LaTeX Style for FAO Yearbook *

Boris Veytsman†

2013/03/23, v1.11

Abstract

This package provides class for typesetting FAO Yearbook

Contents

†borisv@lk.net, boris@varphi.com

List of Figures

1 Introduction

FAO UN publishes yearbooks of statistical data. This class is intended to typeset them. It was commissioned by FAO, and the copyright rests with UN.

2 User Guide

2.1 Installation

The class relies on the following software and packages installed on your system:

1. Fonts and LaTeX font support packages arev [1] and paratype [2].
2. graphics bundle [3].
3. xcolor package [4].
4. fancyhdr package [5].
5. TikZ and PGF packages [6].
6. geometry package [7].
7. paralel and pdfcolparallel packages [8, 9].
8. float and caption packages [10, 11].
9. booktabs and longtable packages [12, 13].
10. lscape package [14].
11. xkeyval package [15].
12. array package [16].
13. multicol package [**?**].
14. hhline package [17].

The installation of the class follows the usual practice [18] for LaTeX packages:

1. Run latex on `faoyearbook.ins`. This will produce the LaTeX class `faoyearbook.cls`.
2. Put the file `faoyearbook.cls` to the place where LaTeX can find it (see [18] or the documentation for your TeX system).
3. Update the database of file names. Again, see [18] or the documentation for your TeX system for the system-specific details.
4. The file `faoyearbook.pdf` provides the documentation for the package (this is the file you are probably reading now).

As an alternative to items 2 and 3 you can just put the file `faoyearbook.cls` in the working directory where your `.tex` file is.

2.1.1 Packages troubleshooting

Paratype Installing the PT sans font from package paratype can prove difficult. On windows, this should not make any major problem with MikTex. On Linux on the other side, it be be advantageous to use the debian package: http://www.bilbo.dynip.com/debian/dists/woody/updates/binary-all/paratype_20110509-1_all.deb. In case a message such as:

```
!pdfTeX error: pdflatex (file ectt3583): Font ectt3583 at 600 not found
```

arises, one should (re) install the package ec, at best from the cm-super bundle available in Debian/Ubuntu repos.

Geometry Another issue that can happen is a message like:

```
! Missing number, treated as zero.
<to be read again>
                   \Gm@cnth
```

This is due to an outdated version of geometry. Version 5.6 will work.

2.2 Invocation

To use the class, put in the preamble of your document

`\documentclass[`⟨*options*⟩`]{faoyearbook}`

option `print` *option* `web` *option* `issuu`

If the option `web` (default) is chosen, the pages of the book have the dimensions corresponding to A4 paper. However, if the option `print` is chosen, then the pages are printed on a wider area, and crop marks are added for the trimming.

If the option `issuu` is chosen, the internal links are transformed to external in the form suitable for http://www.issuu.com. Note that this option probably does not make much sense unless `web` option is also chosen. However, it is still possible to select both `print` and `issuu` option if someone needs it for an obscure purpose.

`altMargins` `altMarginsNarrow`

Some regional books are printed on A4 paper with slightly different margins. These options select alternative margins—either normal or narrow. Note that these options automatically select `print` setup, and should not be combined with the `print` option.

option `Draft` *option* `draft`

The option `Draft` (note the capitalization!) leads to the the large word 'DRAFT' printed across the pages. The standard LaTeX option `draft` leads to the same result, but it also makes other changes, most notably, in the behavior of the `\includegraphics` command and warnings.

`\ifprint`

It is possible to query the current mode using the macro `\ifprint`, for example

```
\ifprint
  Stuff for print version
\else
  Stuff for web version
\fi
```

Any branch of this conditional may be empty, so web-only stuff can be coded as

```
\ifprint\else Web-only stuff\fi
```

\includegraphics

There is a special facilty for \includegraphics command to choose a file depending on the current mode of the package. Namely, if there is a file image_print.pdf visible by LaTeX, then the commands \includegraphics{image} or \includegraphics{image.pdf} selects the file image_print.pdf. In the case this file is not found, the file image.pdf is selected instead. Similarly in the web mode the file image_web.pdf will be selected first, and only if it does not exist, image.pdf is selected. This rule works also for commands \includeLargeGraphics and \includeExtraLargeGraphics described below.

Note that at this time there is no similar facility for the \input command.

2.3 Setting Parameters

\faoset

Some parameters in the class can be set with the command \faoset{⟨*key=value*⟩}, for example

```
\faoset{bgcolor=blue}
```

Most of the parameters are explained below.

2.4 Fonts

\narrowfamily
\textnarrow
\captionfamily
\textcaption

The class uses PT Sans fonts [2] for body text and Arev fonts [1] for math. It defines two additional families: Narrow and Caption, corresponding to the PT Sans Narrow and PT Sans Caption font. They can be selected by the declarations \narrowfamily and \captionfamily or by the commands \textnarrow{⟨*text*⟩} and \textcaption{⟨*text*⟩} following the usual LaTeX conventions. Note that since PT Sans does not provide math alphabet, this choice does not change the mathematical text.

PT Sans Narrow may be useful for typesetting tables, for example,

```
\begin{tablepages}
 \scriptsize\narrowfamily
 \rowcolors{4}{@bgcolor!30}{@bgcolor!20}
 \input{./Tables/P1.DEM_1.tex}
\end{tablepages}
```

2.5 Setting Color

\setbgcolor

Each part of the Yearbook is typeset with it own background color. It is used a theme in the "special sections" (see below), indicator pages, etc. The color is selected with the macro \setbgcolor[⟨*color*⟩], where [⟨*color*⟩] can be in any form acceptable by xcolor package [4], for example

```
\setbgcolor{blue}
\setbgcolor{-green!25!blue!60}
\setbgcolor{rgb,9:red,4;green,2;yellow,1}
```

The command must be issued *before* the corresponding `\part` command.

key `bgcolor` An alternative way is to use the key-value interface with the command `\faoset`:

```
\faoset{bgcolor=rgb,9:red,4;green,2;yellow,1}
```

key `tableheadcolor` The key `tableheadcolor` sets the color for the headers of tables defined by `H` or `P` key (see Section 2.8). Normally it is 30% of the current `bgcolor` color, but it can be set to any required value.

`\selectcolor` The command `\setbgcolor` by itself does not change the background color: it is changed by the next `\part` macro. However, if by eny reason you need to change the color *before* the next `\part`, you can do it using the command `\selectcolor`, which is similar to the `\selectfont` command of the standard LaTeX. In most cases the user probably should not employ this command directly.

2.6 Setting spacing

key `tocpartskip` If necessary user can make the table of contents stretch using the parameter `tocpartskip` (by default, zero), for example:

```
\faoset{tocpartskip=12pt plus 12pt}
```

Not that this command must be issued *before* `\tableofcontents`. This parameter gives the extra space before the Part entries in the table of contents.

2.7 Sectioning

`\part` `\section` `\subsection` The main division of the text are `\part`s. The command `\part{`⟨*title*⟩`}` is used for numbered parts, while the command `\part*{`⟨*title*⟩`}` is used for unnumbered parts. The next division are `\section`s and `\subsection`s. They are never numbered. The style does not use `\chapter`s.

`\EndPartIntro` The sections immediately following new parts are special: they are typeset in one column on specially colored pages. The command `\EndPartIntro` switches to the "normal" sections.

`\appendix` Some parts of the text belong to *appendices*. The usual command `\appendix` should be issued before these parts. The parts typed after this command have special formatting—in particular, they do not produce colored bands in the table of contents.

2.8 (Non)Floats

Illustrative materials in LaTeX are usually called floats because they "float" from page to page, and TeX processor finds the best place for them. We still use the same name, but our floats *do not* float: instead they are pinned to the place where they are defined (so "nonfloats" might be a better name for them). Normally they occur in the graphics part of the indicator pages (see Section 2.11), but the class does not produce an error if they are in any other place of the book. There are three types of floats defined by the class:

`table` `chart` `map` `minitab`

table tabular material,

chart plots and other charts,

map mapped data.

minitab mini tables.

`caption` Each of these kinds of material is typeset using the corresponding environment: `table`, `chart`, `minitab` or `map`. Note that the caption for each of these environments *must* precede the graphical material, for example:

```
\begin{chart}
  \centering
  \caption{Hunger Data}
  \label{chart:hunger}
  \includegraphics{hunger.pdf}\\
\end{chart}
```

`\footnotebar` The source for the graphics sometimes is put after the graphics itself. In this case the command `\footnotebar` produces a bar separating the source from the graphics, for example

```
\begin{chart}
  \centering
  \caption{Hunger Data}
  \label{chart:hunger}
  \includegraphics{hunger.pdf}
  \footnotebar
   Source: FAO, Statistics Division, WEB: \url{http://www.fao.org}
\end{chart}
```

`\listoftables` `\listofcharts` `\listofmaps` The standard LaTeX has the command `\listoftables` to produce the list of tables in the document. Our class retains this command and produces two additional commands `\listofcharts` and `\listofmaps` with the obvious meaning.

key `floatwidth` Normally each (non)float besides long tables is set in the box with the width equal to one column of text and the height equal to the natural height of the float (including the caption). The width of the float can be changed with the key `floatwidth`:

```
\faoset{floatwidth=0.5\columnwidth}
```

key `fixedfloatheight`

For gridded floats it is useful to make the floats to have the same height to align in the adjacent columns. To achieve this effect one can set the Boolean key `fixedfloatheight`, normally `false`, to `true`:

```
\faoset{fixedfloatheight=true}
```

key `floatheight`

By default the height of the float is chosen to be 0.45 of the height of the page (two floats per column). One can override this decision by setting `floatheight` to another value, for example, to get three plots per column, one sets

```
\faoset{floatheight=0.3\textheight}
```

key `fixedcaptionheight` *key* `captionheight`

For gridded floats it is useful to typeset the caption in a box of fixed height, so the graphics is aligned. When the Boolean key `fixedcaptionheight`, normally `false`, is set to `true`, the captions are typeset to fixed height, which is equal to `captionheight` (normally equal to `2\baselinskip`). For example, if all captions are one line long, one can use

```
\faoset{fixedcaptionheight=true, captionheight=2\baselineskip}
```

2.9 Tables

`tablepages`

Long tables at the end of sections are typeset in the special style: no headers of footers, slightly colored pages, small margins. This style is selected by the environment `tablepages`:

```
\begin{tablepages}
  TABLES
\end{tablepages}
```

key `tablebg`

Normally the color selected by the `tablepage` style is the same as the main color of the section. However, it can be changed by the command `\aoset` using the key `tablebg`, for example

```
\faoset{tablebg=blue!5}
\pagestyle{tablepage}
```

The rules for typesetting tables can be found in booktabs and longtable packages. Here we just remind the rules:

1. A table starts with `\toprule` and ends with `\bottomrule`. The header is separated with `\midrule`.

2. For `longtable`s the commands `\endfirsthead` and `\endhead` denotes the header on the first and subsequent pages.

3. The command `\endfoot` dentes the footer repeated on every page.

4. Caption can be set on the first header. The command `\caption[]{`⟨*Caption*⟩`}` is used on the subsequent heads.

By default the rows of a table are colored using the background color of the current part.

key `d` *key* `H` *key* `P` *key* `C`

To typeset numericall items one should use `d` column identifier with the format `d{`⟨*a.b*⟩`}`, where a is the number of decimal in the integer part of the number, and b is the number of decimal digitst in the fractional part. For example, a number 12.345 corresponds to `d{2.3}`. The column headers are usually *not* numerical, so one need to use `\multicolumn` entries to typeset them. The class defines several such entries:

H produces a centered entry.

P produces an entry of a given length, for example, `P{1.5cm}`

C produces an entry of the length corresponding to the given number of numerical columns. For example, `C{2}` corresponds to a header of two numerical columns. Each column is assumed to be of the size enough to store -99.999.

`\hhline`

Unfortunately, the command `\cmidrule` from the `booktabs` package is not working well with the colored headings: the columns that do not the rule display a white band instead. Therefore for the rules that do not span the table width `\hhline{`⟨*specificaiton*⟩`}` command from the `hhline` package should be used. The `{`⟨*specification*⟩`}` argument of this command has many variants, but for our purposes we need only one variant: the command `-` produces a horizontal line spanning one column. The color of this line is determined by the command `\arrayrulecolor{`⟨*color*⟩`}`, issued in the last `>{`⟨*argument*⟩`}` command before the `-` specification. Therefore the command `>{\arrayrulecolor{@tableheadcolor}}-` produces a line of the color `@tableheadcolor`, which is seen as the absence of line. The command `>{\arrayrulecolor{black}}---` produces a black line spanning three columns. Thus if we have a four-column table and want a rule spanning columns 2–3, the following command should be issued:

```
\hhline{>{\arrayrulecolor{@tableheadcolor}}-% Column 1, no rule
  >{\arrayrulecolor{black}}--%  Columns 2 and 3, black rule
  >{\arrayrulecolor{@tableheadcolor}}-}% Column 4, no rule
```

The usual `*` specification may be used for repeating patterns, for example, `*{5}{-}` is equivalent to `-----`.

The vertical bar `|` specification in the `\hhline` argument means an interruption of the line. The interruption is by defalut a black interval, to make it the same color as the header background, use `>{\arrayrulecolor{@tableheadcolor}}|`.

2.10 Large Maps

`\includeLargeGraphics`
`\includeExtraLargeGraphics`

There is a special case of very large maps that can take either three or even four columns of width. They can be typeset in a spread, when the left part takes a recto page while the right part takes a verso page of an indicator (see Section 2.11). The special commands `\includeLargeGraphics{`⟨*file*⟩`}` and `\includeExtraLargeGraphics{`⟨*file*⟩`}` are used to typeset three column and four column maps correspondingly. Since they take care to align the graphics vertically and assume the existence of a caption, they do not work outside a (non)float.

A three-column map can be included inside `indicator` (see Section 2.11) in the following way:

```
\begin{inicator}
  \begin{indicatorText}
    Text part
  \end{indicatorText}
  \begin{indicatorGraphics}
    \begin{map}
      \caption{A Caption}
      \label{map:ALargeMap}
      \includeLargeGraphics{large_map.pdf}
    \end{map}
  \end{indicatorGraphics}
\end{inicator}
```

For a four-column map we need `indicatorC`:

```
\begin{inicatorC}
  \begin{indicatorGraphics}
    \begin{map}
      \caption{A Caption}
      \label{map:AVeryLargeMap}
      \includeExtraLargeGraphics{very_large_map.pdf}
    \end{map}
  \end{indicatorGraphics}
\end{inicatorC}
```

key `leftfraction`

Normally the commands `\includeLargeGraphics` and `\includeExtraLargeGraphics` leave 0.36 or 0.5 of the image on the left and put the rest of the image on the right. However, you can change this using the key `leftfraction`, for example, the command

```
\includeLargeGraphics[leftfraction=0.4]{large_map.pdf}
```

splits the image as 4 : 6.

`\largeGraphicsNotes`

All the material typed before `\includeLargeGraphics` is typeset before the graphics material. All the material typed after `\includeLargeGraphics` is typeset in the recto page after the right part of the graphics. However, sometimes

one need to typeset some material on the verso page after the left part of the map. The command `\largeGraphicsNotes{`⟨*material*⟩`}` should be issued before the graphics. The `{`⟨*material*⟩`}` is saved and typeset after the graphics, for example

```
\begin{map}
  \caption{Caption}
  \largeGraphicsNotes{% Material to be typeset on the left
    \begin{itemize}
    \item One
    \item Two
    \end{itemize}}
  \includeLargeGraphics{map}
  Material to be typeset on the right
\end{map}
```

`graphicKey` Sometimes a large map requires a small rectangular key. The environment `graphicKey` typesets its contents at the right lower corner of the map. If it contains more than an `\includegraphics` command, you need to tell TEX the width of the corresponding box. For this you can use the optional parameter, for example

```
\begin{graphicKey}
  \includegraphics{key}
\end{graphicKey}
or
\begin{graphicKey}[4cm]
  \includegraphics{key}\\
  This is the key
\end{graphicKey}
```

Note that the key should be *outside* the `map` environment, otherwise TEX cannot position it properly.

key `lefparttoffset` *key* `rightpartoffset` Sometimes the parts of the large graphics must be slighly moved to the left or to the right. The keys `leftpartoffset` and `rightpartoffset` allow this manipulation, for example

```
\includeLargeGraphics[leftpartoffset=-1cm,
       rightpartoffset=1cm]{large_map.pdf}
```

The default values for `\includeLargeGraphics` are

```
leftpartoffset=-20pt, rightpartoffset=0pt
```

2.11 Indicator Pages

The main illustrative device for the book is called *indicator*. Indicator pages consist of text and illistrative material (tables, graphs, maps). The text and the illustrations are visually separated and flow independently.

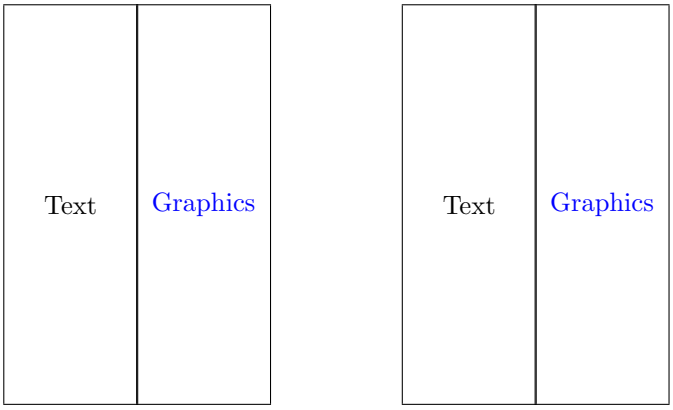

Figure 1: Indicator, Variant A

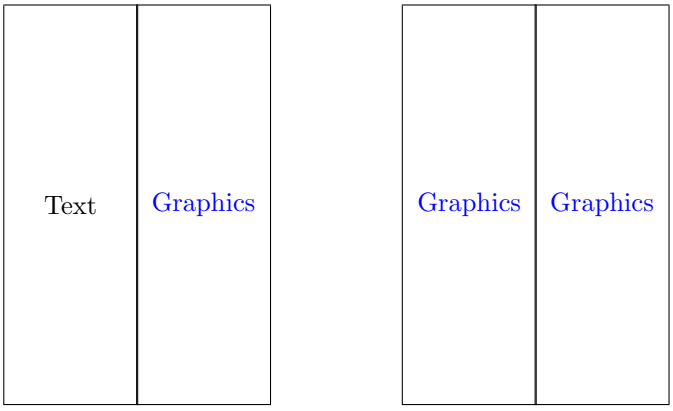

Figure 2: Indicator, Variant B_1

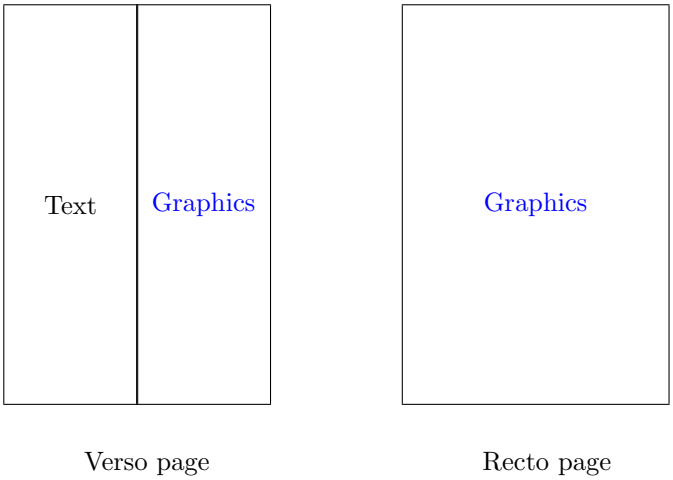

Figure 3: Indicator, Variant B_2

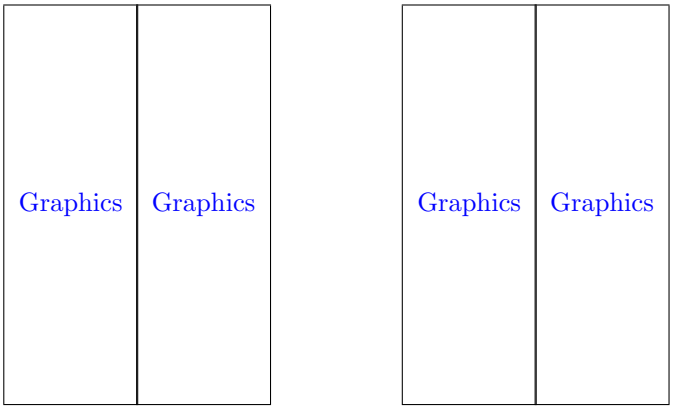

Figure 4: Indicator, Variant C_1

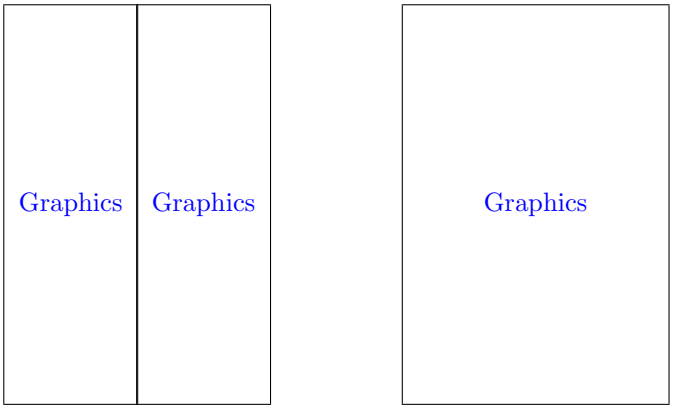

Figure 5: Indicator, Variant C_2

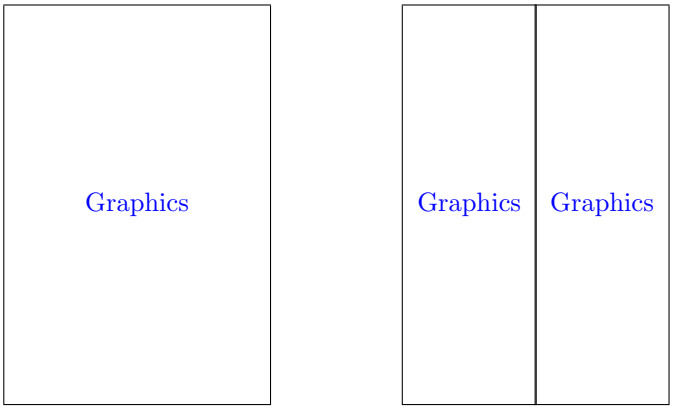

Figure 6: Indicator, Variant C_3

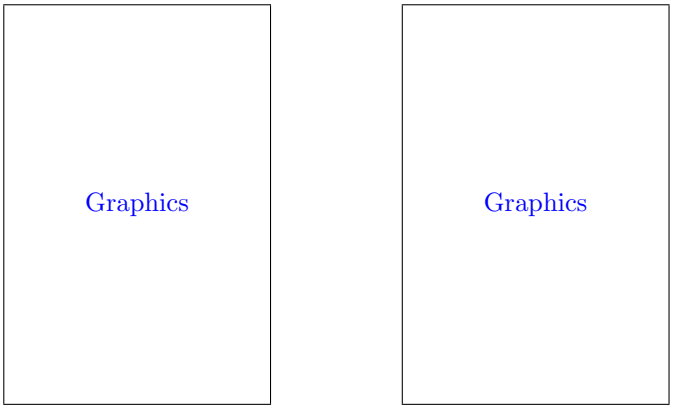

Figure 7: Indicator, Variant C_4

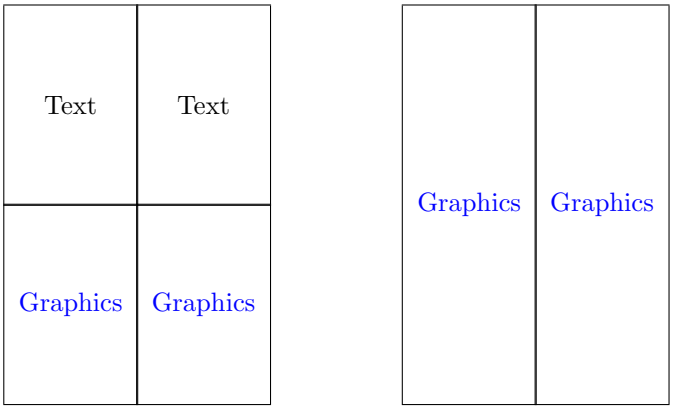

Figure 8: Indicator, Variant D_1

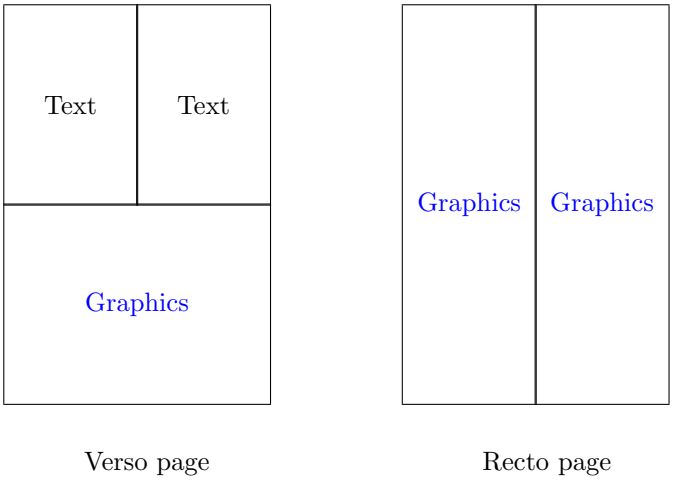

Figure 9: Indicator, Variant D_2

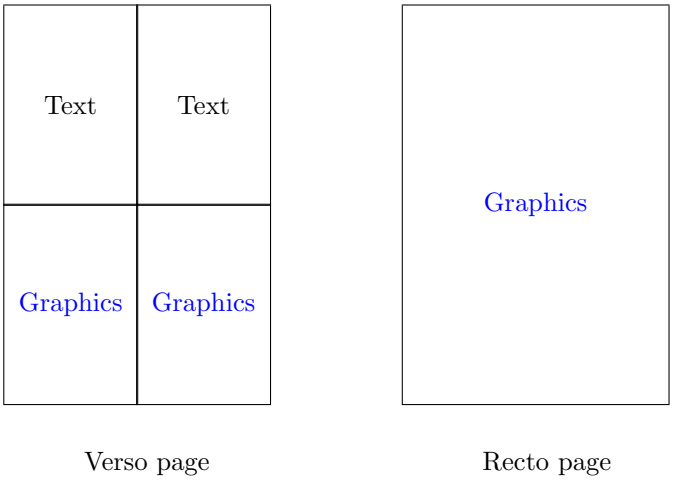

Figure 10: Indicator, Variant D_3

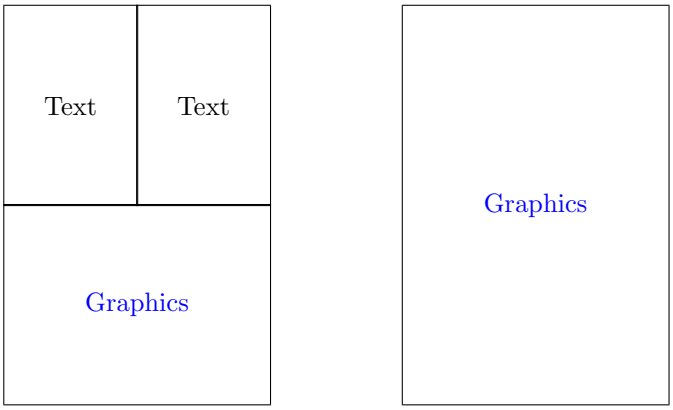

Figure 11: Indicator, Variant D_4

An indicator always starts at a verso (even-numbered) page. Most of the indicators run for a spread (even-odd pair of pages). There are several variants of indicator pages as shown on Figures 1–7. In variant A (Figure 1) there are two columns: text and graphics, running on each page of the indicator. This design continues as needed until both text and illustrations are exhausted. In variant B (Figures 2 and 3) text is present only on the first page of the indicator. The next pagee of the indicator have only illustrative material: either in two column mode (Figure 2) or in one column mode (Figure 3). In variant C (Figures 4–7) there is no text part at all: graphics takes all pages, either in one column or in two column mode. In variant D the verso page is divided horizontally into text part and graphics part. The graphics part on the page can have one large graphics or two smaller, as well as the the graphics part on the recto page can be one or two column wide (see Figures 8–11).

`indicatorA`
`indicatorB`
`indicatorC`
`indicatorD`

The implementation of all these variants is done with the corresponding LaTeX environments: `indicatorA`, `indicatorB` and `indicatorC`. The switch from two column and one column layout in the *graphics* part in variants B and C is done with the usual LaTeX commands `\onecolumn` and `\twocolumn`. Note that these commands in the *text* part may lead to strange results. Similarly, you should not use them in any part of variant A indicator.

`indicatorText`
`indicatorGraphics`

The text and graphics parts are typeset using the environments `indicatorText` and `indicatorGraphics`. The first environment should always precede the second one; of course, there is not text part in variant C.

For example, the layout in variant A is achieved in the following way:

```
\begin{indicatorA}
  \begin{indicatorText}
```

```
    Several pages of text material
  \end{indicatorText}
  \begin{indicatorGraphics}
    Several pages of graphics material
  \end{indicatorGraphics}
\end{indicatorA}
```

while the layout in variant B_2 is achieved in the following way:

```
\begin{indicatorB}
  \begin{indicatorText}
    A page of text material
  \end{indicatorText}
  \begin{indicatorGraphics}
    First page of illustrations
    \onecolumn
    Second page of illustrations
  \end{indicatorGraphics}
\end{indicatorB}
```

and the layout in variant C_4 in the follwing way:

```
\begin{indicatorC}
  \begin{indicatorGraphics}
    \onecolumn
    Several pages of illustrations
  \end{indicatorGraphics}
\end{indicatorC}
```

Up to now we discussed only the indicators that occupy a spread—two consecqutive pages. What happens if we have material for more than one spread? The answer is simple in variants A and C: the patterns there can continue for an unlimited number of pages. The variant B is special: we have a mixed text-graphics page, then a graphics-only page. Should the third page be graphics only or a mixed one? In our interface it can be both. Namely, if the graphics material continues for several pages, they will be graphics-only pages. However, the pair `indicatorText-indicatorGraphics` can be repeated, and in this case we start another spread like the ones on figures 2 and 3.

In the following example we have one mixed page and three graphics page:

```
\begin{indicatorB}
  \begin{indicatorText}
    A page of text material
  \end{indicatorText}
  \begin{indicatorGraphics}
    Three pages of illustrations
  \end{indicatorGraphics}
\end{indicatorB}
```

while in the following example we have a mixed page, a page of graphics material, a mixed page and another graphics page:

```
\begin{indicatorB}
  \begin{indicatorText}
    A page of text material
  \end{indicatorText}
  \begin{indicatorGraphics}
    Three columns of illustrations
  \end{indicatorGraphics}
  \begin{indicatorText}
    A page of text material
  \end{indicatorText}
  \begin{indicatorGraphics}
    Three columns of illustrations
  \end{indicatorGraphics}
\end{indicatorB}
```

indicator
indicatorPage

Actually `indicatorB` is so important, that aliases `indicator` and `indicatorPage` can be used for it:

```
\begin{indicator}
  \begin{indicatorText}
    A page of text material
  \end{indicatorText}
  \begin{indicatorGraphics}
    Three columns of illustrations
  \end{indicatorGraphics}
  \begin{indicatorText}
    A page of text material
  \end{indicatorText}
  \begin{indicatorGraphics}
    Three columns of illustrations
  \end{indicatorGraphics}
\end{indicator}
```

It should be said that such interface decision involves a fair amount of visual formatting. The user must decide how to distribute the material between the pages. A more LaTeXish solution would be to let TeX decide where to make the page breaks. It *is* possible to do, but it is more difficult and it seems that the book design envisions visual formatting anyway. Therefore we, in beamer-like manner, leave the distribution of material between the frames to the author.

2.12 Special Graphics Layouts

While the combination of `\onecolumn` and `\twocolumn` can produce any layout of plots, there are several predefined layouts that help to produce the most common ones. They are described in this section.

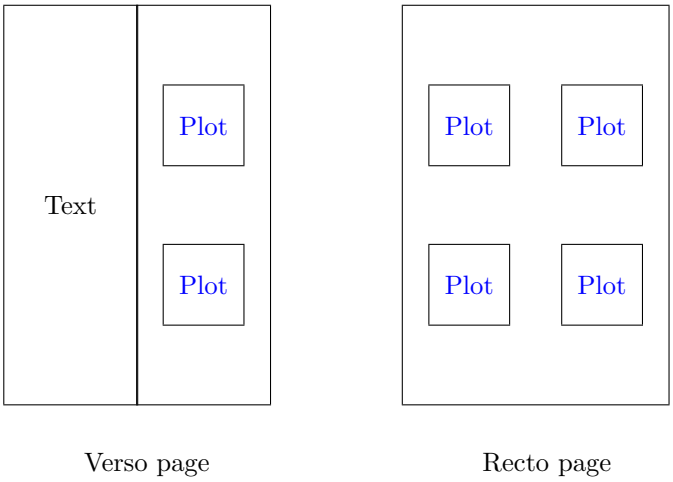

Figure 12: 2 + 4 Layout

`TwoPlusFourPlots` The 2+4 layout (Figure 12) combines two plots on the left page of the indicator spread and four plots on the right page:

```
\begin{indicator}
  \begin{indicatorText}
    Text Part
  \end{indicatorText}
  \begin{indicatorGraphics}
    \begin{TwoPlusFourPlots}
      \begin{chart}
        First chart
      \end{chart}
      \begin{chart}
        Second chart
      \end{chart}
      ....
      \begin{chart}
        Last chart
      \end{chart}
    \end{TwoPlusFourPlots}
  \end{indicatorGraphics}
\end{indicator}
```

`TwoPlusTwoPlots` Similarly `TwoPlusTwoPlots` has two one-column plots on the left and two two-column plots on the right (Figure 13).

`OnePlusTwoPlots` A tall plot can be accomodated in `OnePlusTwoPlots` layout (Figure 14).

Last, it is possible to put three plots on a recto page as shown of Figures 15

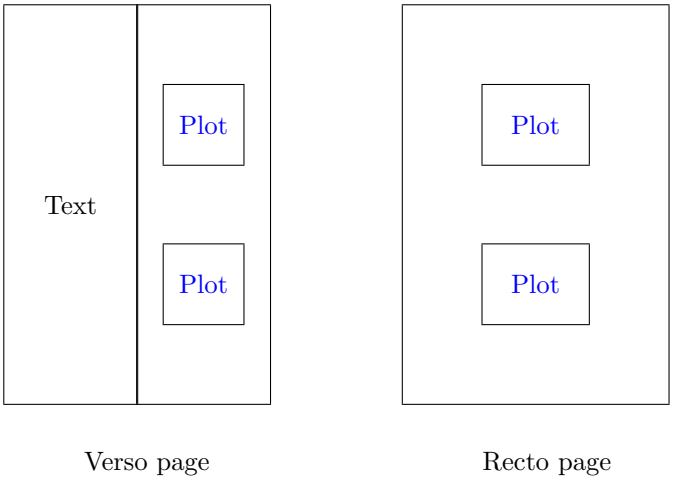

Figure 13: $2 + 2$ Layout

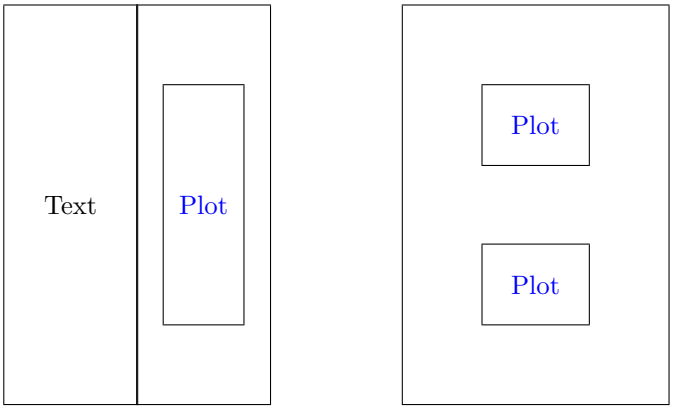

Figure 14: $1 + 2$ Layout

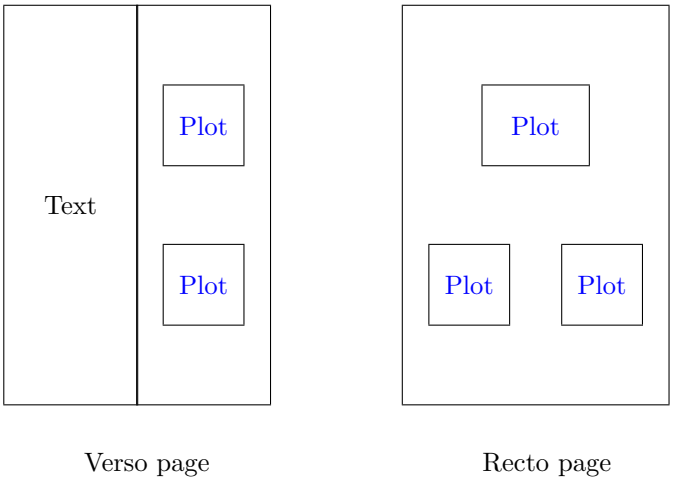

Figure 15: $2 + 3$ Layout, Variant A

and 16. For this one can use `TwoPlusTwoPlots` combined with `multicols` environment as in the following example:

```
\begin{TwoPlusTwoPlots}
  \begin{chart}
    First chart on the verso page
  \end{chart}

  \begin{chart}
    Second chart on the verso page
  \end{chart}

  \begin{chart}
    First chart on the recto page
  \end{chart}

  \begin{multicols}{2}
    \begin{chart}
      Second chart on the recto page
    \end{chart}

    \begin{chart}
      Third chart on the recto page
    \end{chart}
  \end{multicols}
\end{TwoPlusTwoPlots}
```

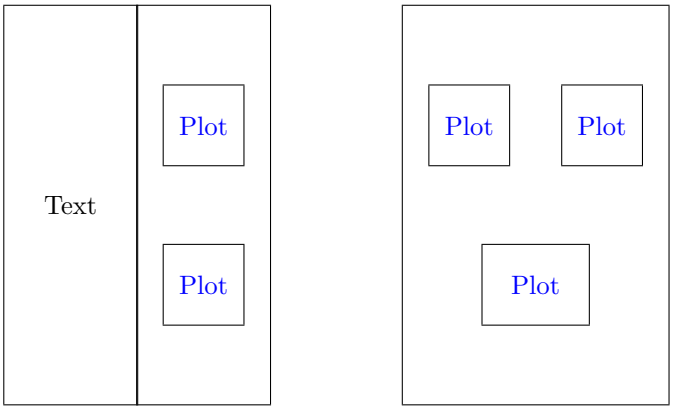

Figure 16: $2 + 3$ Layout, Variant B

2.13 Publication Descriptions

publication

FAO yearboook describes some FAO publications. These publications should be put inside the environment `publication`. The environment has one mandatory argument, which is the title of the publication, and one optional argument, which sets the file name of the publication cover. Note that the option argument, if present, must precede the mandatory one. If this argument is absent, no cover is included. Inside the environment the macros `\pDescription{`⟨*description*⟩`}`, `\pEdition{`⟨*year*⟩`}{`⟨*edition*⟩`}`, `\pWeb{`⟨*URL*⟩`}` and `\pCycle{`⟨*date*⟩`}` are used to typeset the corresponding items related to the publication. For example,

\pDescription
\pEdition
\pCycle
pWeb

```
\begin{publication}[./Plots/StateOfFoodAndAgriculture.png]{The State
    of Food and Agriculture}
  \pDescription{The State of Food and Agriculture, FAO's major
    annual flagship publication, aims at bringing to a wider
    audience balanced science-based assessments of important issues
    in the field of food and agriculture.  Each edition of the
    report contains a comprehensive, yet easily accessible, overview
    of a selected topic of major relevance for rural and
    agricultural development and for global food security.  This is
    supplemented by a synthetic overview of the current global
    agricultural situation.}
 \pEdition{2010}{Livestock in the balance}
 \pEdition{2011}{Women in Agriculture Closing the gender gap for
   development}
 \pCycle{May each year}
 \pWeb{http://www.fao.org/docrep/013/i2050e/i2050e00.htm}
\end{publication}
```

Note that, as in the example, some fields may be repeated.

key publicationparskip *key* publicationskip

Two spacing parameters can be used for typesetting of publications: `publicationskip` is the amount of additional space between the publications, while `publicationparskip` is the space between the paragraphs inside the publication environment. The default values correspond to the command

```
\faoset{publicationskip=6pt plus 2pt minus 2pt,
        publicationparskip=6pt plus 6pt minus 4pt}
```

2.14 Metadata

MetadataCollection metadata

Each chart, map of table in the book has a *source*. Soruces are collected in the environemnt `MetadataCollection`, which consists of separate `metadata` environments. Each `metadata` environment has two obligatory arguments—the name of the source and the key. The key is used to identify the metadata in the charts, maps, tables and other objects. The environment may include other commands.

\source `\source{`⟨***source***⟩`}` sets the source of the data.

\owner `\owner{`⟨***owner***⟩`}` sets the owner of the data.

Note that there is no "description" command because any text which is not an argument of the commands above is considered to belong to the description of the data.

Example of the usage of these commands:

```
\begin{MetadataCollection}
\begin{metadata}{Agricultural population}{P1.DEM.FAO.POP.AGR}

   Agricultural population is defined as all persons depending for
   their livelihood on agriculture, hunting, fishing and forestry.
   It comprises all persons economically active in agriculture as
   well as their non-working dependents. It is not necessary that
   this referred population exclusively come from rural population.

   \source{FILL ME}
   \owner{FILL ME}
\end{metadata}
\end{MetadataCollection}
```

\refMetadata The metadata is referenced by the command `\refMetdata{`⟨*key*⟩`}`, for example

```
\refMetadata{P1.DEM.FAO.POP.AGR}
```

This command will be typset as

Source: Agricultural population, page NNNN.

This command must *not* occur in the caption of the chart, map or table.

Note that the package automatically provides backreferencing: all charts, maps and tables where the medatada is referenced, are mentioned in the corresponding metadata section.

key `metadataInLists`

The sources of each chart, map or table can be shown in the lists of charts, tables, maps or not. The key `metadataInLists` (by default `false`) determines whether they are shown there. To make them visible, put before the lists

```
\faosetup{metadataInLists=true}
```

2.15 Concepts and Methods

`ConceptsAndMethods`

The environment `ConceptsAndMethods` starts a new section "Concepts and Methods". Concepts and methods are collected in the series of `concept` environments. Each environment has one obligatory field: the name of the concept, for example:

```
\begin{ConceptsAndMethods}
  \begin{concept}{Gross domestic product}
    Gross domestic product (GDP) is the market value of all officially
    recognized final goods and services produced within a country in a
    given period of time.
  \end{concept}
  \begin{concept}{Gross state product}
    Gross state product (GSP), or gross regional product (GRP), is a
    measurement of the economic output of a state or province (i.e.,
    of a subnational entity). It is the sum of all value added by
    industries within the state and serves as a counterpart to the
    gross domestic product (GDP).
  \end{concept}
\end{ConceptsAndMethods}
```

2.16 Further Reading

`freading`

The special environment `freading` is used for the "further reading" sections of the book. It starts the text from the new page and changes some defaults.

2.17 Subscripts in Text

`\textsubscript`

The standard LaTeX defines `\textsuperscript`. The class adds a similar `\textsubscript` command.

3 Implementation

3.1 Identification

We start with the declaration who we are. Most `.dtx` files put driver code in a separate driver file `.drv`. We roll this code into the main file, and use the pseudo-guard `<gobble>` for it.

```
⟨class⟩\NeedsTeXFormat{LaTeX2e}
⟨*gobble⟩
\ProvidesFile{faoyearbook.dtx}
⟨/gobble⟩
⟨class⟩\ProvidesClass{faoyearbook}
[2013/03/23 v1.11 Typesetting FAO Yearbook]
```

And the driver code:

```
⟨*gobble⟩
\documentclass{ltxdoc}
\usepackage{array,pict2e,graphpap,xcolor,amsmath}
\usepackage{url,amsfonts,hypdoc}
\usepackage{hyperref}
\PageIndex
\CodelineIndex
\RecordChanges
\EnableCrossrefs
\begin{document}
  \DocInput{faoyearbook.dtx}
\end{document}
⟨/gobble⟩
⟨*class⟩
```

3.2 Options

`\if@factbook` Fact book requires some special treatment, so we check whether we are actually in a factbook:

```
\newif\if@factbook
\@ifclassloaded{faofactbook}{\@factbooktrue}{\@factbookfalse}

\if@factbook\else
\PassOptionsToPackage{a4paper,twoside}{geometry}
\fi
```

`\faoyearbook@size@warning` The font-changing options are not used in our setup, so we just produce a warning:

```
\long\def\faoyearbook@size@warning#1{%
  \ClassWarning{faoyearbook}{Size-changing option #1 will not be
    honored}}%
\DeclareOption{8pt}{\faoyearbook@size@warning{\CurrentOption}}%
\DeclareOption{9pt}{\faoyearbook@size@warning{\CurrentOption}}%
\DeclareOption{10pt}{\faoyearbook@size@warning{\CurrentOption}}%
\DeclareOption{11pt}{\faoyearbook@size@warning{\CurrentOption}}%
\DeclareOption{12pt}{\faoyearbook@size@warning{\CurrentOption}}%
```

\ifprint We have a flag shich shows whether we are in Web or print mode

```
\newif\ifprint
\printfalse
\DeclareOption{web}{\printfalse}
\DeclareOption{print}{\printtrue
  \if@factbook
  \PassOptionsToPackage{papersize={12cm,21cm},layoutsize={10cm,19cm},
    layouthoffset=1cm,layoutvoffset=1cm,twoside}{geometry}%
  \else
  \PassOptionsToPackage{papersize={230mm,317mm},layout=a4paper,
    layouthoffset=1cm,layoutvoffset=1cm,twoside}{geometry}%
   \fi}
```

\if@altMargins Alternative print margins

```
\newif\if@altMargins
\@altMarginsfalse
\DeclareOption{altMargins}{\printtrue\@altMarginstrue
  \PassOptionsToPackage{papersize={220mm,307mm},layout=a4paper,
    layouthoffset=5mm,layoutvoffset=5mm,twoside}{geometry}
    \setlength{\footskip}{10pt}}
```

\if@altMarginsNarrow Yet another pair of alternative print margins

```
\newif\if@altMarginsNarrow
\@altMarginsNarrowfalse
\DeclareOption{altMarginsNarrow}{\printtrue\@altMarginsNarrowtrue
  \PassOptionsToPackage{papersize={220mm,307mm},layout=a4paper,
    layouthoffset=5mm,layoutvoffset=5mm,twoside}{geometry}
    \setlength{\footskip}{10pt}}
```

\ifDraft If we are in 'Draft' or 'draft mode', we print a word 'draft' across the page:

```
\newif\ifDraft
\Draftfalse
\DeclareOption{Draft}{\Drafttrue}
\DeclareOption{draft}{\Drafttrue}
```

\if@issuumode Whether we need issuu-style links

```
\newif\if@issuumode
\@issuumodefalse
\DeclareOption{issuu}{\@issuumodetrue}
```

All other options are just sent to the main class:

```
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{report}}
\ProcessOptions\relax
```

3.3 Loading Class and Packages

We start with the base class and some packages

```
\if@factbook
\LoadClass[10pt,twoside,twocolumn,a4paper]{report}
\else
\LoadClass[10pt,twoside,twocolumn]{report}
\fi
\RequirePackage{graphicx,xkeyval}
\RequirePackage[table,cmyk]{xcolor}
\RequirePackage{tikz,geometry,dcolumn}
\RequirePackage{fancyhdr,pdfcolparcolumns,pdfcolparallel}
\RequirePackage{float,caption,lscape,longtable,siunitx,booktabs}
\RequirePackage{multicol,atbegshi,picture,hhline,afterpage}
\RequirePackage[T1]{fontenc}
\RequirePackage[utf8x]{inputenc}
\RequirePackage{pdfpages}
\RequirePackage[authoryear]{natbib}
\RequirePackage[breaklinks]{hyperref}
\RequirePackage{bookmark}
\if@issuumode
\RequirePackage{issuulinks}
\fi
```

Options for the hyperef package are set as follows:

```
\ifprint
\hypersetup{breaklinks,colorlinks=false,pdfborder=0 0 0,
  pdfauthor={FAO},
  pdfsubject={Statistical Yearbook of the Food And Agricultural Organization for the United Na
  pdftitle={Statistical Yearbook of the Food And Agricultural Organization for the United Nati
  pdfkeywords={FAO, Food Security, Undernourishment, Sustainable agriculture},
  pdfpagelayout=TwoColumnLeft,
  pdfnewwindow=true
}
\else
\hypersetup{breaklinks,colorlinks=false,pdfborder=0 0 0,
  pdfauthor={FAO},
  pdfsubject={Statistical Yearbook of the Food And Agricultural Organization for the United Na
  pdftitle={Statistical Yearbook of the Food And Agricultural Organization for the United Nati
  pdfkeywords={FAO, Food Security, Undernourishment, Sustainable agriculture},
  pdfpagelayout=TwoColumnRight,
  pdfnewwindow=true
}
\fi
```

3.4 Color

We need to tell the printer that we are using CMYK color model. The following is taken from the `pdfx` package (the package itself is not too easy to make work).

```
\def\@pctchar{\expandafter\@gobble\string\%}
\def\@bchar{\expandafter\@gobble\string\\}
\immediate\pdfobj stream attr{/N 4}  file{FOGRA39L.icc}
\edef\OBJ@CVR{\the\pdflastobj}
```

```
\pdfcatalog{/OutputIntents [ <<
  /Type/OutputIntent
  /S/GTS_PDFX
  /OutputCondition (FOGRA39)
  /OutputConditionIdentifier (FOGRA39 \@bchar(ISO Coated v2
   300\@pctchar\space \@bchar(ECI\@bchar)\@bchar))
  /DestOutputProfile \OBJ@CVR\space 0 R
  /RegistryName(http://www.color.org)
 >> ]}
```

3.5 Key-Value Interface

`\faoset` We define the family `fao` for our keys:

```
\def\faoset#1{\setkeys{fao}{#1}}
```

3.6 Fonts

We use arev for mathematics:

```
\RequirePackage{arevmath}
```

For body text we use PT Sans:

```
\def\PTSans@scale{0.95}
\def\PTSansNarrow@scale{0.95}
\def\PTSansCaption@scale{0.95}
\renewcommand{\sfdefault}{PTSans-TLF}
\renewcommand{\familydefault}{\sfdefault}
\renewcommand{\bfdefault}{b}
```

`\narrowfamily` We declare a new family, `\captionfamily`:

```
\DeclareRobustCommand\narrowfamily{\fontfamily{PTSansNarrow-TLF}\selectfont}
```

`\textnarrow` And the matching `\textnarrow` command:

```
\DeclareTextFontCommand{\textnarrow}{\narrowfamily}
```

`\captionfamily` Same with `\captionfamily`:

```
\DeclareRobustCommand\captionfamily{\fontfamily{PTSansCaption-TLF}\selectfont}
```

`\textcaption` And the matching `\textcaption` command:

```
\DeclareTextFontCommand{\textcaption}{\captionfamily}
```

`\normalsize` The basic size is 9.6pt:

```
\renewcommand\normalsize{%
   \@setfontsize\normalsize{9.6pt}{\@xiipt}%
   \abovedisplayskip 10\p@ \@plus2\p@ \@minus5\p@
   \abovedisplayshortskip \z@ \@plus3\p@
   \belowdisplayshortskip 6\p@ \@plus3\p@ \@minus3\p@
   \belowdisplayskip \abovedisplayskip
   \let\@listi\@listI}
\normalsize
```

\small This is the small size:

```
\renewcommand\small{%
   \@setfontsize\small\@ixpt{10}%
   \abovedisplayskip 8.5\p@ \@plus3\p@ \@minus4\p@
   \abovedisplayshortskip \z@ \@plus2\p@
   \belowdisplayshortskip 4\p@ \@plus2\p@ \@minus2\p@
   \def\@listi{\leftmargin\leftmargini
               \topsep 4\p@ \@plus2\p@ \@minus2\p@
               \parsep 2\p@ \@plus\p@ \@minus\p@
               \itemsep \parsep}%
   \belowdisplayskip \abovedisplayskip}
```

We use `rm` style of URL:

```
\urlstyle{sf}
```

3.7 Margins and Paragraphing

We use a4paper. Note that we want to move this definition to the beginning of the document to catch the options:

```
\if@factbook
\AtBeginDocument{%
  \geometry{layoutsize={10cm,19cm},
  left=1cm,right=1cm,bottom=1cm,top=1cm,twoside}%
  \setlength\columnsep{10pt}%
\savegeometry{standard}\@twosidetrue}
\else\if@altMargins
\AtBeginDocument{\geometry{layout=a4paper,
  left=2cm,right=1.5cm,bottom=2cm,top=2cm,twoside}%
  \setlength\columnsep{50pt}%
\savegeometry{standard}\@twosidetrue}
\else\if@altMarginsNarrow
\AtBeginDocument{\geometry{layout=a4paper,
  left=1.5cm,right=1cm,bottom=2cm,top=2cm,twoside}%
  \setlength\columnsep{50pt}%
\savegeometry{standard}\@twosidetrue}
\else
\AtBeginDocument{\geometry{layout=a4paper,
  left=2cm,right=2cm,bottom=2cm,top=2cm,twoside}%
  \setlength\columnsep{50pt}%
\savegeometry{standard}\@twosidetrue}
\fi\fi\fi
```

\parindent \parskip We use not indented paragraphs with paragraph borders given by skips

```
\setlength\parindent\z@
\setlength\parskip{6\p@ plus 6\p@ minus 4\p@}
```

3.8 Cropmarks

There are several packages that provide crop marks. Unfortunately they do not work for us because they put crop marks at the background. Since we have colored pages, we want crop marks to be on the foreground.

In this section we re-implement cropmarks of the geometry package, putting the marks on the foreground.

We postpone the code to the beginning of the document to get the proper value of the switch

```
\AtBeginDocument{\ifprint
  \AtBeginShipout{%
    \AtBeginShipoutUpperLeftForeground{%
      \color{black}%
      \@tempdima=\Gm@layouthoffset
      \@tempdimb=\Gm@layoutvoffset
      \put(\@tempdima,-\@tempdimb+6\p@){\line(0,1){50}}%
      \put(\@tempdima-6\p@,-\@tempdimb){\line(-1,0){50}}%
      \advance\@tempdima by \Gm@layoutwidth
      \put(\@tempdima,-\@tempdimb+6\p@){\line(0,1){50}}%
      \put(\@tempdima+6\p@,-\@tempdimb){\line(1,0){50}}%
      \advance\@tempdimb by \Gm@layoutheight
      \put(\@tempdima,-\@tempdimb-6\p@){\line(0,-1){50}}%
      \put(\@tempdima+6\p@,-\@tempdimb){\line(1,0){50}}%
      \advance\@tempdima by -\Gm@layoutwidth
      \put(\@tempdima-6\p@,-\@tempdimb){\line(-1,0){50}}%
      \put(\@tempdima,-\@tempdimb-6\p@){\line(0,-1){50}}%
    }}\fi}
```

In draft mode we put the word 'DRAFT' across the page:

```
\AtBeginDocument{\ifDraft
  \AtBeginShipout{%
    \AtBeginShipoutUpperLeftForeground{%
      \color{black}%
      \@tempdima=\Gm@layouthoffset
      \@tempdimb=\Gm@layoutvoffset
      \advance\@tempdima by 0.2\Gm@layoutwidth
      \advance\@tempdimb by 0.7\Gm@layoutheight
      \put(\@tempdima,-\@tempdimb){%
        \rotatebox{45}{%
          \fontsize{5cm}{5cm}\selectfont
          \tikz\node[opacity=0.25,inner sep=\z@]{DRAFT};}}}}\fi}
```

3.9 Page Styles

\fao@partblobtop
\fao@partblobbottom

Some pages have "part blobs": colored blobs on the specific positions of the page. These macros set the top and the bottom of the blob corresponding to the part set in the second parameter:

```
\def\fao@partblobtop#1#2{\expandafter\gdef\csname fao@blobstart#1\endcsname{#2}}
\def\fao@partblobbottom#1#2{\expandafter\gdef\csname fao@blobend#1\endcsname{#2}}
```

\fao@blobposition
\fao@blobheight
These lengths keep the current position and length of the colored blob:

```
\newlength{\fao@blobposition}
\newlength{\fao@blobheight}
```

\fao@oddstrip
This is the strip on odd pages

```
\def\fao@oddstrip{%
  \if@altMargins
    \begin{picture}(0,0)%
      \put(530,-219){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@bgcolor}\rule{0.6cm}{1.017\Gm@layoutheight}}}%
    \end{picture}%
  \else\if@altMarginsNarrow
    \begin{picture}(0,0)%
      \put(544,-219){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@bgcolor}\rule{0.6cm}{1.017\Gm@layoutheight}}}%
    \end{picture}%
  \else
    \begin{picture}(0,0)%
      \put(530,-219){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@bgcolor}\rule{0.6cm}{1.017\Gm@layoutheight}}}%
    \end{picture}%
  \fi\fi}
```

\fao@evenstrip
This is the strip on even pages

```
\def\fao@evenstrip{%
  \if@altMargins
    \begin{picture}(0,0)%
      \put(-49,-219){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@bgcolor}\rule{0.6cm}{1.017\Gm@layoutheight}}}%
     \end{picture}%
   \else\if@altMarginsNarrow
     \begin{picture}(0,0)%
       \put(-35,-219){\raisebox{-0.717\Gm@layoutheight}{%
           \color{@bgcolor}\rule{0.6cm}{1.017\Gm@layoutheight}}}%
     \end{picture}%
    \else
      \begin{picture}(0,0)%
        \put(-63,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor}\rule{0.6cm}{1.017\Gm@layoutheight}}}%
      \end{picture}%
  \fi\fi}
```

standardpagestyle
The pagestyle for "normal" pages

```
\fancypagestyle{standardpagestyle}{%
  \fancyhf{}%
  \fancyhfoffset{\z@}%
  \fancyhead[LO]{\fao@oddstrip}
  \fancyhead[LE]{\fao@evenstrip}%
  \fancyhead[RO]{%
```

```
    \color{black}\leftmark}%
  \fancyfoot[RO,LE]{\color{black}\thepage}%
  \renewcommand{\headrulewidth}{\z@}%
  \renewcommand{\footrulewidth}{\z@}}
```

`partpagestyle` Special page style for parts:

```
\fancypagestyle{partpagestyle}{%
  \fancyhf{}%
  \fancyhfoffset{\z@}%
  \fancyhead[LO]{%
    \begin{picture}(0,0)%
      \put(-216,-174){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@bgcolor!10}\rule{0.295\Gm@layoutwidth}{1.017\Gm@layoutheight}%
          \color{@bgcolor!7}\rule{0.728\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
    \end{picture}}%
  \fancyhead[LE]{%
    \begin{picture}(0,0)%
      \put(-63,-174){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@bgcolor!7}\rule{0.728\Gm@layoutwidth}{1.017\Gm@layoutheight}%
          \color{@bgcolor!10}\rule{0.295\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
    \end{picture}}
  \renewcommand{\headrulewidth}{\z@}%
  \renewcommand{\footrulewidth}{\z@}}
```

`indicatorfirstpagestyle` The style for the first indicator page (or all pages for `indicatorA`):

```
\fancypagestyle{indicatorfirstpagestyle}{%
  \fancyhf{}%
    \if@altMargins
    \fancyhead[LE]{%
      \begin{picture}(0,0)
        \put(-105,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{white}\rule{0.6\Gm@layoutwidth}{1.017\Gm@layoutheight}%
            \color{@bgcolor!7}\rule{0.515\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@evenstrip}%
    \fancyhead[LO]{%
      \begin{picture}(0,0)
        \put(-63,75){\raisebox{-0.717\Gm@layoutheight}{%
            \color{white}\rule{0.6\Gm@layoutwidth}{1.017\Gm@layoutheight}%
            \color{@bgcolor!7}\rule{0.515\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@oddstrip}%
    \else\if@altMarginsNarrow
    \fancyhead[LE]{%
      \begin{picture}(0,0)
        \put(-91,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{white}\rule{0.6\Gm@layoutwidth}{1.017\Gm@layoutheight}%
            \color{@bgcolor!7}\rule{0.515\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@evenstrip}%
    \fancyhead[LO]{%
      \begin{picture}(0,0)
        \put(-63,75){\raisebox{-0.717\Gm@layoutheight}{%
```

```
            \color{white}\rule{0.6\Gm@layoutwidth}{1.017\Gm@layoutheight}%
            \color{@bgcolor!7}\rule{0.515\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@oddstrip}%
    \else
    \fancyhead[LE]{%
      \begin{picture}(0,0)
        \put(-120,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{white}\rule{0.6\Gm@layoutwidth}{1.017\Gm@layoutheight}%
            \color{@bgcolor!7}\rule{0.515\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@evenstrip}%
    \fancyhead[LO]{%
      \begin{picture}(0,0)
        \put(-63,75){\raisebox{-0.717\Gm@layoutheight}{%
            \color{white}\rule{0.6\Gm@layoutwidth}{1.017\Gm@layoutheight}%
            \color{@bgcolor!7}\rule{0.515\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@oddstrip}%
  \fi\fi
  \fancyhead[RO]{\color{black}\leftmark}%
  \renewcommand{\headrulewidth}{\z@}%
  \renewcommand{\footrulewidth}{\z@}%
  \fancyfoot[RO]{\color{black}\thepage}%
  \fancyfoot[LE]{\color{black}\thepage}}
```

`indicatorpagestyle` The style for the indicator pages:

```
\fancypagestyle{indicatorpagestyle}{%
  \fancyhf{}%
  \fancyhfoffset{\z@}
  \if@altMargins
    \fancyhead[LE]{%
      \begin{picture}(0,0)%
        \put(-63,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@evenstrip}%
    \fancyhead[LO]{%
      \begin{picture}(0,0)%
        \put(-63,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@oddstrip}%
  \else\if@altMarginsNarrow
    \fancyhead[LE]{%
      \begin{picture}(0,0)%
        \put(-49,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@evenstrip}%
    \fancyhead[LO]{%
      \begin{picture}(0,0)%
        \put(-49,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@oddstrip}%
  \else
```

```
    \fancyhead[LE]{%
      \begin{picture}(0,0)%
        \put(-63,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@evenstrip}%
    \fancyhead[LO]{%
      \begin{picture}(0,0)%
        \put(-63,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
      \end{picture}\fao@oddstrip}%
  \fi\fi
  \fancyhead[RO]{\color{black}\leftmark}%
  \renewcommand{\headrulewidth}{\z@}%
  \renewcommand{\footrulewidth}{\z@}%
  \fancyfoot[RO,LE]{\color{black}\thepage}}
```

`indicatorDpagestyle` This is the style for the first page of indicatorD. The page is divided vertically instead of horizontally.

```
\fancypagestyle{indicatorDpagestyle}{%
  \fancyhf{}%
  \fancyhfoffset{\z@}
  \if@altMargins
    \fancyhead[LE]{%
      \begin{picture}(0,0)%
        \put(-49,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{0.5\Gm@layoutheight}}}%
      \end{picture}\fao@evenstrip}%
    \fancyhead[LO]{%
      \begin{picture}(0,0)%
        \put(-63,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{0.5\Gm@layoutheight}}}%
      \end{picture}\fao@oddstrip}%
  \else\if@altMarginsNarrow
    \fancyhead[LE]{%
      \begin{picture}(0,0)%
        \put(-35,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{0.5\Gm@layoutheight}}}%
      \end{picture}\fao@evenstrip}%
    \fancyhead[LO]{%
      \begin{picture}(0,0)%
        \put(-49,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{0.5\Gm@layoutheight}}}%
      \end{picture}\fao@oddstrip}%
  \else
    \fancyhead[LE]{%
      \begin{picture}(0,0)%
        \put(-63,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{0.5\Gm@layoutheight}}}%
      \end{picture}\fao@evenstrip}%
    \fancyhead[LO]{%
```

```
      \begin{picture}(0,0)%
        \put(-63,-219){\raisebox{-0.717\Gm@layoutheight}{%
            \color{@bgcolor!7}\rule{1.02\Gm@layoutwidth}{0.5\Gm@layoutheight}}}%
      \end{picture}\fao@oddstrip}%
  \fi\fi
  \fancyhead[RO]{\color{black}\leftmark}%
  \renewcommand{\headrulewidth}{\z@}%
  \renewcommand{\footrulewidth}{\z@}%
  \fancyfoot[RO]{\color{black}\thepage}%
  \fancyfoot[LE]{\color{black}\thepage}}
```

`tablepage` Pages for tables

```
\fancypagestyle{tablepage}{%
  \fancyhf{}%
  \fancyhfoffset{\z@}
  \if@altMargins
  \fancyhead[LE]{%
    \begin{picture}(0,0)%
      \put(-48,-255){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@tablebg!10}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
    \end{picture}}%
  \fancyhead[LO]{%
    \begin{picture}(0,0)%
      \put(-62,-255){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@tablebg!10}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
    \end{picture}}%
  \else\if@altMarginsNarrow
  \fancyhead[LE]{%
    \begin{picture}(0,0)%
      \put(-34,-255){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@tablebg!10}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
    \end{picture}}%
  \fancyhead[LO]{%
    \begin{picture}(0,0)%
      \put(-48,-255){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@tablebg!10}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
    \end{picture}}%
  \else
  \fancyhead[LE]{%
    \begin{picture}(0,0)%
      \put(-20,-260){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@tablebg!10}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
    \end{picture}}%
  \fancyhead[LO]{%
    \begin{picture}(0,0)%
      \put(-20,-260){\raisebox{-0.717\Gm@layoutheight}{%
          \color{@tablebg!10}\rule{1.02\Gm@layoutwidth}{1.017\Gm@layoutheight}}}%
    \end{picture}}%
  \setlength{\footskip}{15pt}%
  \fi\fi
```

```
  \renewcommand{\headrulewidth}{\z@}%
  \renewcommand{\footrulewidth}{\z@}%
  \fancyfoot[RO,LE]{\thepage}}
```

`conceptspagestyle` The pagestyle for "concept" pages

```
\fancypagestyle{conceptpagestyle}{%
  \fancyhf{}%
  \fancyhfoffset{\z@}%
  \fancyhead[LE]{\color{black}}%
  \fancyhead[RO]{%
    \color{black}\leftmark}%
  \fancyfoot[RO,LE]{\color{black}\thepage}%
  \renewcommand{\headrulewidth}{\z@}%
  \renewcommand{\footrulewidth}{\z@}}
```

3.10 Sectioning

`\if@mainmatter` This is used to check whether we are at main matter

```
\newif\if@mainmatter
```

`\frontmatter` We want Roman numbers for front matter:

```
\def\frontmatter{\cleardoublepage
  \pagenumbering{roman}\onecolumn\@mainmatterfalse}
```

`\mainmatter` We want Arabic numbers for main matter:

```
\def\mainmatter{\cleardoublepage\pagenumbering{arabic}\twocolumn
  \@mainmattertrue}
```

`\tocdepth` Only sections and up are allowed in TOC:

```
\setcounter{tocdepth}{1}
```

`\secnumdepth` Only the parts are numbered in out setup:

```
\setcounter{secnumdepth}{-1}
```

`\thepart` And the parts are numbered using Arabic numbers:

```
\renewcommand \thepart {\@arabic\c@part}
```

`\c@fao@partnum` To draw the blobs in part color in the proper position, we need to associate them with parts. However, some parts are numbered, some are note. The macro `\fao@partnum` keeps the current part number counted continuously from the beginning to end.

```
\newcounter{fao@partnum}
\setcounter{fao@partnum}{0}
```

`\fao@currentpartnum` The current value of `\fao@partnum` used in TOC:

```
\def\fao@currentpartnum{0}
```

\part This is basicale the same as in [19], with the change that we want to use our own page style. it.

```
\renewcommand\part{%
  \cleardoublepage
  \addtocontents{toc}{\string\colorlet{@bgcolor}[cmyk]{\fao@color@string}}%
  \stepcounter{fao@partnum}%
```

Now we set \fao@blobposition and \fao@blobheight to the correct values—if defined

```
  \expandafter\ifx\csname fao@blobstart\thefao@partnum\endcsname\relax
     \setlength{\fao@blobposition}{-100em}%
  \else
     \setlength{\fao@blobposition}{\csname
       fao@blobstart\thefao@partnum\endcsname}%
  \fi
  \typeout{DEBUG: position \the\fao@blobposition}%
  \setlength{\fao@blobheight}{-\fao@blobposition}%
  \expandafter\ifx\csname fao@blobend\thefao@partnum\endcsname\relax
     \setlength{\fao@blobheight}{0em}%
  \else
     \addtolength{\fao@blobheight}{\csname
       fao@blobend\thefao@partnum\endcsname}%
  \fi
  \typeout{DEBUG: height \the\fao@blobheight}%
  \addtocontents{toc}{%
    \string\gdef\string\fao@currentpartnum{\thefao@partnum}}%
  \ifprint
    \if@altMargins
     \newgeometry{layout=a4paper,left=0.5\textwidth,right=2cm,
      layouthoffset=5mm,layoutvoffset=5mm}%
    \else\if@altMarginsNarrow
     \newgeometry{layout=a4paper,left=0.5\textwidth,right=2cm,
      layouthoffset=5mm,layoutvoffset=5mm}%
    \else
     \newgeometry{layout=a4paper,left=0.5\textwidth,right=2cm,
      layouthoffset=1cm,layoutvoffset=1cm}%
    \fi\fi
  \else
  \newgeometry{layout=a4paper,left=0.5\textwidth,right=2cm}%
  \fi
  \selectcolor
\rowcolors{2}{@bgcolor!10}{}%
  \pagestyle{partpagestyle}%
  \if@twocolumn
    \onecolumn
  \fi
  \secdef\@part\@spart}
```

\@part This is the actual part making macro.

```
\def\@part[#1]#2{%
```

```
      \refstepcounter{part}%
      \addcontentsline{toc}{part}{\thepart\hspace{1em}#1}%
    \markboth{\MakeUppercase{#1}}{\MakeUppercase{\partname}~\thepart}%
    {\interlinepenalty \@M
     \normalfont\null\hspace{-70mm}%
     \fontsize{24\p@}{32\p@}\selectfont \bfseries
     \raisebox{25\p@}{\color{@bgcolor!70}\MakeUppercase{\partname}}%
     \hspace{35\p@}%
     \raisebox{-180\p@}{\fontsize{250\p@}{200\p@}\selectfont \bfseries
       \color{@bgcolor}\begin{tikzpicture}%
         \node[opacity=0.25,inner sep=\z@]{\thepart};%
       \end{tikzpicture}}%
     \parbox[t]{0.8\textwidth}{\raggedright
       \color{@bgcolor}#2}\par}\vspace{100\p@}\bgroup
 \large}
```

`\@spart` This produces *unnumbered* parts:

```
\def\@spart#1{%
     \phantomsection
      \addcontentsline{toc}{spart}{#1}%
    \markboth{#1}{#1}%
    {\interlinepenalty \@M
     \normalfont\null\hspace{-70mm}%
     \fontsize{24\p@}{32\p@}\selectfont \bfseries
     \raisebox{25\p@}{\color{@bgcolor!70}\phantom{\MakeUppercase{\partname}}}%
     \hspace{35\p@}%
     \raisebox{-180\p@}{\fontsize{250\p@}{200\p@}\selectfont \bfseries
       \color{@bgcolor!20}\begin{tikzpicture}%
         \node[opacity=0.25,inner sep=\z@]{\phantom{\thepart}};%
       \end{tikzpicture}}%
     \parbox[t]{0.8\textwidth}{\raggedright
       \color{@bgcolor}#1}\par}\vspace{100\p@}\bgroup
 \large}
```

`\EndPartIntro` This command switches the special formatting of part pages back:

```
\def\EndPartIntro{%
  \egroup\twocolumn
  \pagecolor{white}\color{black}%
  \loadgeometry{standard}%
  \pagestyle{standardpagestyle}}
```

`\chaptermark` We use chapter mark in headers

```
\renewcommand{\chaptermark}[1]{\markright{#1}}
```

`\sectionmark` We do not use section info in headers:

```
\renewcommand{\sectionmark}[1]{}
```

`\section` We changed the standard `\ection` defaults:

```
\renewcommand\section{\@startsection {section}{1}{\z@}%
```

```
                                    {\z@ plus 6\p@}%
                                    {\parskip}%
                                    {\normalfont\large\bfseries}}
```

`\subsection` We changed the standard \ubsection defaults:

```
\renewcommand\subsection{\@startsection {subsection}{2}{\z@}%
                                    {\z@ plus 6\p@}%
                                    {\parskip}%
                                    {\normalfont\large\itshape}}
```

3.11 Setting Colors

`\fao@color@string` This is the command that remembers the present color for TOC

```
\def\fao@color@string{0,0,0}
```

`@bgcolor@next` We store the next background color in `@bgcolor@next`. We store the next heading background in `@tableheadcolor@next`.

`\setbgcolor` The command `\setbgcolor` selects the next background color:

```
\def\setbgcolor#1{\colorlet{@bgcolor@next}[cmyk]{#1}%
  \addtocontents{toc}{\string\colorlet{@bgcolor}[cmyk]{#1}}%
  \gdef\fao@color@string{#1}}
\setbgcolor{red}
```

The key-value interface for the same command:

```
\define@key{fao}{bgcolor}{\setbgcolor{#1}}
```

And for separate setting of `@tableheadcolor`

```
\define@key{fao}{tableheadcolor}{\colorlet{@tableheadcolor}[cmyk]{#1}}
```

`@bgcolor` The current color is in the macro `@bgcolor`.

`@tableheadcolor@next` `\selectcolor` This command makes the actual color change:

```
\def\selectcolor{\colorlet{@bgcolor}{@bgcolor@next}%
  \colorlet{@tableheadcolor}{@bgcolor!30}}
\selectcolor
```

`@tablebg` The color for table pages

```
\define@key{fao}{tablebg}{\colorlet{@tablebg}[cmyk]{#1}}
```

3.12 Indicator Pages

`\indicatorText` Normally (outside of indicator) this should produce an error message:

```
\newenvironment{indicatorText}{\PackageError{faoyearbook}{%
  Environment `indicatorText' in a wrong place!}{%
This environment must be inside indicator pages.}}{}
```

`\indicatorGraphics` Normally (outside of indicator) this should produce an error message:

```
\newenvironment{indicatorGraphics}{\PackageError{faoyearbook}{%
  Environment `indicatorGraphics' in a wrong place!}{%
This environment must be inside indicator pages.}}{}
```

`\indicatorA` Version A. We use parallel processing. Most of the code of the here follows [8] with the patches from [9].

```
\def\indicatorA{%
```

First, we switch to the next recto page::

```
  \clearpage\ifodd\c@page\hbox{}\newpage\fi
```

Next, we need to change to one column and remember whether to switch back to one column mode. We also now are ready for the new page style

```
  \if@twocolumn
    \onecolumn
    \@tempswatrue
  \else
    \@tempswafalse
  \fi
  \pagestyle{indicatorfirstpagestyle}%
```

Now we switch on `Parallel` environment. We want to redefine the command `\ParallelParOnePage` (more on this below), so we do this inside a group to keep the change private:

```
  \bgroup
  \def\ParallelParOnePage{\FAO@ParallelParOnePage}%
  \begin{Parallel}{0.45\textwidth}{0.45\textwidth}%
```

We define `\indicatorText` and `\indicatorGraphics` inside the `\indicator` so they are local to the group.

```
\def\indicatorText{\ParallelLText\bgroup\columnwidth=\ParallelLWidth
  \strut}%
\def\endindicatorText{\egroup}
\def\indicatorGraphics{\ParallelRText\bgroup\strut
  \columnwidth=\ParallelRWidth}%
\def\endindicatorGraphics{\egroup}}%
```

`\endindicatorA` Wrapping up

```
\def\endindicatorA{\end{Parallel}\egroup
```

Switching to two column if needed:

```
    \if@tempswa\twocolumn\else\onecolumn\fi
    \clearpage\color{black}%
    \global\let\default@color\current@color\normalcolor
    \loadgeometry{standard}%
    \pagestyle{standardpagestyle}}
```

`\FAO@ParallelParOnePage` The package parallel tries to achieve aligned lines in two columns. The way it does it is the following: it takes one line from each column and puts them into boxes, then again and again. Which is fine, but we explicitly do *not* want this

effect, otherwise the lines in the text part look really funny. So we redefine the routine `\ParallelParOnePage` to use larger boxes. We also need the patches from pdfcolparallel to keep the color stack.

So we change the lines `\vsplit BOX to\dp\strutbox` to `\vsplit BOX to\@tempdima` where `\@tempdima` is the page height minus one basline. We also use `\vtop` instead of `\vbox` to achieve the proper alignment.

```
\def\FAO@ParallelParOnePage{%
  \ifnum\ParallelBoolVar=\@ne
    \par
    \begingroup
      \leftmargin=\z@
      \rightmargin=\z@
      \vbadness=10000 %
      \vfuzz=3ex %
      \splittopskip=\z@skip
      \@tempdima=\@colroom
      \advance\@tempdima by -\baselineskip
      \loop
        \ifnum\ParallelBoolVar=\@ne
          \noindent
          \hbox to\textwidth{%
            \hskip\ParallelLeftMargin
            \hbox to\ParallelTextWidth{%
              \ifvoid\ParallelLBox
                \hskip\ParallelLWidth
              \else
                \pcp@SetCurrent{Left}%
                \ParallelWhichBox=\z@
                \vtop{%
                  \setbox\ParallelBoxVar
                      =\vsplit\ParallelLBox to\@tempdima
                  \unvbox\ParallelBoxVar
                }%
              \fi
              \ifnum\ParallelBoolMid=\@ne
                \hskip\ParallelMainMidSkip
                \begingroup
                  \pcp@RuleBetweenColor
                  \vrule
                \endgroup
              \else
                \hss
              \fi
              \hss
              \ifvoid\ParallelRBox
                \hskip\ParallelRWidth
              \else
                \pcp@SetCurrent{Right}%
                \ParallelWhichBox=\@ne
```

```
                \vtop{%
                  \setbox\ParallelBoxVar
                      =\vsplit\ParallelRBox to\@tempdima
                      \vfill
                  \unvbox\ParallelBoxVar
                }%
              \fi
            }%
          }%
          \ifvoid\ParallelLBox
            \ifvoid\ParallelRBox
              \global\ParallelBoolVar=\z@
            \fi
          \fi%
        \fi%
      \ifnum\ParallelBoolVar=\@ne
        \penalty\interlinepenalty
      \repeat
      \par
    \endgroup
    \pcp@SetCurrent{}%
  \fi}
```

`\indicatorB` Here we have the first page with text and graphics, but consecutive pages have graphics only. This follows `\indicatorA`:

```
\def\indicatorB{%
  \clearpage\ifodd\c@page\hbox{}\clearpage\fi
```

Next, we need to change to two column and remember whether to switch back. We also now are ready for the new page style

```
  \if@twocolumn
    \@tempswatrue
  \else
    \twocolumn
    \@tempswafalse
  \fi
  \pagestyle{indicatorpagestyle}%
```

Then we define `indicatorText` and `indicatorGraphics`. Note that `indicatorText` starts a spread and introduces special page style:

```
  \bgroup
  \def\indicatorText{\bgroup\twocolumn
    \clearpage\ifodd\c@page\hbox{}\clearpage\fi
    \thispagestyle{indicatorfirstpagestyle}}%
  \def\endindicatorText{\newpage\egroup}
  \def\indicatorGraphics{\bgroup}%
  \def\endindicatorGraphics{\egroup}}%
```

`\endindicatorB` Wrapping up `indicatorB`:

```
\def\endindicatorB{\egroup\color{black}%
```

```
  \global\let\default@color\current@color\normalcolor
  \if@tempswa\twocolumn\else\onecolumn\fi
  \clearpage\loadgeometry{standard}%
  \pagestyle{standardpagestyle}}
```

`\indicatorC` This follows `\indicatorB`:

```
\def\indicatorC{%
  \clearpage\ifodd\c@page\hbox{}\clearpage\fi
```

Next, we need to change to two column and remember whether to switch back. We also now are ready for the new page style

```
  \if@twocolumn
    \@tempswatrue
  \else
    \twocolumn
    \@tempswafalse
  \fi
  \pagestyle{indicatorpagestyle}%
```

Then we define `indicatorText` and `indicatorGraphics`:

```
  \bgroup
  \def\indicatorText{\PackageError{faoyearbook}{%
    Bad place for indicatorText}{%
  On indicatorC pages only graphics is allowed}}%
  \def\endindicatorText{\newpage\egroup}
  \def\indicatorGraphics{\bgroup}%
  \def\endindicatorGraphics{\egroup}}%
```

`\endindicatorC` Wrapping up `indicatorC`:

```
\def\endindicatorC{\egroup\color{black}%
  \global\let\default@color\current@color\normalcolor
  \if@tempswa\twocolumn\else\onecolumn\fi
  \clearpage\loadgeometry{standard}%
  \pagestyle{standardpagestyle}}
```

`\indicatorD` This the indicator for horizontally divided graphics:

```
\def\indicatorD{%
  \clearpage\ifodd\c@page\hbox{}\clearpage\fi
```

Next, we need to change to one column and remember whether to switch back. We also now are ready for the new page style

```
  \if@twocolumn
    \@tempswatrue
    \onecolumn
  \else
    \@tempswafalse
  \fi
  \thispagestyle{indicatorDpagestyle}%
  \pagestyle{indicatorpagestyle}%
%   Then we define |indicatorText| and |indicatorGraphics|:
```

```
% \begin{macrocode}
  \bgroup
  \def\indicatorText{\begin{minipage}[t][0.5\textheight]{\textwidth}%
      \begin{multicols}{2}}%
  \def\endindicatorText{\end{multicols}\end{minipage}}
  \def\indicatorGraphics{\bgroup}%
  \def\endindicatorGraphics{\egroup}}%
```

`\endindicatorD` Wrapping up `indicatorD`:

```
\def\endindicatorD{\egroup\color{black}%
  \global\let\default@color\current@color\normalcolor
  \if@tempswa\twocolumn\else\onecolumn\fi
  \clearpage\loadgeometry{standard}%
  \pagestyle{standardpagestyle}}
```

`\indicator` An alias for `indicatorB` is just `indicator`:

```
\let\indicator\indicatorB
\let\endindicator\endindicatorB
```

`\indicatorPage` Another alias for `indicatorB` is `indicatorPage`:

```
\let\indicatorPage\indicatorB
\let\endindicatorPage\endindicatorB
```

3.13 Floats

`\float@caption` Package float defines `\float@caption`, and hyperref tries to redefine it, which sometimes fails. Since we use caption anyway, we can safely disable this macro:

```
\let\float@caption\@undefined
```

Since we intend to use babel in the future, we do not hard code the names for the floats. Here we provide the English variants.

`\listchartname` List of charts:

```
\newcommand\listchartname{List of Charts}
```

`\listmapname` List of maps:

```
\newcommand\listmapname{List of Maps}
```

`\chartname` A chart:

```
\newcommand\chartname{Chart}
```

`\minitabname` A chart:

```
\newcommand\minitabname{Mini Table}
```

`\mapname` A map:

```
\newcommand\mapname{Map}
```

We introduce two new floats:

```
\newfloat{chart}{H}{loc}
\floatname{chart}{\chartname}
\newfloat{minitab}{H}{loc}
\floatname{minitab}{\minitabname}
\newfloat{map}{H}{lom}
\floatname{map}{\mapname}
```

`\listofcharts` The list of charts:

```
\def\listofcharts{\listof{chart}{\protect\hypertarget{list:chart}{\listchartname}}}
```

`\listofmaps` The list of maps:

```
\def\listofmaps{\listof{map}{\protect\hypertarget{list:map}{\listmapname}}}
```

`\listoftables` The list of tables:

```
\def\listoftables{\listof{table}{\protect\hypertarget{list:table}{\listtablename}}}
```

`\@floatwidth` The width of the floatbox:

```
\define@key{fao}{floatwidth}{\def\@floatwidth{#1}}
\faoset{floatwidth=\columnwidth}
```

`\@floatheight` The height of the floatbox:

```
\define@key{fao}{floatheight}{\def\@floatheight{#1}}
\faoset{floatheight=0.45\textheight}
```

`\ifKV@fao@fixedfloatheight` Whether to use fixed float height

```
\define@boolkey{fao}{fixedfloatheight}{}
\faoset{fixedfloatheight=false}
```

`\@beginfloat@hook` This is done at the beginning of a float:

```
\def\@beginfloat@hook{}
```

`\@endfloat@hook` This is done at the beginning of a float:

```
\def\@endfloat@hook{}
```

`\@nonfloat` The main environment for setting up the the nonfloats. The argument is the type of the float

```
\def\@nonfloat#1{\@beginfloat@hook
  \ifKV@fao@fixedfloatheight
    \begin{minipage}[t][\@floatheight]{\@floatwidth}\begin{list}{}{%
          \topsep=\z@\partopsep=\z@\leftmargin=\z@\rightmargin=\z@}%
        \def\@captype{#1}\item
        \global\let\default@color\current@color\normalcolor
  \else
    \begin{minipage}{\@floatwidth}\begin{list}{}{%
          \topsep=\z@\partopsep=\z@\leftmargin=\z@\rightmargin=\z@}%
        \def\@captype{#1}\item
        \global\let\default@color\current@color\normalcolor
  \fi}
```

`\end@nonfloat` And ending the environment:

```
\def\end@nonfloat{\end{list}\end{minipage}%
\ifKV@fao@fixedfloatheight\else\vspace{\belowdisplayskip}\fi
\@endfloat@hook}
```

This environment makes it easy to redefine other environments:

`table` Table:

```
\renewenvironment{table}{\begin{@nonfloat}{table}}{\end{@nonfloat}}
```

`chart` Chart:

```
\renewenvironment{chart}{\begin{@nonfloat}{chart}}{\end{@nonfloat}}
```

`minitab` Chart:

```
\renewenvironment{minitab}{\begin{@nonfloat}{minitab}}{\end{@nonfloat}}
```

`map` Map:

```
\renewenvironment{map}{\begin{@nonfloat}{map}}{\end{@nonfloat}}
```

Now we set up captions. All our captions come *before* the graphics material. Note that due to the strange way caption works this means that `\aboveskip` is actually the skip *below* the caption. Go figure.

```
\DeclareCaptionLabelFormat{uppercase}{\hyperlink{list:\@captype}{\MakeUppercase{#1}~#2}}
\captionsetup{position=top,labelformat=uppercase,format=plain,
  textfont=bf,justification=centering,
  singlelinecheck=false}
\captionsetup{aboveskip=\z@}
\AtEndCaption{\par}
```

`\ifKV@fao@fixedcaptionheight` Sometimes we typeset the caption in a box:

```
\define@boolkey{fao}{fixedcaptionheight}{%
  \ifKV@fao@fixedcaptionheight
    \def\caption@parbox{\parbox[t][\@fao@captionheight]}%
  \else
    \def\caption@parbox{\parbox[t]}%
  \fi}
\faoset{fixedfloatheight=false}
```

`\@fao@captionheight` The height of the caption in fixed caption height mode:

```
\define@key{fao}{captionheight}{\def\@fao@captionheight{#1}}
\faoset{captionheight=2\baselineskip}
```

`\footnotebar` A rule used mostly in footnotes:

```
\def\footnotebar{\par\vskip\skip\@mpfootins\footnoterule\footnotesize}
```

We define new column types for table headers:

```
\newcolumntype{d}[1]{D{.}{.}{#1}}
\newcolumntype{H}{>{\columncolor{@tableheadcolor}[1.01\tabcolsep][1.01\tabcolsep]}c}
```

`P` columntype is much more complex. Basically we want a centered entry with a parbox of the given width inside.:

```
\newcolumntype{P}[1]{>{\columncolor{@tableheadcolor}[1.01\tabcolsep][1.01\tabcolsep]%
    \@fao@Pentry{#1}}c<{\end@fao@Pentry}}
```

`\@fao@Pentry` Since `\parbox` needs "real" braces to delimit the argument, we use this trick. Note `\hspace{0pt}` to allow TEX to hyphenate the first word.

```
\def\@fao@Pentry#1#2\end@fao@Pentry{%
\parbox[t]{#1}{\centering\strut\hspace{\z@}#2\strut}}
```

Same with `C` entry:

```
\newcolumntype{C}[1]{>{\columncolor{@tableheadcolor}[1.01\tabcolsep][1.01\tabcolsep]%
    \@fao@Centry{#1}}c<{\end@fao@Centry}}
```

`\@fao@Centry` This macro is similar to `\@fao@Pentry`, but with different way to set the width of the `\parbox`:

```
\def\@fao@Centry#1#2\end@fao@Centry{%
\settowidth{\@tempdima}{$-99.999$}%
\@tempdima=#1\@tempdima\relax
\parbox[t]{\@tempdima}{\centering\strut\hspace{\z@}#2\strut}}
```

`\tablepages` The special style for table pages:

```
\def\tablepages{\clearpage\bgroup\normalcolor
  \ifprint
    \if@altMargins
    \newgeometry{layout=a4paper,
      layouthoffset=5mm,layoutvoffset=5mm,
      left=2cm,right=1.5cm,bottom=1.7cm,top=.7cm,twoside}%
    \else\if@altMarginsNarrow
    \newgeometry{layout=a4paper,
      layouthoffset=5mm,layoutvoffset=5mm,
      left=1.5cm,right=1cm,bottom=1.7cm,top=.7cm,twoside}%
    \else
    \newgeometry{layout=a4paper,
      layouthoffset=1cm,layoutvoffset=1cm,
      left=0.2in,right=0.2in,top=0.2in,bottom=0.25in}%
    \fi\fi
  \else
  \newgeometry{layout=a4paper,
    left=0.2in,right=0.2in,top=0.2in,bottom=0.25in}%
  \fi
  \pagestyle{tablepage}\onecolumn}
```

`\endtablepages` And reverting back:

```
\def\endtablepages{\egroup\loadgeometry{standard}%
  \pagestyle{standardpagestyle}\clearpage}
```

To prevent the ugly white spacese between the rules and the text we put the separations in booktabs to zero. To prevent the crammed tables we increase \arraystretch:

```
\aboverulesep\z@
\belowrulesep\z@
\setlength\minrowclearance{\z@}
\def\arraystretch{1.5}
```

\arrayrulewidth This is the thickness of \hhline

```
\setlength{\arrayrulewidth}{0.9pt}
```

\vline We use \vline for interrupting \hhline only.

```
\def\vline{\vrule \@width 7\p@ \relax}
```

3.14 Special Plot Layouts

\TwoPlusFourPlots This is simple: we make fixed plot heights:

```
\def\TwoPlusFourPlots{\bgroup
  \faoset{fixedfloatheight=true,floatheight=0.45\textheight}%
  \faoset{fixedcaptionheight=true,captionheight=2\baselineskip}}%
\def\endTwoPlusFourPlots{\pagebreak\egroup}
```

\@fao@plotnum The counter to keep the current plot number

```
\newcount\@fao@plotnum
```

\TwoPlusTwoPlots This is more tricky than 2+4: we need to keep the current plot number and switch to one column when needed.

```
\def\TwoPlusTwoPlots{\bgroup
  \faoset{fixedfloatheight=true,floatheight=0.45\textheight}%
  \faoset{fixedcaptionheight=true,captionheight=2\baselineskip}%
  \global\@fao@plotnum=0\relax
  \def\@beginfloat@hook{\global\advance\@fao@plotnum by 1\relax
    \ifnum\@fao@plotnum=3\relax\onecolumn\fi}}
```

\endTwoPlusTwoPlots . . . and switch back:

```
\def\endTwoPlusTwoPlots{\egroup\twocolumn}
```

\OnePlusTwoPlots Here we want one tall float and two wide floats:

```
\def\OnePlusTwoPlots{\bgroup
  \faoset{fixedfloatheight=true,floatheight=0.95\textheight}%
  \faoset{fixedcaptionheight=true,captionheight=2\baselineskip}%
  \global\@fao@plotnum=0\relax
  \def\@beginfloat@hook{\global\advance\@fao@plotnum by 1\relax
    \ifnum\@fao@plotnum=2\relax\onecolumn\fi
    \ifnum\@fao@plotnum>1\faoset{floatheight=0.45\textheight}\fi}}
```

\endOnePlusTwoPlots . . . and switch back:

```
\def\endOnePlusTwoPlots{\egroup\twocolumn}
```

3.15 Patching graphicx Package

Package graphicx provides a useful keyword `viewport` which allows to show just a part of an image. However, you need to put there the actual coordinates of the viewport window. For large maps we would like to have relative coordinates as fractions of natural width size. Here we define the new option `rviewport` for Relative Viewport. It works like this. Suppose the image has the bounding box x_{ll}, y_{ll}, x_{ur}, y_{ur}. We give four numbers ξ_{ll}, η_{ll}, ξ_{ur}, η_{ur}, and the viewport coordinates become

$$\begin{aligned} x'_{ll} &= x_{ll} + \xi_{ll}\Delta_x \\ y'_{ll} &= y_{ll} + \eta_{ll}\Delta_y \\ x'_{ur} &= x_{ll} + \xi_{ur}\Delta_x \\ y'_{ur} &= y_{ll} + \eta_{ur}\Delta_y \end{aligned}$$

where

$$\begin{aligned} \Delta_x &= x_{ur} - x_{ll} \\ \Delta_y &= y_{ur} - y_{ll} \end{aligned}$$

This means that the left half of the image is `rviewport = 0 0 0.5 1`, and the right half is `rviewport = 0.5 0 1 1`.

We follow [3].

```
\define@key{Gin}{rviewport}
           {\let\Gin@viewport@code\Gin@rviewport\Gread@parse@rvp#1 \\}
```

`\Gread@parse@rvp` We parse four numbers into the corresponding macros. Note that their names are significant: `pdftex.def` would not clip the image if `\Gin@vllx` is not defined.

```
\def\Gread@parse@rvp#1 #2 #3 #4 #5\\{%
  \def\Gin@vllx{#1}%
  \def\Gin@vlly{#2}%
  \def\Gin@vurx{#3}%
  \def\Gin@vury{#4}%
}%
```

`\Gin@rviewport` And the viewport code. Note that `pdftex.def` relies on the values of `\Gin@v...` macros, so we redefine them as well.

```
\def\Gin@rviewport{%
  \let\Gin@ollx\Gin@llx
  \let\Gin@olly\Gin@lly
  \let\Gin@ourx\Gin@urx
  \let\Gin@oury\Gin@ury
  \Gin@nat@width\Gin@urx\p@
  \advance\Gin@nat@width-\Gin@llx\p@
  \Gin@nat@height\Gin@ury\p@
  \advance\Gin@nat@height-\Gin@lly\p@
  \dimen@\Gin@vurx\Gin@nat@width
```

```
                        \edef\Gin@vurx{\strip@pt\dimen@}%
  \advance\dimen@\Gin@llx\p@
                        \edef\Gin@urx{\strip@pt\dimen@}%
  \dimen@\Gin@vury\Gin@nat@height
                        \edef\Gin@vury{\strip@pt\dimen@}%
  \advance\dimen@\Gin@lly\p@
                        \edef\Gin@ury{\strip@pt\dimen@}%
  \dimen@\Gin@vllx\Gin@nat@width
                        \edef\Gin@vllx{\strip@pt\dimen@}%
  \advance\dimen@\Gin@llx\p@
                      \edef\Gin@llx{\strip@pt\dimen@}%
  \dimen@\Gin@vlly\Gin@nat@height
                        \edef\Gin@vlly{\strip@pt\dimen@}%
  \advance\dimen@\Gin@lly\p@
                       \edef\Gin@lly{\strip@pt\dimen@}%
}
```

We want all plots to keep the aspect ratio of the originals:

```
\setkeys{Gin}{keepaspectratio=true}
```

`\Gin@getbase` This command selects the file for `\includegraphics`. We patch it to select print or web version if they exist:

```
\def\Gin@getbase#1{%
  \ifprint\def\filename@add{_print}\else\def\filename@add{_web}\fi
  \edef\Gin@tempa{%
    \def\noexpand\@tempa####1#1\space{%
      \def\noexpand\Gin@base{####1}}}%
  \IfFileExists{\filename@area\filename@base\filename@add#1}%
  {\Gin@tempa
    \expandafter\@tempa\@filef@und
    \edef\Gin@ext{#1}}{%
    \IfFileExists{\filename@area\filename@base#1}%
    {\Gin@tempa
      \expandafter\@tempa\@filef@und
      \edef\Gin@ext{#1}}{}}}%
```

3.16 Multipage Maps

`\largeGraphicsNotes` This puts its argument in the storage for the processing:

```
\long\def\largeGraphicsNotes#1{%
  \long\gdef\@largeGraphicsNotes{#1}}
\largeGraphicsNotes{}
```

`\@largegraphics@footnotes` Our `\@nonfloat` is a minipage. Since `\includeLargeGraphics` splits it, we need some place to store footnote captions.

```
\newinsert\@largegraphics@footnotes
```

`\Gin@leftfraction` This key might occur as the argument to `\includeLargeGraphics` or `\includeExtraLargeGraphics`. Since we want to be able to use there "normal" arguments, we put it into the `Gin` family.

```
\define@key{Gin}{leftfraction}{\def\Gin@leftfraction{#1}}
\setkeys{Gin}{leftfraction=0.415}
```

`\Gin@leftpartoffset` The offset for the left part of the includegraphics

```
\newdimen\Gin@leftpartoffset
\define@key{Gin}{leftpartoffset}{\Gin@leftpartoffset=#1\relax}
\setkeys{Gin}{leftpartoffset=-87pt}
```

`\Gin@rightpartoffset` The offset for the right part of the includegraphics

```
\newdimen\Gin@rightpartoffset
\define@key{Gin}{rightpartoffset}{\Gin@rightpartoffset=#1\relax}
\setkeys{Gin}{rightpartoffset=0pt}
```

`\if@extra@large@graphics` This is false inside large graphics, but true inside large graphics.

```
\newif\if@extra@large@graphics
\@extra@large@graphicsfalse
```

`\@includeLargeGraphics` This is the generic macro for `\includeLargeGraphics` or `\includeExtraLargeGraphics`:

```
\def\@includeLargeGraphics{\@ifnextchar[% ] To make Emacs happy
   {\@@includeLargeGraphics}{\@@includeLargeGraphics[]}}
```

`\@@includeLargeGraphics` This is the work horse of the (extra) large graphcis...

```
\def\@@includeLargeGraphics[#1]#2{%
```

First we read the keys:

```
  \setkeys{Gin}{#1}%
```

We need to check whether we are in a float environment and in the proper column mode:

```
  \ifx\@captype\@undefined\PackageError{faoyearbook}{%
    Wrong place for \string\includeLargeGraphics{} or
    \string\includeExtraLargeGraphics}{%
    The commands \string\includeLargeGraphics{} and
    \string\includeExtraLargeGraphics{} should be inside
    map or chart environment}\fi
  \if@extra@large@graphics
    \if@twocolumn\PackageError{faoyearbook}{%
    Wrong place for \string\includeExtraLargeGraphics}{%
    The command \string\includeExtraLargeGraphics{} should be inside
    a one column page}\fi
  \else
    \if@twocolumn\else\PackageError{faoyearbook}{%
    Wrong place for \string\includeLargeGraphics}{%
    The command \string\includeLargeGraphics{} should be inside
    a two column page}\fi
  \fi
```

We save the footnotes so far

```
\ifvoid\@mpfootins\else
  \global\setbox\@largegraphics@footnotes\vbox{\unvbox\@mpfootins}%
\fi
```

Now we close the minipage and remember the vertical offset. We also want to restore two column mode after we close *if* we are not in "extra large graphics":

```
  \end{list}\end{minipage}%
  \if@extra@large@graphics
    \def\end@nonfloat{\end{list}\end{minipage}}%
  \else
    \def\end@nonfloat{\end{list}\end{minipage}\afterpage{\twocolumn}}%
  \fi
  \par
  \@tempdima=-\topskip
  \advance\@tempdima\pagetotal
```

We are ready to print the left part of the image and the notes. Note that we re-read the keys inside the minipage

```
    \ifvoid\@largegraphics@footnotes\else
      \global\setbox\@mpfootins\vbox{%
        \unvbox\@largegraphics@footnotes}%
    \fi
    \begin{minipage}{\@floatwidth}\begin{list}{}{%
          \setkeys{Gin}{#1}%
          \topsep=\z@
          \partopsep=\z@\leftmargin=\z@\rightmargin=\z@}%
          \let\default@color\current@color\normalcolor\item
          \hspace*{\Gin@leftpartoffset}%
          \includegraphics[rviewport=0 0 {\Gin@leftfraction} 1,clip]{#2}\par
          \@largeGraphicsNotes
```

Reset the notes:

```
        \largeGraphicsNotes{}%
  \end{list}\end{minipage}%
```

Now go to the next page and put the right part:

```
    \if@extra@large@graphics\newpage\else\onecolumn\fi
    \vspace*{\@tempdima}%
    \begin{minipage}{\@floatwidth}\begin{list}{}{%
          \setkeys{Gin}{#1}%
          \topsep=\z@
          \partopsep=\z@\leftmargin=\z@\rightmargin=\z@}%
          \let\default@color\current@color\normalcolor\item
          \hspace*{\Gin@rightpartoffset}%
          \includegraphics[rviewport={\Gin@leftfraction} 0 1 1,clip]{#2}}
```

`\includeLargeGraphics` Include three column graphics:

```
\def\includeLargeGraphics{%
  \@extra@large@graphicsfalse
  \setkeys{Gin}{leftfraction=0.415, leftpartoffset=-87\p@}%
```

```
  \@includeLargeGraphics}
```

\includeExtraLargeGraphics Include four column graphics

```
\def\includeExtraLargeGraphics{%
  \@extra@large@graphicstrue
  \setkeys{Gin}{leftfraction=0.5}%
  \@includeLargeGraphics}
```

\graphicKey This typesets everything in a box in the lower right corner

```
\newenvironment{graphicKey}[1][]{\vfill\par\hfill
  \def\@tempa{#1}\ifx\@empty\@tempa\else\begin{minipage}{#1}\fi}{%
    \ifx\@tempa\@empty\else\end{minipage}\fi}
```

3.17 Formatting of Table of Contents and Lists

\tableofcontents Table of contents is formatted in a special way:

```
\renewcommand\tableofcontents{\cleardoublepage
  \makebox[0pt][l]{\fontsize{24pt}{32pt}\selectfont \bfseries
    \color{black!70}\MakeUppercase{\contentsname}\space}%
  \par\vspace{-2\baselineskip}\vspace{-\parskip}%
  \@starttoc{toc}}
```

\@tocpartskip This is the skip between the parts in TOC:

```
\newlength{\@tocpartskip}
\define@key{fao}{tocpartskip}{\setlength{\@tocpartskip}{#1}}
\faoset{tocpartskip=\z@}
```

\@fao@tocrule@start The start of the current TOC colored rule

```
\newdimen\@fao@tocrule@start
```

\@fao@tocrule@height The height of the current TOC rule

```
\newdimen\@fao@tocrule@height
```

\@draw@tocrule@part Drawing the toc rule for a part

```
\def\@draw@tocrule@part{\@fao@tocrule@height=\pagetotal
  \protected@write\@auxout{}{\string\fao@partblobbottom{\fao@currentpartnum}{\the\@fao@tocrule
  \advance\@fao@tocrule@height-\@fao@tocrule@start
  \bgroup\parskip\z@
  \parbox[b][\z@]{\z@}{\hspace*{-15\p@}\color{@bgcolor}\rule{2\p@}{\@fao@tocrule@height}}%
  \parbox[b][\z@]{\z@}{\hspace*{330\p@}%
    \if@altMarginsNarrow\hspace{10mm}\fi
    \color{@bgcolor}\rule{41\p@}{\@fao@tocrule@height}}%
  \par\vspace{-0.5\baselineskip}\egroup}
```

\@draw@tocrule@section Drawing the toc rule for a section

```
\def\@draw@tocrule@section{\@fao@tocrule@height=\pagetotal
  \protected@write\@auxout{}{\string\fao@partblobbottom{\fao@currentpartnum}{\the\@fao@tocrule
  \advance\@fao@tocrule@height-\@fao@tocrule@start
```

```
  \advance\@fao@tocrule@height5\p@
  \bgroup\parskip\z@\small
  \raisebox{\baselineskip}[\z@][\z@]{\parbox[b][\z@]{\z@}{\hspace*{-35\p@}\color{@bgcolor}\rul
  \raisebox{\baselineskip}[\z@][\z@]{\parbox[b][\z@]{\z@}{\hspace*{310\p@}%
    \if@altMarginsNarrow\hspace{10mm}\fi
      \color{@bgcolor}\rule{41\p@}{\@fao@tocrule@height}}}%
  \par\vspace{-\baselineskip}\egroup}
```

`\l@part` This prints the part in TOC:

```
\renewcommand*\l@part[2]{%
  \ifnum \c@tocdepth >-2\relax
    \addpenalty{-\@highpenalty}%
    \setlength\@tempdima{3em}%
    \addvspace{\@tocpartskip}%
    \begingroup
```

We store the current vertical position of the page into `\@fao@tocrule@start`

```
%     \addvspace{-2pc}\par
     \@fao@tocrule@start=\pagetotal
     \protected@write\@auxout{}{\string\fao@partblobtop{\fao@currentpartnum}{\the\@fao@tocrule
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      \leftskip180\p@
      {\leavevmode
       \color{@bgcolor}\bfseries\partname\space#1:
       \hfil \hb@xt@\@pnumwidth{\hss #2}}%
    \par\@draw@tocrule@part
      \nobreak
        \global\@nobreaktrue
        \everypar{\global\@nobreakfalse\everypar{}}%
    \endgroup
  \fi}
```

`\l@spart` This adds unnumbered part to TOC

```
\newcommand*\l@spart[2]{%
  \ifnum \c@tocdepth >-2\relax
    \addpenalty{-\@highpenalty}%
    \setlength\@tempdima{3em}%
    \begingroup
     \@fao@tocrule@start=\pagetotal
     \protected@write\@auxout{}{\string\fao@partblobtop{\fao@currentpartnum}{\the\@fao@tocrule
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      \leftskip180\p@
      {\leavevmode
       \color{@bgcolor}\bfseries#1:
       \hfil \hb@xt@\@pnumwidth{\hss #2}}%
    \par\@draw@tocrule@part
      \nobreak
        \global\@nobreaktrue
```

```
        \everypar{\global\@nobreakfalse\everypar{}}%
    \endgroup
  \fi}
```

`\l@section` This prints the section in TOC:

```
\renewcommand*\l@section[2]{%
  \ifnum \c@tocdepth >-2\relax
    \addpenalty{-\@highpenalty}%
    \setlength\@tempdima{3em}%
    \begingroup
    \small
     \@fao@tocrule@start=\pagetotal
     \leftskip200\p@\relax\parskip\z@
      \parindent \z@ \rightskip \@pnumwidth
      \parfillskip -\@pnumwidth
      {\leavevmode\small\strut
       #1\hfil \hb@xt@\@pnumwidth{\hss #2}}\par\@draw@tocrule@section
       \nobreak
         \global\@nobreaktrue
         \everypar{\global\@nobreakfalse\everypar{}}%
    \endgroup
  \fi}
```

`\appendix` We do not draw colored rules in the TOC part of the appendix:

```
\renewcommand\appendix{%
  \bookmarksetup{startatroot}%
  \addtocontents{toc}{\string\let\string\@draw@tocrule@part\string\relax
    \string\let\string\@draw@tocrule@section\string\relax}}
```

We use special formatting of metadata in the lists of... This requires explicit `\pars` at the end:

```
\AtEndDocument{%
  \immediate\write\@auxout{\string\@writefile{loc}{\string\par}}%
  \immediate\write\@auxout{\string\@writefile{lot}{\string\par}}%
  \immediate\write\@auxout{\string\@writefile{lom}{\string\par}}}
```

3.18 Metadata

`\metadata` This starts the metadata section. The commands inside are local to the metadata.

```
\def\metadata#1#2{\bgroup
  \def\meta@key{#2}%
```

`\source` This typesets the source:

```
    \def\source##1{\par\penalty10000\emph{Source: }##1\par\penalty10000}%
```

`\owner` This typesets the owner:

```
    \def\owner##1{\par\penalty10000\emph{Owner: }##1\par\penalty10000}%
```

```
  \begin{list}{}{\topsep4\p@\labelwidth\z@
      \labelsep\z@\itemindent\z@\parsep0.4ex plus 0.5ex minus
      0.2ex\relax\listparindent\z@\leftmargin\z@\rightmargin\z@
    \partopsep\z@}%
  \NR@gettitle{#1}\phantomsection\label{metadata:#2}%
  \item{\bfseries#1\par\penalty10000}}
```

`\endmetadata` This closes the environment:

```
\def\endmetadata{%
  \expandafter\ifx\csname
     metaback@\meta@key\endcsname\relax
  \else
    \par\penalty10000\emph{Referenced in: }
    \csname metaback@\meta@key\endcsname
     \par\penalty10000
  \fi
  \end{list}\egroup}
```

`\refMetadataname` The name for the metadata reference:

```
\def\refMetadataname{Metadata:}
```

`\refMetadata` The way we actually reference the metadata:

```
\def\refMetadata#1{%
  \ifx\@captype\@undefined\def\@captype{table}\fi
    \if@filesw
      \immediate\write\@mainaux{%
        \string\fao@metaback{#1}{\@captype}{\csname the\@captype\endcsname}{\thepage}{\@curre
    \addtocontents{\csname ext@\@captype\endcsname}{\string\listmetadata{#1}}%
    \fi
%  \refMetadataname~\nameref{metadata:#1}, page~\pageref{metadata:#1}
}
```

`\fao@metaback` This reads the backreferences to metadata and prepares the the list. The arguments are: key, float type, number of float, page and hyperref

```
\def\fao@metaback#1#2#3#4#5{%
  \expandafter\ifx\csname metaback@#1\endcsname\relax
    \expandafter\gdef\csname metaback@#1\endcsname{%
    \hyper@linkstart{link}{#5}#2~#3\hyper@linkend}%
  \else
    \expandafter\g@addto@macro\csname metaback@#1\endcsname{,
    \hyper@linkstart{link}{#5}#2~#3\hyper@linkend}%
  \fi}
```

`\ifKV@fao@metadataInLists` Whether put metadata in lists of...

```
\define@boolkey{fao}{metadataInLists}{}
\faoset{metadataInLists=false}
```

`\listmetadata` The way metadata is presented in the lists:

```
\def\listmetadata#1{\ifKV@fao@metadataInLists
```

```
  \bgroup\small
  \ifvmode\relax
    \leavevmode
    \textit{\refMetadataname}~%
   \else
     \unskip;~%
  \fi
  \nameref{metadata:#1}, page~\pageref{metadata:#1}%
  \egroup\hangafter=0\hangindent=3.8em\rightskip=3.8em\relax
  \fi}
```

`\MetadataCollection` The section with metadata

3.19 Concepts and Methods

`\conceptsname` The name for concepts section

```
\def\conceptsname{Concepts and Methods}
```

`\ConceptsAndMethods` This is the style for the concepts and methods

```
\newenvironment{ConceptsAndMethods}{\clearpage\twocolumn\small
  \loadgeometry{standard}%
  \pagestyle{conceptpagestyle}%
  \section{\conceptsname}}{\clearpage\pagestyle{standardpagestyle}\normalsize}
```

`\concept` This is the environment for concepts:

```
\newenvironment{concept}[1]{\begin{description}\item[#1:]}{\end{description}}
```

3.20 Further Reading

`\fitemize` This is the special version of `itemize` for further reading pages. Basically it is a patched kernel version.

```
\def\fitemize{%
  \ifnum \@itemdepth >\thr@@\@toodeep\else
    \advance\@itemdepth\@ne
    \edef\@itemitem{labelitem\romannumeral\the\@itemdepth}%
    \expandafter
    \list
      \csname\@itemitem\endcsname
      {\def\makelabel##1{\color{@bgcolor}{##1}\space}%
        \itemsep\z@\labelwidth\z@
        \leftmargin\z@\labelsep\z@}%
  \fi}
```

`\endfitemize` This is standard:

```
\let\endfitemize =\endlist
```

`\freading` This is the “Further Reading environment”

```
\newenvironment{freading}{%
```

```
  \vfill\section*{Further reading}\par
  \vspace{-\baselineskip}{\color{@bgcolor}{\rule{\columnwidth}{1.5pt}}}\par
  \vspace{-\baselineskip}\bgroup
  \let\itemize=\fitemize
  \let\enditemize=\endfitemize}{\egroup}
```

3.21 Publication

\@publicationskip Skip between the publications. By default \medskip:

```
\newlength{\@publicationskip}
\define@key{fao}{publicationskip}{\setlength{\@publicationskip}{#1}}
\faoset{publicationskip=6pt plus 2pt minus 2pt}
```

\@publicationparskip Paragraph skip between the publications.

```
\newlength{\@publicationparskip}
\define@key{fao}{publicationparskip}{\setlength{\@publicationparskip}{#1}}
\faoset{publicationparskip=6pt plus 6pt minus 4pt}
```

\publication This typesets one publication:

```
\newenvironment{publication}[2][]{%
  \par{\bfseries#2\par}\begin{minipage}[t]{0.49\columnwidth}%
    \setlength\parskip{\@publicationparskip}%
    \gdef\@pub@cover{#1}%
    \long\def\pDescription##1{\par##1\par}%
    \def\pEdition##1##2{\par##1: ##2\par}%
    \def\pCycle##1{\par Publication cycle: ##1\par}%
    \def\pWeb##1{\par \raggedright Webpage: \url{##1}\par}}%
  {\end{minipage}%
  \ifx\@pub@cover\@empty\else
  \hspace{0.1\columnwidth}%
  \raisebox{\dimexpr\baselineskip-\totalheight}{%
    \includegraphics[width=0.4\columnwidth]{\@pub@cover}}\fi\par
  \vspace{\@publicationskip}}
```

3.22 Subscripts

\textsubscript This follows standard LaTeX:

```
\DeclareRobustCommand*\textsubscript[1]{%
  \@textsubscript{\selectfont#1}}
\def\@textsubscript#1{%
  {\m@th\ensuremath{_{\mbox{\fontsize\sf@size\z@#1}}}}}
```

3.23 LyX code

\lyxlist It seems Lyx wants this:

```
\newenvironment{lyxlist}[1]
{\begin{list}{}
{\settowidth{\labelwidth}{#1}
```

```
 \setlength{\leftmargin}{\labelwidth}
 \addtolength{\leftmargin}{\labelsep}
 \renewcommand{\makelabel}[1]{##1\hfil}}}
{\end{list}}
```

3.24 The final word

```
\twocolumn
```
⟨/class⟩

References

[1] Stephen G. Hartke. *Arev Sans for TEX and LATEX*, May 2006. http://mirrors.ctan.org/fonts/arev.

[2] Pavel Farář. *Support Package for Free Fonts by ParaType*, May 2011. http://mirrors.ctan.org/fonts/paratype.

[3] D. P. Carlisle. *Packages in the 'Graphics' Bundle*, November 2005. http://mirrors.ctan.org/macros/latex/required/graphics.

[4] Uwe Kern. *Extending LATEX's Color Facilities: the xcolor Package*, January 2007. http://mirrors.ctan.org/macros/latex/contrib/xcolor.

[5] Piet van Oostrum. *Page Layout in LATEX*, March 2004. http://mirrors.ctan.org/macros/latex/contrib/fancyhdr.

[6] Till Tantau. *The TikZ and PGF Packages*, February 2008.

[7] Hideo Umeki. *The geometry package*, December 2008. http://mirrors.ctan.org/macros/latex/contrib/geometry.

[8] Matthias Eckermann. *The Parallel Package*, April 2003. http://mirrors.ctan.org/macros/latex/contrib/parallel.

[9] Heiko Oberdiek. *The pdfcolparallel package*, January 2010. http://mirrors.ctan.org/macros/latex/contrib/oberdiek.

[10] Anselm Lingnau. *An Improved Environment for Floats*, November 2008. http://mirrors.ctan.org/macros/latex/contrib/float.

[11] Axel Sommerfeldt. *Typesetting Captions with the caption Package*, February 2007. http://mirrors.ctan.org/macros/latex/contrib/caption.

[12] Simon Fear. *Publication Quality Tables in LATEX*, April 2005. http://mirrors.ctan.org/macros/latex/contrib/booktabs.

[13] David Carlisle. *The longtable Package*, February 2004. http://mirrors.ctan.org/macros/latex/required/tools.

[14] D. P. Carlisle. *The lscape package*, October 2000. http://mirrors.ctan.org/macros/latex/required/graphics.

[15] Hendri Adriaens. *The xkeyval package*, November 2011. http://mirrors.ctan.org/macros/latex/contrib/xekyval.

[16] Frank Mittelbach and David Carlisle. *A new implementation of LATEX's tabular and array environment*, August 2006. http://mirrors.ctan.org/macros/latex/required/tools.

[17] David Carlisle. *The hhline Package*, May 1994. http://mirrors.ctan.org/macros/latex/required/tools.

[18] UK TEX Users Group. UK list of TEX frequently asked questions. http://www.tex.ac.uk/cgi-bin/texfaq2html, 2008.

[19] Leslie Lamport, Frank Mittelbach, and Johannes Braams. *Standard Document Classes for LATEX version 2e*, 1997. http://mirrors.ctan.org/macros/latex/base.

Change History

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; numbers in roman refer to the code lines where the entry is used.

A

B

C

D

E

F

G

H

I

K

L

R

S

T

U