, 2211, 2350, 2420 | 3626 |
| \count | 133 | 3626 |
| \countdef | 2460, 2461, 2462 | 3626 |
| \countvalues | 12, 2432 | 3626 |
| D | | |
| \default@color | 241 | 3626 |
| \diaghead | 2715 | 3626 |
| \dimen | 2798, 2799,
2800, 2807, 2817, 2821, 2824, 2825, 2830, 2839, 2842 | 3626 |
| \dimendef | 2809, 2810, 2811 | 3626 |
| \dump | 93 | 3626 |
| E | | |
| \embedfile | 2920, 2939 | 3626 |
| \embedthisfile | 19, 2915, 2918, 2919 | 3626 |
| \endlinechar | 443, 446 | 3626 |
| \epreappto | 3618 | 3626 |
| \eremovefrom | 3640 | 3626 |
| \eremovefromstart | 3664 | 3626 |
| F | | |
| \f@baselineskip | 266, 269, 276 | 3626 |
| \f@linespread | 266, 269, 275, 276 | 3626 |
| \f@size | 266, 269, 276 | 3626 |
| \fancy@setoffs | 1714, 1739, 1764, 1789, 1814, 1839, 1864, 1889 | 3626 |
| \fancyhf | 1122 | 3626 |
| \fancyhoffset | 1123 | 3626 |
| \figure | 2257 | 3626 |
| \firstmarks | 1997, 2013, 2020, 2023 | 3626 |
| \fontdimen | 192, 198 | 3626 |
| \fontencoding | 257 | 3626 |
| \fontfamily | 259 | 3626 |
| \fontseries | 261 | 3626 |
| \fontshape | 263 | 3626 |
| \footrule | 1202, 1203, 1204, 1205 | 3626 |
| \footrulewidth | 1208, 1209, 1210 | 3626 |
| \FPeval | 2607, 2636, 2656, 2657 | 3626 |
| \FPresult | 2657, 2658, 2660 | 3626 |
| \futurelet | 81, 144, 3248, 3255, 3271, 3279 | 3626 |
| G | | |
| \getrefbykeydefault | 953 | 3626 |
| \getthebotmarks | 2020 | 3626 |
| \getthefirstmarks | 2020 | 3626 |
| \getthemarks | 11, 2013, 2020, 2021, 2022, 2026, 2027 | 3626 |
| \getthetopmarks | 2020 | 3626 |
| \GetString | 1020, 1038 | 3626 |
| \GetStringResult | 1020, 1038 | 3626 |
| \globaldefs | 1023, 1041 | 3626 |
| \gluetomu | 189 | 3626 |
| \gpreappto | 3618 | 3626 |
| \gremovefrom | 3640 | 3626 |
| \gremovefromstart | 3664 | 3626 |
| H | | |
| \hbs@checkarg | 989 | 3626 |
| \hbs@do@patch | 983 | 3626 |
| \hbs@process@patch | 992 | 3626 |
| \headrule | 1182, 1183, 1184, 1185 | 3626 |

| | | | |
|--|------------------------------------|---|------------------------------------|
| \headrulewidth | 1188, 1189, 1190 | \itf@hbs@last@starred@ttl | 1005, 1032, 1036 |
| \headwidth 1189, 1210, 1963, 1964, 1965, 1968, 1969, 1972 | | \itf@hbs@process | 993, 1008 |
| \htruncadjust | 18, 2782 | \itf@hbs@ssect | 988, 1002 |
| \htruncwidth | 2854 | \itf@htruncadjust | 2783, 2785, 2849 |
| \hyper@anchor | 1011 | \itf@htruncwidth | 2855, 2857, 2862, 2864 |
| \HyperLocalCurrentHref | 1049 | \itf@makecell | 2458, 2459, 2709 |
| \HyperRaiseLinkHook | 793, 794, 795, 796 | \itf@marks@new | 1980, 1984, 1991 |
| \HyperRaiseLinkLength | 797 | \itf@message | 2529, 2533 |
| \hypersetup | 2287, 2319 | \itf@musk | 115, 2086, 2087, 2306, 2307 |
| I | | | |
| \ifcsundef | 477, | \itf@Not@titlesec | 1003, 1009, 1076 |
| 480, 521, 580, 587, 615, 1147, 1151, 1985, 2754 | | \itf@number | 132, 2540, 2559, 2587, 2744 |
| \ifcsvoid | 673, 674, 675 | \itf@pagesetup | 1127, 1128, 1130, 1132 |
| \ifitf@celltranspose | 2454, 2490, 2697 | \itf@patch | 81, |
| \ifmarksequal | 11, 2023 | 732, 816, 992, 2349, 2356, 2363, 2370, 2377, 2384 | |
| \ifnotempty | 2958 | \itf@patch@inside@ | 101, 107, 113 |
| \ifttl@explicit | 481, 530 | \itf@pgfk@color | 3236, 3244, 3248, 3255 |
| \ifundef | 1109, 2900, 2947 | \itf@pgfk@colorcmd | 3259, 3267, 3271, 3279 |
| \ignorespaces | 2531, 2774, 2796, 2819 | \itf@pgfkeys@aliaserror | 3130, 3132, 3138 |
| \in@ 210, 225, 226, 415, 418, 1080, 3661, 3662, 3683, 3685 | | \itf@pgfkeys@case@three | 3012, 3018, 3104, 3113, 3115 |
| \interitem | 16, 2732 | \itf@pgfkeys@case@three@handleall | 3013, 3115 |
| \itf@AfterPackage | 64 | \itf@pgfkeys@default@blank@required | |
| \itf@BKM@enable | 826, | 3064, 3074, 3079, 3108 | |
| 828, 829, 830, 831, 832, 833, 927, 928, 932, 934 | | \itf@pgfkeys@ifaliasexists | 3119, 3126, 3131, 3140 |
| \itf@BKM@pageref | 940 | \itf@preappto | 3631 |
| \itf@BKM@ref | 957 | \itf@preaptocmd | 74, 84, 732 |
| \itf@BKM@restore | 885, 913, 917, 926 | \itf@rem@mu | 182, 186 |
| \itf@BKM@write | 817, 925 | \itf@removefrom | 3644, 3658 |
| \itf@calc@Askip | 163, 177 | \itf@removefromstart | 3668, 3682 |
| \itf@calc@Bskip | 163, 177 | \itf@repeatcell | 2455, 2456, 2532 |
| \itf@calc@multiply@by@real | 162, 172, 175 | \itf@resetlastsections | 1022, 1040, 1095 |
| \itf@cell@calc | 2642 | \itf@scaledimen | 2823, 2839, 2851, 2853 |
| \itf@cell@formulaerror | | \itf@scalefactor | 2850, 2852 |
| 2569, 2570, 2572, 2595, 2604, 2607, 2612, 2641 | | \itf@sectionformat | 455, 456, 458 |
| \itf@cell@FPcalc | 2608, 2642 | \itf@setbottomtitlespace | 775, 778, 781 |
| \itf@cell@getformula | 2574, 2592 | \itf@setlength | 150, |
| \itf@cell@init | 2469, | 178, 3422, 3443, 3463, 3522, 3543, 3553, 3568 | |
| 2472, 2478, 2479, 2483, 2485, 2489, 2506, 2525 | | \itf@setlength@calc | 161 |
| \itf@cell@insert | 2507, 2526, 2662 | \itf@setlist | 2736, 2737, 2739, 2760 |
| \itf@cell@numcalc | 2585, 2586, 2601, 2642 | \itf@setlist@error | 2755, 2761, 2766 |
| \itf@cell@pgfcalc | 2605, 2642 | \itf@sk | 114, 278, 279, 280, 281, 282, 283, |
| \itf@celltransposefalse | 2579 | 776, 779, 782, 3422, 3424, 3443, 3444, 3522, 3524, | |
| \itf@cft@label | 2391 | 3543, 3544, 3549, 3551, 3553, 3556, 3558, 3573, 3579 | |
| \itf@cft@new@section | 2274, 2275, 2334 | \itf@strip@at | 118, 508, 984 |
| \itf@cft@newlistof | 2053, 2203, 2204, 2205, 2378 | \itf@strip@dot | 118, 191, 197 |
| \itf@cft@newsection | | \itf@strip@mu | 181, 2087, 2307 |
| 604, 2274, 2337, 2340, 2342, 2345, 2347, 2385 | | \itf@tk | 116, 1961 |
| \itf@cft@secname | 2246, 2247, 2249, 2330, 2331 | \itf@tocloft | 2193, 2350, 2357, 2364, 2391, 2395 |
| \itf@cft@sectionname | 2246 | \itf@ttl@BKM | 733, 743, 771 |
| \itf@countvalues | 2433, 2435, 2439, 2441 | \itf@ttl@createsection | 593, 616 |
| \itf@diaghead | 2698, 2702, 2712 | \itf@ttl@getformat | 468 |
| \itf@expandonceifcs | 123, 2562, 2590 | \itf@ttl@getspacing | 468 |
| \itf@getvalue | 2442, 2474, 2487 | \itf@ttl@newsection | 476, 612 |
| \itf@hbs@c1@ORI | 1060, 1062, 1063 | \itf@ttl@sectionformat | 467 |
| \itf@hbs@currentname | | \itf@ttl@setkey | 489, |
| 984, 1004, 1012, 1014, 1028, 1035, 1050, 1057 | | 493, 495, 496, 497, 498, 499, 510, 511, 512, 513, 514 | |
| \itf@hbs@currenttitle | 1020, 1024, 1038, 1042 | \itf@ttl@setup | 468 |
| \itf@hbs@do | 983, 990, 991 | \itf@ttl@titleclass | 560, 620, 622 |
| \itf@hbs@last | 1005, 1017, 1033, 1055, 1092, 1093 | \itf@ttl@titleclass@i | 563, 564, 566, 578 |
| \itf@hbs@last@nostar | 1037 | \itf@ttl@titleclass@ii | 574, 579, 592 |
| \itf@hbs@last@nostar@nottl | 1014, 1037 | \itf@ttl@update | 471, 516 |
| \itf@hbs@last@nostar@nottl@Hy | 1012, 1037 | \itf@undefinedpagestyle | 1148, 1152, 1155, 1158 |
| \itf@hbs@last@nostar@ttl | 1017, 1037 | \itf@units@mu | 201, 2086, 2306 |
| \itf@hbs@last@starred | 1004, 1019, 1053 | \itf@units@mutopt | 188, 229 |
| | | \itf@units@pt | 220 |
| | | \itf@units@pttomu | 188, 214 |

| | | | |
|-------------------------------------|--|---|--|
| \itf@units@scale | 151, 3432, 3452, 3470, 3532, 3553, 3578 | \pgfkeysmeaning | 3024 |
| \itf@units@scale@normalize | 153, 154, 158, 160, 179 | \pgfkeysnovalue | 2071, 3089, 3124, 3233, 3241, 3249, 3256 |
| \itf@units@scale@normalize@calc | 161 | \pgfkeysvalueof@unexpanded | 3028, 3184, 3198, 3294, 3387, 3487 |
| \itf@units@scale@normalizeNoCalc | 159, 160 | \pgfkeysvaluerequired | 3094 |
| \itf@units@scaleNoCalc | 159, 3389, 3411, 3489, 3511 | \pgfmathparse | 2604, 2632, 2651 |
| \itf@usetikzcode | 404, 413 | \pgfmathresult | 2652 |
| L | | | |
| \label | 2416 | \preappto | 3618 |
| \lastchapteranchor | 1113 | \pretocmd | 75, 96, 992, 2192, 2349, 2356, 2363 |
| \lastchapterbookmark | 1112 | R | |
| \lastchaptertitle | 1110 | \R | 2469, 2473, 2669 |
| \lastchaptertos | 1111 | \r | 2469, 2471, 2671 |
| \lastsection | 1000 | \raisedhyperdef | 6, 789, 2401 |
| \lastsectionname | 1028, 1050, 1102 | \ratio | 168, 169 |
| \listfigurename | 2222, 2223, 2357 | \rem@pt | 275 |
| \listofsetup | 9, 2041, 2200, 2376, 2383, 2390 | \remove@to@nnil | 128, 2437, 2447, 2449, 2610 |
| \listtablename | 2228, 2229, 2364 | \removefrom | 24, 3640 |
| \LoadInterface | 6, 7, 9, 39 | \removefromstart | 2535, 2537, 3664 |
| \lofsetup | 2041, 2362 | \ReNewEnviron | 2947, 2948 |
| \loop | 108 | \RenewEnviron | 2948 |
| \lotsetup | 2041, 2369 | \repeat | 112 |
| \ltx@clsextension | 2973 | \repeatcell | 12, 2454, 2591, 2623, 2626 |
| \ltx@ifclassloaded | 2975 | \row | 2469, 2586 |
| \ltx@ifpackageloaded | 52, 56, 65, 68, 235, 239, 2097, 2974 | S | |
| \ltx@pkgextension | 2972 | \scr@load@hook | 2977, 2999, 3001 |
| M | | | |
| \makecell | 2459 | \scr@set@load@hook_(optimisation) | 2981 |
| \marks | 42, 1981, 1989 | \sectionformat | 4, 454, 764 |
| \marksthe | 11, 667, 1979 | \selectfont | 252 |
| \marksthecs | 1983 | \set@fontsize | 266, 269, 276 |
| \maxdimen | 2772 | \setlength | 164 |
| \muexpr | 196 | \setlist | 16, 16 |
| \multicolssep | 2104 | \setlist_(extension,_star_form) | 2735 |
| \muskip | 188, 201, 220 | \skip | 188, 201, 220 |
| \mutoglu | 196 | \strip@pt | 2850 |
| N | | | |
| \newcounter | 581 | \strut | 1166 |
| \newlist | 2764 | \ subparagraph | 2256 |
| \newlistof | 2053, 2370, 2373, 2375, 2377, 2380, 2382 | \subsubsection | 2258 |
| \newmarks | 1992 | \subsubsection | 2254 |
| \newmuskip | 115 | \subtable | 2260 |
| \newskip | 114, 177 | T | |
| \newtoks | 116 | \table | 2259 |
| \normalbaselineskip | 1231 | \textfont | 192, 198 |
| \normalcolor | 2141, 3239, 3265, 3274, 3282 | \textheight | 779 |
| P | | | |
| \pagesetup | 10, 1124, 1168, 1961 | \the@dimen | 2789, 2833, 2842 |
| \pagestyle | 1139, 1147, 1170, 1970 | \thebotmarks | 1995 |
| \paragraph | 2255 | \thefirstmarks | 11, 1995 |
| \patchcmd | 83, 101, 106, 816, 2370 | \thepage | 949, 966 |
| \pdfstringdefDisableCommands | 255 | \thetopmarks | 1995 |
| \pgfkeys@case@three | 3012, 3018, 3103 | \thispagestyle | 1151 |
| \pgfkeys@case@three@handleall | 3013 | \tikzset | 304 |
| \pgfkeys@exp@call | 3020, 3059 | \titleclass | 435, 568, 571, 596 |
| \pgfkeys@global@temp | 3034, 3041, 3050, 3057 | \titleformat | 435 |
| \pgfkeys@ifexecutehandler | 3009, 3032, 3044, 3048, 3061, 3065, 3114 | \titlespacing | 435 |
| \pgfkeys@unpack | 3011, 3017 | \tocloftpagestyle | 2061 |
| \pgfkeys@saddvalue_(bug_correction) | 3032 | \tocsetup | 8, 8, 2041, 2355 |
| \pgfkeysEaddvalue | 2160, 2166, 2575, 3044 | \toks | 2799, 2805, 2823, 2827, 2839 |
| \pgfkeysEsetvalue | 3044 | \toksdef | 2463 |
| | | \topmarks | 2009, 2022 |
| | | \truncate | 2776, 2777 |
| | | \TruncateMarker | 2770 |
| | | \trunctext | 18, 2770 |
| | | \ttl@finmarks | 1005, 1017, 1032, 1054, 1090 |

| | | |
|--------------------|-------------------|-------------------------------|
| \ttl@getkeys | 518, 519 | V |
| \ttl@spacing@ii | 553 | \vsize 2772 |
| \ttlh@runinifempty | 449 | \vskip 1190, 1209, 1210 |
| \ttlp@append | 430, 437 | |
| | | W |
| U | | \write 817, 930, 935 |
| \unless | 111 | |
| \unskip | 2774, 2796, 2819 | X |
| \usepgfmodule | 401, 419 | \xpreappto 3618 |
| \usetikz | 3, 405, 406, 2632 | \xremovefrom 3640 |
| \usetikzlibrary | 391, 402, 420 | \xremovefromstart 3664 |