\textbf{\LaTeX{} Packages for Unified Process Methodology}

\textbf{\LaTeX{} Packages for Unified Process Methodology}

Official Documentation

Reference: UPM-2019-01
Version: 24.0
Updated: 2019/09/17
Status: Public

Authors: Stéphane Galland, Frans van Dunné
This document describes the \TeX\ Packages for Unified Process Methodology project.

\TeX\ is a trademark of the American Mathematical Society.

\texttt{tex-upmethodology} is owned by Stéphane Galland, \textit{Arakné.org}, France.

This document was realised with \TeX\ and \texttt{tex-upmethodology}.

Copyright © 2017-2019 Stéphane GALLAND.

This document is published by the Arakné.org Group. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publishers.

Reference : UPM-2019-01
## Document Summary

<table>
<thead>
<tr>
<th>Project</th>
<th>LaTeX Packages for Unified Process Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>Document</td>
<td>Official Documentation</td>
</tr>
<tr>
<td>Reference</td>
<td>UPM-2019-01</td>
</tr>
<tr>
<td>Version</td>
<td>24.0</td>
</tr>
<tr>
<td>Last Update</td>
<td>2019/09/17</td>
</tr>
</tbody>
</table>

## Authors

<table>
<thead>
<tr>
<th>Names</th>
<th>Comments</th>
<th>Emails</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stéphane Galland</td>
<td>Original Author</td>
<td><a href="mailto:galland@arakhne.org">galland@arakhne.org</a></td>
</tr>
<tr>
<td>Frans van Dunné</td>
<td>Reviewer</td>
<td></td>
</tr>
</tbody>
</table>

## Validators

<table>
<thead>
<tr>
<th>Names</th>
<th>Comments</th>
<th>Emails</th>
<th>Initials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stéphane Galland</td>
<td>Original Author</td>
<td><a href="mailto:galland@arakhne.org">galland@arakhne.org</a></td>
<td></td>
</tr>
</tbody>
</table>

## Version History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>23.0</td>
<td>2017/02/17</td>
<td>Replace the package <code>subfigure</code> by <code>subcaption</code>.</td>
</tr>
<tr>
<td>23.1</td>
<td>2017/03/10</td>
<td>Fixing <code>subfigure</code> invalid alignment.</td>
</tr>
<tr>
<td>23.2</td>
<td>2017/08/08</td>
<td>Fixing spelling errors and typos.</td>
</tr>
<tr>
<td>23.3</td>
<td>2017/11/28</td>
<td>Add <code>standardlists</code> option.</td>
</tr>
<tr>
<td>23.4</td>
<td>2019/08/04</td>
<td>Add <code>graphicspathcontext</code> option.</td>
</tr>
<tr>
<td>24.0</td>
<td>2019/09/17</td>
<td>Add class options to include optional packages.</td>
</tr>
</tbody>
</table>
List of Figures
List of Tables
1

INTRODUCTION

This documentation is written for and compiled by the version 20190917 of tex-upmethodology.

This set of packages enables users to write documents according to the Unified Process Methodology. It was initially written by Stéphane Galland from the laboratory “Systèmes et Transports” and is distributed by the Arakhêne website. The provided packages and classes may also be used for other types of documents (reports, theses...). Since 2012, it is used to support the layout and the style for the PhD theses of the Doctoral School SPIM.

Packages are:

- upmethodology-version.sty: makes it possible to set the version and the status of the document. It also makes it possible to manage the document history;
- upmethodology-fmt.sty: provides some useful functions to format the UP documents;
- upmethodology-document.sty: provides functions to manage the project, the subproject and the status of the document;
- upmethodology-frontpage.sty: formats and provides a front page for the document;
- upmethodology-backpage.sty: formats and provides a back page for the document;
- upmethodology-task.sty: is the optional package that provides macros to manage project’s tasks.
- upmethodology-spec.sty: is the optional package that provides macros to build a specification description.
- upmethodology-document.cls: is the class that provides the whole document specification. It is based on book and on the previous packages;
- upmethodology-code.sty: provides optional macros for source code formatting;
- upmethodology-extension.sty: provides macros for extension mechanism.

---


I

GENERAL USER DOCUMENTATION
Download and Installation

This chapter describes where to download tex-upmethodology and how to install it.

2.1/ Download

tex-upmethodology is available on the Arakhne website: [http://www.arakhne.org/tex-upmethodology/](http://www.arakhne.org/tex-upmethodology/)

Different types of installation are available: manual installation, Debian packages.

2.2/ Manual System-wide Installation

To make tex-upmethodology available to all users, copy the content of the tex-upmethodology archive inside one of your system texmf directory, usually one of:

- /usr/share/texmf-texlive/tex/latex/upmethodology.
- /usr/share/texmf/tex/latex/upmethodology.

The second is to rebuild the \LaTeX{} databases by invoking on a console (Unix syntax us used):

```
$> sudo mktexlsr
$> sudo update-updmap --quiet
```

sudo is a standard Linux tool that allows authorized users to temporarily obtain the administration rights.

2.3/ Manual User-wide Installation

To make tex-upmethodology available to one user, copy the content of the tex-upmethodology archive inside the $HOME/texmf directory.

It is not required to rebuild the system-wide \LaTeX{} databases because the user’s texmf are dynamically parsed by the \LaTeX{} distributions.

2.4/ Debian Package Installation

Debian packages are available on Arakhne website: [http://www.arakhne.org/ubuntu.html](http://www.arakhne.org/ubuntu.html) Please follow the given rules.
2.5/ **Package Dependencies**

This section contains the list of all the package dependencies for the upmethodology packages.

2.5.1/ **UPMETHODOLOGY-BACKPAGE.STY**

upmethodology-backpage package depends on:

- upmethodology-extension
- upmethodology-p-common

2.5.2/ **UPMETHODOLOGY-CODE.STY**

upmethodology-code package depends on:

- upmethodology-p-common

2.5.3/ **UPMETHODOLOGY-DOCUMENT.CLS**

upmethodology-document class depends on:

- a4wide
- hyperref
- upmethodology-backpage
- upmethodology-code (optional)
- upmethodology-document
- upmethodology-extension
- upmethodology-frontpage
- upmethodology-p-common
- upmethodology-spec (optionnal)
- upmethodology-task (optionnal)
- url

2.5.4/ **UPMETHODOLOGY-DOCUMENT.STY**

upmethodology-document package depends on:

- babel
- upmethodology-extension
- upmethodology-fmt
- upmethodology-p-common
- upmethodology-version
- vmargin
2.5. PACKAGE DEPENDENCIES

2.5.5/ UPMETHODOLOGY-EXTENSION.STY

upmethodology-extension package depends on:

- upmethodology-p-common

2.5.6/ UPMETHODOLOGY-FMT.STY

upmethodology-fmt package depends on:

- amsmath
- amsthm
- colortbl
- environ
- graphicx
- hyphenat
- mathbb
- multicol
- picinpar
- pifont
- setspace
- subcaption
- tabularx
- thmtools
- txfonts
- upmethodology-p-common
- xkeyval

2.5.7/ UPMETHODOLOGY-FRONTPAGE.STY

upmethodology-frontpage package depends on:

- upmethodology-document
- upmethodology-extension
- upmethodology-p-common

2.5.8/ UPMETHODOLOGY-P-COMMON.STY

upmethodology-p-common package depends on:

- ifthen
- xcolor
- xspace
2.5.9/ upmethodology-spec.sty

depends on:

- ulem
- upmethodology-code
- upmethodology-fmt
- upmethodology-p-common

2.5.10/ upmethodology-task.sty

package depends on:

- upmethodology-p-common
- upmethodology-version

2.5.11/ upmethodology-version.sty

package depends on:

- upmethodology-fmt
- upmethodology-p-common
II

PACKAGE DOCUMENTATION
CLASS UPMETHODOLOGY-DOCUMENT

Version: 2019/09/17

The \texttt{upmethodology-document} class provides the basic configuration for a document. According to an option, this class is able to extend the standard \texttt{book}, \texttt{report} or \texttt{article} \LaTeX{} classes. It also include several of the other \texttt{upmethodology} packages.

3.1/ TYPES OF DOCUMENTS

upmethodology-document supports three particular options, which permit to set the type of document:

- \texttt{book}: A book-specification is a two-sided document composed of parts and chapters, and with a copyright page and document information page. This option indicates to \texttt{upmethodology-document} to load the \LaTeX{} standard \texttt{book} class. In addition the \texttt{\part} and \texttt{\chapter} macros are supported, and the following macros are automatically expanded: \texttt{\makefrontcover}, \texttt{\upmpubpage}, \texttt{\upmdocsummary}, \texttt{\makebackcover}. This behaviour may be overridden by the other class options.

- \texttt{report}: A report-specification is a one-sided document composed of chapters (no part), and with a document information page. This option indicates to \texttt{upmethodology-document} to load the \LaTeX{} standard \texttt{report} class. In addition the \texttt{\part} macro is ignored\footnote{The macro is redefined to print a warning message when used, no error message is generated.} and \texttt{\chapter} macro is supported, and the following macros are automatically expanded: \texttt{\makefrontcover}, \texttt{\upmdocsummary}, \texttt{\makebackcover}. This behaviour may be overridden by the other class options.

- \texttt{article}: An article-specification is a one-sided document composed of sections (no part nor chapter). This option indicates to \texttt{upmethodology-document} to load the \LaTeX{} standard \texttt{article} class. In addition the \texttt{\part} and \texttt{\chapter} macros are ignored\footnote{The macro is redefined to print a warning message when used, no error message is generated.}, and the following macros are automatically expanded: \texttt{\makefrontcover}, \texttt{\makebackcover}. This behaviour may be overridden by the other class options.

3.2/ CLASS OPTIONS

Table\ref{table:options} contains the options supported by \texttt{upmethodology-document}. Any option not explicitly supported by the class is directly passed to the underlying standard \LaTeX{} class (\texttt{book}, \texttt{report} or \texttt{article} according to the type of document, see\ref{table:options}).
3.3/ **Additional Features**

*upmethodology-document* provides a constant behaviour for all types of document:

- `hyperref` is loaded and set with the document informations;
- `\setpdc` is redefined and linked to `hyperref`;
## 3.3. ADDITIONAL FEATURES

<table>
<thead>
<tr>
<th>Option</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>book</td>
<td>see section 3.1</td>
</tr>
<tr>
<td>report</td>
<td>see section 3.1</td>
</tr>
<tr>
<td>article</td>
<td>see section 3.1</td>
</tr>
<tr>
<td>oneside</td>
<td>the document is generated assuming that each page will be printed on its recto side. This option overrides any previous occurrence of twoside option.</td>
</tr>
<tr>
<td>twoside</td>
<td>the document is generated assuming that each page will be printed on both recto and verso sides. This option overrides any previous occurrence of oneside option.</td>
</tr>
<tr>
<td>francais</td>
<td>same as french</td>
</tr>
<tr>
<td>french</td>
<td>the document is written in French. upmethodology packages use the French translations for the generated texts. This option overrides any previous occurrence of english option.</td>
</tr>
<tr>
<td>english</td>
<td>the document is written in English. upmethodology packages use the English translations for the generated texts. This option overrides any previous occurrence of french option.</td>
</tr>
<tr>
<td>documentinfo</td>
<td>invoke \upmdocumentsummary, \upmdocumentauthors, \upmdocumentvalidators, \upmdocumentinformedpeople, and \upmhistory macros at the begining of the document. This option overrides any previous occurrence of nodocumentinfo option.</td>
</tr>
<tr>
<td>nodocumentinfo</td>
<td>do not invoke \upmdocumentsummary, \upmdocumentauthors, \upmdocumentvalidators, \upmdocumentinformedpeople, nor \upmhistory macros at the begining of the document. This option overrides any previous occurrence of documentinfo option.</td>
</tr>
<tr>
<td>pubpage</td>
<td>invoke \upmpublicationpage macro at the begining of the document. This option overrides any previous occurrence of nopubpage option.</td>
</tr>
<tr>
<td>nopubpage</td>
<td>do not invoke \upmpublicationpage macro at the begining of the document. This option overrides any previous occurrence of pubpage option.</td>
</tr>
<tr>
<td>standardlist</td>
<td>disable the override of the lists (enumeration, etc.) for restoring the standard LATEX lists.</td>
</tr>
<tr>
<td>frontmatter</td>
<td>invoke \frontmatter (and other related macros).</td>
</tr>
<tr>
<td>nofrontmatter</td>
<td>do not invoke \frontmatter (and other related macros).</td>
</tr>
<tr>
<td>frontcover</td>
<td>put the cover page at the beginning of the document.</td>
</tr>
<tr>
<td>nofrontcover</td>
<td>do not put a cover page at the beginning of the document.</td>
</tr>
<tr>
<td>backcover</td>
<td>put the cover page at the end of the document.</td>
</tr>
<tr>
<td>nobackcover</td>
<td>do not put a cover page at the end of the document.</td>
</tr>
<tr>
<td>standardlists</td>
<td>The style does not override the standard list, description and enumeration definitions.</td>
</tr>
<tr>
<td>codepackage</td>
<td>Include the upmethodology-code package.</td>
</tr>
<tr>
<td>specpackage</td>
<td>Include the upmethodology-spec package.</td>
</tr>
<tr>
<td>taskpackage</td>
<td>Include the upmethodology-task package.</td>
</tr>
</tbody>
</table>

Table 3.1: Options of upmethodology-document class
4

PACKAGE UPMETHODOLOGY-VERSION

Version: 2013/08/26

The package upmethodology-version makes it possible to set the version and the status of the document. It also provides functions to manage the document history;

4.1/ Constants for the Document Status

Some \LaTeX\ variables provide strings that describe the status of the document. They can be used in functions such as \texttt{updateversion}.

- \texttt{upmrestricted}: the document is under a restricted access, generally corresponding to the list of authors;
- \texttt{upmvalidable}: authors indicates with this flag that the document could be sent to validators;
- \texttt{upmvalidated}: the document was validated, but not published;
- \texttt{upmpublic}: the document published and accessible to all people;

4.1.1/ Information about the Document

The following functions permit to access to the informations about the document:

- \texttt{theupmversion}: replies the last version number for the document;
- \texttt{upmdate\{version\}}: replies the updating date of the document corresponding to the given version number;
- \texttt{upmdescription\{version\}}: replies the updating comment of the document corresponding to the given version number;
- \texttt{upmstatus\{version\}}: replies the status of the document corresponding to the given version number;
- \texttt{theupmdate}: replies the last updating date for the document. It is equivalent to \texttt{upmdate\{theupmversion\}};
- \texttt{theupmlastmodif}: replies the last updating comment for the document. It is equivalent to \texttt{upmdescription\{theupmversion\}};
- \texttt{theupmstatus}: replies the last status for the document. It is equivalent to \texttt{upmstatus\{theupmversion\}};
4.2/ **Register Revisions**

The package `upmethodology-version` makes it possible to register revisions for building an history. The available functions are:

- \texttt{\textbackslash updateversion}[\texttt{version}][\texttt{date}][\texttt{description}][\texttt{status}]: registers a revision for the document. The revision indicates that the given version was produced at the given date. A small description of the changes and the resulting document’s status must be also provided. The function \texttt{\textbackslash updateversion} is a generalization of the following functions;

- \texttt{\textbackslash initialversion}[\texttt{version}][\texttt{date}][\texttt{description}][\texttt{status}]: registers the initial version of the document. If not given, the version is assumed to be \texttt{0.1};

- \texttt{\textbackslash incversion}[\texttt{date}][\texttt{description}][\texttt{status}]: registers a revision corresponding to the next major version. For example, if the version number was \texttt{2.67} before \texttt{\textbackslash incversion}, this function add the version \texttt{3.67} with the given informations (incrementation of the major part of the version number);

- \texttt{\textbackslash incsubversion}[\texttt{date}][\texttt{description}][\texttt{status}]: registers a revision corresponding to the next minor version. For example, if the version number was \texttt{2.67} before \texttt{\textbackslash incsubversion}, this function add the version \texttt{2.68} with the given informations (incrementation of the minor part of the version number);

4.3/ **Formatted List of Versions**

To obtain a formatted list of versions, you could use the macro \texttt{\textbackslash upmhistory}[\texttt{width}] which produces:

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Updates</th>
</tr>
</thead>
<tbody>
<tr>
<td>23.0</td>
<td>2017/02/17</td>
<td>Replace the package \texttt{subfigure} by \texttt{subcaption}.</td>
</tr>
<tr>
<td>23.1</td>
<td>2017/03/10</td>
<td>Fixing \texttt{subfigure} invalid alignment.</td>
</tr>
<tr>
<td>23.2</td>
<td>2017/08/08</td>
<td>Fixing spelling errors and typos.</td>
</tr>
<tr>
<td>23.3</td>
<td>2017/11/28</td>
<td>Add \texttt{'standardlists'} option.</td>
</tr>
<tr>
<td>23.4</td>
<td>2019/08/04</td>
<td>Add \texttt{'graphicspathcontext'} option.</td>
</tr>
<tr>
<td>24.0</td>
<td>2019/09/17</td>
<td>Add class options to include optional packages.</td>
</tr>
</tbody>
</table>

4.4/ **Localization**

The following macros defines some localized strings used by `upmethodology-version`:

- \texttt{\textbackslash upm@lang@date}: Date;
- \texttt{\textbackslash upm@lang@updates}: Updates;
- \texttt{\textbackslash upm@lang@version}: Version;
- \texttt{\textbackslash upm@lang@version@history}: Version History;
The package upmethodology-fmt provides some useful facilities to format a document.

5.1/ Default Configuration for the Package graphicx

The package graphicx is included, and the following configuration is applied:

- **Image extensions**: By default, the supported image extensions are, in the preference order: pdf, png, jpg, jpeg, tiff, gif. Note that, the tiff picture format is not always supported by the \TeX tools.

  To redefine these extensions, you must invoke:

  \DeclareGraphicsExtensions{extensions}

  where extensions must be replaced by a list of extensions separated by commas.

  Example: \DeclareGraphicsExtensions{.pdf,.png,.eps}

- **Image search path**: By default, the images are search inside the path "./". To redefine the search paths, you must invoke: \graphicspath{\{path1,\{path2,\{path3\}...\}}

  where path1, path2, path2, etc. must be replaced by the names of the directories in which the images are located. The paths in the list are separated by commas. Do not forget to write a slash or a backslash character (depending on the path naming conventions for your operating system) at the end of each path.

  Example: \graphicspath{\{./imgs/\},\{./imgs/auto/\}}

5.2/ Contextual Search Path for graphicx

As described into the previous section, the graphicx package is able to search for files into a set of defined paths.

In order to define a search path that is valid for a part of the document, the \texttt{graphicspathcontext} environment is defined. This environment redefines the graphicx path with the environment’s parameter. The original value of the graphicx path is restored when existing of the environment.

The defined environment is:
\begin{verbatim}
\begin{graphicspathcontext}[path]
...
\end{graphicspathcontext}
\end{verbatim}
The parameter path must follow the syntactic definition of the graphicx path. If you want to reuse the current value of the graphicx path, you could obtain it by using the \old macro, such as:
\begin{graphicspathcontext}[mypath,\old]
\end{graphicspathcontext}

Note that \old must not be inside curly braces.

5.3/ Figures

It may be verbose to put \LaTeX code to include a figure inside your document. To simplify your life, you could include a figure with the following macros:
\mfigure[position][include_graphics_options][filename][caption][label]
\mfigure*[position][include_graphics_options][filename][caption][label]

These two macros make it possible to include an image in your document. The parameters are:

- position: is the desired position of the figure (see \begin{figure}[position]). It could be t (top of the page), b (bottom of the page), h (at the macro location if possible) or H (at macro location);
- include_graphics_options: are the options passed to \includegraphics;
- filename: is the filename passed to \includegraphics;
- caption: is the caption of the figure (see \caption{caption});
- label: is the label used to reference the figure (see \label{fig:label}).

The difference between \mfigure and \mfigure* is the same as the difference between \begin{figure} and \begin{figure*}: the star-version fits to the entire paper width event if the document has two or more columns.

Because the two macros above register a label with string starting with fig:, we propose the following function to easily access to the figure’s references:

- \figref{label}: is equivalent to \ref{fig:label};
- \figpageref{label}: is equivalent to \pageref{fig:label}.

The figure 5.1 page 29 is obtained with the macro: \mfigure[ht][width=.4\linewidth][slogo][Example of figure inclusion with texttt{[textbackslash]mfigure}][example:mfigure].
The reference and page reference are obtained with \figref{example:mfigure} and \figpageref{example:mfigure}.

5.4/ Sub-figures

In some case, it is useful to put several images inside the same floating figure, but without loosing the possibility to reference each of the subfigures. This feature was proposed by the package subcaption. The following environments provides helper functions for subcaption:
\begin{mfigures}[position][caption][label]
\end{mfigures}
\begin{mfigures*}[position][caption][label]
\end{mfigures*}
5.5. FIGURES WITH EMBEDDED \TeX MACROS

Figure 5.1: Example of figure inclusion with \mfigure

\end{mfigures*}

These two macros enable you to include an image in your document. The parameters are:

- **position**: is the desired position of the figure (see \begin{figure}{position}). It could be t (top of the page), b (bottom of the page), h (at the macro location if possible) or H (at macro location);
- **caption**: is the caption of the figure (see \caption{caption});
- **label**: is the label used to reference the figure (see \label{fig:label}).

Inside the environment \mfigures[*], you could use the macro \mfigure to properly include a subfigure (the first optional parameter is ignored), or you could use the macro \msubfigure\{options\}\{file\}\{caption\}.

The figure 5.2 page 30 is obtained with the environment:
\begin{mfigures}\Example of subfigures with \texttt{msubfigures}\{example:msubfigure}\mfigure\{width=.4\linewidth\}\{img1\}\{First subfigure\}\{example:firstsubfigure\}\hspace{1cm}\msubfigure\{width=.4\linewidth\}\{img2\}\{Second subfigure\}\end{mfigures}

The reference and page reference are obtained with \figref{example:msubfigure} and \figpageref{example:msubfigure}.

The references to the subfigures could be obtained in two way:

- using the label given as the last parameter of \mfigure, eg. the label example:firstsubfigure corresponds to 5.2a;
- using the label of the enclosing figure to which the index of the subfigure could be appended (in its Roman representation and prefixed by the character ":"), eg. the label example:msubfigure:b corresponds to 5.2b.

5.5/ FIGURES WITH EMBEDDED \TeX MACROS

In several cases it is useful to include \TeX macros inside a figure. It is possible to combine figures and \TeX macros. Several figure editors provide exporting features to obtain combined figures: xfig, inkscape, GNU Plot, etc. Basically, these tools create two files per source figure:

- the figure in PDF or Postscript format (filename extensions, .pdf or .ps); and
To include this combined figure in your document, you simply need to include the generated $\LaTeX$ file (see below for details).

### 5.5.1 Include a Combined Picture/$\LaTeX$ Figure

To include a figure with $\LaTeX$ macros inside, you must have:

1. a Postscript figure (.eps), and a $\LaTeX$ file .pstex.t related to the Postscript figure; or
2. a PDF figure (.pdf), and a $\LaTeX$ file .pdftex.t related to the PDF figure.

With the `upmethodology-fmt` package, the inclusion of the figure with embedded $\LaTeX$ macros is similar to the inclusion of figures with \includegraphics. You must type the following macro:

\begin{verbatim}
\includegraphicswtex[options]{filename}
\end{verbatim}

where options must be one or more of:

- **width=xxx**: specification of the width of the figure (xxx must be replaced by the length);
- **height=xxx**: specification of the height of the figure (xxx must be replaced by the length);

If the filename given to the macro \includegraphicswtex does not specify a filename extension, the macro tries to add the extensions .pdftex.t, .pstex.t, .pdf.tex, or .ps-.tex, by default. If you want to specify other file extensions, you must use the macro:\

\begin{verbatim}
\DeclareGraphicsExtensionsWtex[extensions]
\end{verbatim}

where the extensions is a list of file extensions (including the point character), separated by coma characters.

Example: \begin{verbatim}
\DeclareGraphicsExtensionsWtex[.pdftex,.pstex]
\end{verbatim}

If the filename does not correspond to a file on the disk, the macro \includegraphicswtex tries to find the file in the directories specified in \graphicspath (declared in the package graphicx for example).

Example: \graphicspath[./imgs/,./imgs/additional/]

Note that each of the given directories must be finished by the separation character of your operating system: / on Unix, \ on Windows. You must always use the Unix standard because it is assumed by a lot of $\LaTeX$ compilers, even on Windows platforms.

Figure 5.3 gives an example of a floating figure combined with $\LaTeX$ macros, which is using the macro \includegraphicswtex.
5.5. FIGURES WITH EMBEDDED $\LaTeX$ MACROS

5.5.2/ FLOATING FIGURE WITH EMBEDDED $\LaTeX$ MACROS

To put a floating figure with $\LaTeX$ macro inside, you may use one of the macros:
\mfigurewtex[position]|include_graphics_options|filename|caption|label
\mfigurewtex*[position]|include_graphics_options|filename|caption|label

The parameters are:

- **position**: is the desired position of the figure (see \begin{figure}[position]). It could be t (top of the page), b (bottom of the page), h (at the macro location if possible) or H (at macro location);
- **include_graphics_options**: are the options to pass to \includegraphicswtex. For ascendant compatibility, if you pass a length without a key, e.g. \{.8\textwidth\}, the length is assumed to be the width of the figure;
- **filename**: is the name of the file of the figure (see \includegraphicswtex for details);
- **caption**: is the caption of the figure (see \caption{caption});
- **label**: is the label used to reference the figure (see \label{fig:label}).

The difference between \mfigurewtex and \mfigurewtex* is the same as the difference between \begin{figure} and \begin{figure*}: the star-version fits to the entire paper width event if the document has two or more columns.

Because the two macros above register a label with string starting with fig:, the macros \figref and \figpageref could be used.

Example of a figure combined with $\LaTeX$ macros

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{example}
\caption{Example of a figure combined with $\LaTeX$ macros}
\end{figure}

Figure 5.3 gives an example of a floating figure combined with $\LaTeX$ macros. Note that:

- the title of the figure contains the macro \LaTeX, which produces: $\LaTeX$;
- a small equation, written in $\LaTeX$, is put between the two planes;
5.5.3 Helpers for embedded \TeX

To help you to put \TeX macros in a figure, and to define its real test inside the \LaTeX document, several functions are provided:

- \texttt{\figmath\{id\}|expr} will associate to the given identifier the given mathematical expression,
- \texttt{\figtext\{id\}|expr} will associate to the given identifier the given text expression;

These expressions, defined with the two previous functions, may be referenced in the figure by a \TeX macro with a name similar to \texttt{\FIG\delta}, where \(\delta\) must be replaced by an identifier of your choice and used as parameter of one of the two previous functions (example: \texttt{\FIGmyid}).

Figure 5.3 gives an example where the equation is written as: \texttt{\FIGexampleofexpression} in the figure, and it is replaced by the real equation with:\texttt{\figmath|exampleofexpression|t_i = \sum_i \left(\alpha + \beta\right)}

5.6 Tabulars

You could include a tabular inside your document with the following environment:\texttt{\begintabular\{width\}[ncolumns]|columns|...\endtabular}\n
This tabular is an extension of the \texttt{tabularx} environment which provides dynamic columns with the specifier \texttt{X}. The parameters are:

- \texttt{width}: is the desired width of the tabular;
- \texttt{ncolumns}: is the count of columns in the tabular. It must be consistent with the column description;
- \texttt{columns}: is the description of the columns according to the \texttt{tabular} and \texttt{tabularx} packages.

You must note put any text nor \TeX macro before the first use of \texttt{\tabulartitle} or \texttt{\tabularheader}. Otherwhise, you will obtain a \TeX error.

The \texttt{mtabular} environment provides:

- \texttt{\tabulartitle\{title\}}
  This macro allows you to define the title of the tabular. It uses the colors backtableheader and fronttableheader for the background and the foreground respectively. The title has a single line at the top, and a single line below;

- \texttt{\tabulartitleinside\{title\}}
  This macro allows you to define the title of the tabular. It uses the colors backtableheader and fronttableheader for the background and the foreground respectively. The title has two lines at the top, and a single line below;

- \texttt{\tabularheader\{header\}_1|...|\{header\}_n}
  This macro allows you to define the titles of the columns. It uses the colors backtableheader and fronttableheader for the background and the foreground respectively. Because the count of columns was given to the environment this function takes the same count of parameters as the count of columns. This macro adds a line after the header, BUT NOT BEFORE.
5.7. TABLES

Because \tabularheader is adding a \hline at the end of its expansion. You must put a \tabularheader just after \tabularheader. Otherwise you may obtain a \TeX error.

- \tabularrowheader{title}
  This macro is designed to be used in the first cell of a row. It is rendering the cell as a row’s header.

- \tabulartitlespec{column_spec}
  This macro defines the specification of the column used to render the title of the table. The default value of the column specification is |c|.

The following example of table is obtained by:

\begin{mtabular}{\linewidth}{4}{lXrX}
  \tabulartitle{Example of \texttt{mtabular}}
  \tabularheader{Col1}{Col2}{Col3}{Col4}
  a & b & c & d \\
  \hline
  e & f & g & h \\
  \tabulartitleinside{Example of second title in the table}
  \hline
  \tabularrowheader{i} & j & k & l \\
  \tabularheader{Col1-2}{Col2-2}{Col3-2}{Col4-2}
  m & n & o & p \\
\end{mtabular}

\begin{tabular}{|l|c|c|c|}
  \hline
  \textbf{Col1} & \textbf{Col2} & \textbf{Col3} & \textbf{Col4} \\
  \hline
  a & b & c & d \\
  e & f & g & h \\
  \hline
  \textbf{Example of second title in the table} \\
  \hline
  \textbf{Col1-2} & \textbf{Col2-2} & \textbf{Col3-2} & \textbf{Col4-2} \\
  \hline
  m & n & o & p \\
\end{tabular}

5.7/ Tables

You could include a table inside your document with the following environment:

\begin{mtable}[options][width][ncolumns][columns][caption][label]...
\end{mtable}

This environment is based on the \mtabular environment. The parameters are:

- options: are the options to pass to the \mtable environment:
  - a table placement composed of one or more of the following characters. The order in which the placement options are specified does not make any difference, as the placement options are always attempted in the order h-t-b-p. Thus [hb] and [bh] are both attempted as h-b. The more float placement options are given to \LaTeX, the better it handles float placement. Consequently, and because we want a simple \TeX code in the background, all the permutations are not supported by the \mtable environment. We recommend to put placement letters in the order they appear in the following list:
• h: Place the float here, i.e., approximately at the same point it occurs in the source text (however, not exactly at the spot),
• t: Position at the top of the page,
• b: Position at the bottom of the page,
• p: Put on a special page for floats only,
• H: Places the float at precisely the location in the LATEX code. Requires the float package.
This is somewhat equivalent to h!;
• !: Override internal parameters LATEX uses for determining “good” float positions,

If you specify more than one table placement in the options, the last one is used.
- size=<macro>: specify the size of the text in the table (by default, \normalsize);
  • width: is the desired width of the table (ie., the tabular inside the table);
  • ncolumns: is the count of columns in the table (ie., the tabular inside the table). It must be consistent with the column description;
  • columns: is the description of the columns according to the tabular and tabularx packages;
  • caption: is the caption of the table;
  • label: is the label referencing the table.

Because the mtable environment registers a label with a string starting with tab:, the following functions are proposed to easily access to the table’s references:
  • \tabref{label}: is equivalent to \ref{tab:label};
  • \tabpageref{label}: is equivalent to \pageref{tab:label}.

The table [5.1] page [34] is an illustration of the following LATEX code:
\begin{mtable} \linewidth \XrX \Example of \texttt{mtable} \example:mtable \captionastitle \tabularheader{Col1}{Col2}{Col3}{Col4}
a & b & c & d \hline
e & f & g & h \hline
\end{mtable}

<table>
<thead>
<tr>
<th>Col1</th>
<th>Col2</th>
<th>Col3</th>
<th>Col4</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
</tr>
<tr>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
</tr>
</tbody>
</table>

Table 5.1: Example of mtable

The macro \captionastitle is equivalent to a call to the macro \tabulartitle with the caption in parameter.

5.8/ Enumerations

The package upmethodology-fmt provides a set of macros dedicated to enumeration lists.
5.8. ENUMERATIONS

5.8.1/ Enumeration Counters

Sometimes it is useful to start an enumeration list from a specific given number. This package provides several macros for saving and restoring the counter use by the enumeration lists.

> Only one counter could be saved at a given time. It means that you cannot save the counters for an enumeration and for an enclosing enumeration at the same time.

Two general macros are defined for helping you to save a counter value into the global variable:

- \texttt{\textbackslash savecounter\{name\}}
  save the value of the counter identified by the given name in a global variable. The name of the counter must be previously defined with one of the standard \LaTeX or \TeX macros, e.g. \texttt{\textbackslash newcounter};

- \texttt{\textbackslash restorecounter\{name\}}
  put the previously saved value into the counter with the given name. The name of the counter must be previously defined with one of the standard \LaTeX or \TeX macros, e.g. \texttt{\textbackslash newcounter};

The counter is extensively used in enumeration lists. The following macros will help you for managing the enumeration counter:

- \texttt{\textbackslash setenumcounter\{value\}}
  force the value of the counter used by the enumeration environments;

- \texttt{\textbackslash getenumcounter}
  replies the value of the counter used by the enumeration environments;

- \texttt{\textbackslash saveenumcounter}
  save the value of the counter used by the enumeration environment with \texttt{\textbackslash savecounter};

- \texttt{\textbackslash restoreenumcounter}
  restore the value of the counter used by the enumeration environment with \texttt{\textbackslash restorecounter}.

Example: The following \LaTeX code produces the result below:

This is a text: \texttt{\begin\{enumerate\}
\item This is an item.
\item This is another item.
\texttt{\textbackslash saveenumcounter}
\end\{enumerate\}}
This is a text in the between.
\texttt{\begin\{enumerate\}
\texttt{\textbackslash restoreenumcounter}
\item The list goes on
\item and on.
\end\{enumerate\}}
This is a second text in the between.
\texttt{\begin\{enumerate\}
\texttt{\textbackslash setenumcounter\{18\}}
\item The list goes on again
\item and on.
\end\{enumerate\}} This is the text after.
5.8.2/ **Inline Enumeration**

In several documents, an enumeration of things is written inside a paragraph instead of inside a list of points.

**Example:** The following \LaTeX code produces the result below:

```latex
This is a text: \begin{inlineenumeration}
\item first thing;
\item second thing;
\item etc.
\end{inlineenumeration} This is the text after.
```

This is a text: (i) first thing; (ii) second thing; (iii) etc. This is the text after.

5.9/ **Environment Description**

The environment description is redefined as following:

```latex
\begin{description}[separator]
\item[desc] text
\end{description}
```

The text put in place of desc represents the text which may be emphasized in the description item. The separator is the text that is inserted at the end of the head of each description item.

**Example 1:** The following \LaTeX code, using Roman numbers, produces the description just below:

```latex
\begin{description}
\item[first thing] this is a text for the first thing;
\item[second thing] this is a text for the second thing;
\item[more] etc.
\end{description}
```

- **first thing:** this is a text for the first thing;
- **second thing:** this is a text for the second thing;
- **more:** etc.
5.10. DESCRIPTIONS IN CONJUNCTION WITH ENUMERATION

Example 2: The following \LaTeX code produces the description just below:

\begin{description}
\item[first thing] this is a text for the first thing;
\item[second thing] this is a text for the second thing;
\item[more] etc.
\end{description}

- first thing/// this is a text for the first thing;
- second thing/// this is a text for the second thing;
- more/// etc.

5.10/ DESCRIPTIONS IN CONJUNCTION WITH ENUMERATION

It may be helpful to put a list of descriptions in conjunction with an enumeration. In other words, the following environment provides a mix between the standards \LaTeX environments description and enumerate.

5.10.1/ ENVIRONMENT ENUMDESCRIPTION

The environment \texttt{enumdescription} is:

\begin{verbatim}
\begin{enumdescription}[type]
\item[desc] text
\end{enumdescription}
\end{verbatim}

where the \texttt{type} is the type of the enumeration. It may be one of:

- "i": for an enumeration with Roman numbers (this is the default),
- "1": for an enumeration with Arabic numbers,
- "a": for an enumeration with letters.

The text put in place of \texttt{desc} represents the text which may be emphasized in the description item.
To change the rendering of the labels, you must redefine the macro as:

\begin{verbatim}
\renewcommand{\enumdescriptionlabel}[1]{ ... #1 ... }
\end{verbatim}

To change the separator between the counter and the description, you must redefine the macro as:

\begin{verbatim}
\renewcommand{\enumdescriptioncounterseparator}{ ... }
\end{verbatim}

To change the separator between the description and the rest of the text, you must redefine the macro as:

\begin{verbatim}
\renewcommand{\enumdescriptionlabelseparator}{ ... }
\end{verbatim}
Example 1: The following \LaTeX{} code, using Roman numbers, produces the enumerated description just below:

\begin{enumdescription}
\item[first thing] this is a text for the first thing;
\item[second thing] this is a text for the second thing;
\item[more] etc.
\end{enumdescription}

i - first thing: this is a text for the first thing;
ii - second thing: this is a text for the second thing;
iii - more: etc.

Example 2: The following \LaTeX{} code, using numeric numbers, produces the enumerated description just below:

\begin{enumdescription}[1]
\item[first thing] this is a text for the first thing;
\item[second thing] this is a text for the second thing;
\item[more] etc.
\end{enumdescription}

1 - first thing: this is a text for the first thing;
2 - second thing: this is a text for the second thing;
3 - more: etc.

Example 3: The following \LaTeX{} code, using letter numbers, produces the enumerated description just below:

\begin{enumdescription}[a]
\item[first thing] this is a text for the first thing;
\item[second thing] this is a text for the second thing;
\item[more] etc.
\end{enumdescription}

a - first thing: this is a text for the first thing;
b - second thing: this is a text for the second thing;
c - more: etc.

5.10.2/ Environment \texttt{enumerate}

The environment \texttt{enumerate} exists in the standard \LaTeX{} distributions. The UPM package redefines this environment to provide a behavior similar to the one of the environment \texttt{enumdescription}.

Additionally, you could specify the format of the counter in the first optional parameter. This format is a text in which the first occurrence of one of the following characters is replaced by the value of the counter with the associated number format:

- 1: the counter is an arabic number;
- a: the counter is a sequence of lower-case alphabetic letters;
5.10. DESCRIPTIONS IN CONJUNCTION WITH ENUMERATION

- \textit{A}: the counter is a sequence of upper-case alphabetic letters;
- \textit{i}: the counter is a lower-case roman number;
- \textit{I}: the counter is an upper-case roman number.

\textbf{Example 1:} The following \LaTeXX code produces a list, which is similar to the one generated by the standard \LaTeXX environment \texttt{enumerate}:

\begin{verbatim}
\begin{enumerate}
\item this is a text for the first thing;
\item this is a text for the second thing;
\item etc.
\end{enumerate}
\end{verbatim}

\begin{verbatim}
1. this is a text for the first thing;
2. this is a text for the second thing;
3. etc.
\end{verbatim}

\textbf{Example 2:} The following \LaTeXX code illustrates how the environment is reacting to a given description:

\begin{verbatim}
\begin{enumerate}
\item this is a text for the first thing;
\item[description] this is a text for the second thing;
\item etc.
\end{enumerate}
\end{verbatim}

\begin{verbatim}
1. this is a text for the first thing;
2. description: this is a text for the second thing;
3. etc.
\end{verbatim}

\textbf{Example 3:} The following \LaTeXX code illustrates the alphabetic counter specification. Note that the parenthesis characters are directly rendered in the list:

\begin{verbatim}
\begin{enumerate}[(a)]
\item this is a text for the first thing;
\item this is a text for the second thing;
\item etc.
\end{enumerate}
\end{verbatim}

\begin{verbatim}
(a) this is a text for the first thing;
(b) this is a text for the second thing;
(c) etc.
\end{verbatim}

\textbf{Example 4:} The following \LaTeXX code illustrates the roman counter specification. Note that the dot character is directly rendered in the list:

\begin{verbatim}
\begin{enumerate}[I.]
\item this is a text for the first thing;
\item this is a text for the second thing;
\item etc.
\end{enumerate}
\end{verbatim}
I. this is a text for the first thing;
II. this is a text for the second thing;
III. etc.

5.10.3/ Environment enumdescriptionx

The environment enumdescriptionx extends the environment enumdescription by enabling a finer configuration with more parameters.

The environment enumdescriptionx is:

\begin{enumdescriptionx}[type]{counter\_prefix}{counter\_postfix}
\item[desc] text
\end{enumdescriptionx}

where the type is the type of the enumeration. It may be one of:

- **“i”**: for an enumeration with Roman numbers (this is the default),
- **“1”**: for an enumeration with Arabic numbers,
- **“a”**: for an enumeration with letters.

The text put in place of desc represents the text which may be emphasized in the description item. The text counter\_prefix is put before all the counter values in the enumeration. The text counter\_postfix is put after all the counter values in the enumeration.

To change the rendering of the labels, you must redefine the macro as:

\renewcommand{\enumdescriptionlabel}[1]{ ... #1 ... }

Example: The following \LaTeX code, using letter numbers, produces the enumerated description just below:

\begin{enumdescriptionx}[a]{$\langle$}{$\rangle$}
\item[first thing] this is a text for the first thing;
\item[second thing] this is a text for the second thing;
\item[more] etc.
\end{enumdescriptionx}

⟨a⟩ - first thing: this is a text for the first thing;
⟨b⟩ - second thing: this is a text for the second thing;
⟨c⟩ - more: etc.

5.11/ Footnotes

The package upmethodology-fmt provides a set of macros allowing to save the reference number of a footnote and to recall this reference many time as required.
5.12. **UML DIAGRAMS ON THE SIDE OF PARAGRAPHS**

The package `upmethodology-fmt` provides an environment that makes it possible to put an UML diagram (or any other picture) on the side of a paragraph.

- \begin{umlinpar}[width][picture_path]
  text
  \end{umlinpar}

  put the specified picture on the side of the given text. The optional parameter `width` corresponds to the desired width of the picture. By default it is \$0.5\textwidth$.

This paragraph is an typical example of the usage of the environment `umlinpar`. To obtain it, the following \LaTeX\ code was typed:

\begin{verbatim}
\begin{umlinpar}[smalllogo]
  This paragraph is an typical example of the usage of the environment \texttt{umlinpar}.
\end{umlinpar}
\end{verbatim}

Example of a figure combined

\textbf{Example of a figure combined}

5.13/ **DATE FORMATTING**

Because the concept of date was important and unfortunately localized, this package provides a set of functions to define and extract information from dates (the supported date formats are described in table 5.2):

---

1 This is an example of a recallable footnote
2 This is a second example of a recallable footnote
3 This is a third example of a recallable footnote
\makedate{day}{month}{year}
allows you to create the text corresponding to the given date according to the current localized date format.

\extractyear{formatted_date}
extract the year field from a date respecting the localized date format.

\extractmonth{formatted_date}
extract the month field from a date respecting the localized date format.

\extractday{formatted_date}
extract the day field from a date respecting the localized date format.

<table>
<thead>
<tr>
<th>yyyy/mm/dd</th>
<th>default format</th>
</tr>
</thead>
<tbody>
<tr>
<td>dd/mm/yyyy</td>
<td>french format</td>
</tr>
</tbody>
</table>

Table 5.2: List of supported date formats

5.14/ Text formatting

The package \texttt{upmethodology-fmt} provides a set of macros to format the text.

\textsup{text}
put a text as exponent in text mode instead of the basic \LaTeX exponent in math mode. In opposite to the standard \LaTeX macro ~\texttt{textsuperscript}, this macro adds an extra space after the macro when needed.
Example: \texttt{\textsup{this is an exponent}} this is an exponent this is the following text;

\textup{text}
same as \textsup.

\textsub{text}
put a text as indice in text mode instead of the basic \LaTeX indice in math mode. In opposite to \texttt{textsubscript}, this macro adds an extra space after the macro when needed. In opposite to \texttt{textdown}, the size of the text is not changed in the text down.
Example: \texttt{\textsub{this is an indice}} this is an indice this is the following text;

\textdown{text}
put a text as indice in text mode instead of the basic \LaTeX indice in math mode. In opposite to \texttt{textsubscript}, this macro adds an extra space after the macro when needed. In opposite to \texttt{textsup}, the size of the text is changed in the text down.
Example: \texttt{\textdown{this is an indice}} this is an indice this is the following text;

\textsubscript{text}
put a text as indice in text mode instead of the basic \LaTeX indice in math mode. As for the standard \LaTeX macro \texttt{textsuperscript}, this macro does not add an extra space after the macro.
Example: \texttt{\textsubscript{this is an indice}} this is an indice this is the following text;

\Emph{text}
put a \texttt{very important} text. This macro is similar to the standard \LaTeX macro \texttt{emph}. The difference is: \texttt{emph} is for “important things”; and \texttt{Emph} is for “very important things”.
Example: This text is \texttt{emph{important}}, but this one is \texttt{Emph{very important}} gives: This text is important, but this one is \texttt{very important};
5.14. TEXT FORMATTING

- `{\makefilename[von]{first name}{last name}}`
  format the specified people name components according to the document standards. By default, the format first von last is used.
  Example: `{\makefilename[von]{Ludwig Otto Frederik Wilhelm}{Wittelsbach}}, "LUDWIG OTTO FREDERIK WILHELM VON WITTELSBACH";`

- `{\upmmakefilename[von]{first name}{last name}{separator}}`
  format the specified people name components according to the document standards. By default, the format first von last is used.
  Example: `{\upmmakefilename[von]{Ludwig Otto Frederik Wilhelm}{Wittelsbach}{/}}, "LUDWIG OTTO FREDERIK WILHELM/VON WITTELSBACH";`

- `{\makenamespacing{name}}`
  format the specified name to be sure that the spaces after the points of the initials are demi-spaces.
  Example: `{\makenamespacing{S.G.}}{Galland}, "S. G. Galland";`

- `{\makelastname[name]}`
  format the specified last/family name.
  Example: `{\makelastname{Galland}}, "Galland";`

- `{\makefirstname[name]}`
  format the specified first name.
  Example: `{\makefirstname{Stéphane}}, "Stéphane";`

- `{\prfilename[von]{first name}{last name}}`
  `{\prfilename*[von]{first name}{last name}}`
  format the specified people name components according to the document standards for Professor title. By default, the format first von last is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: `{\prfilename{Pierre}{Martin}}, "Pr. Pierre Martin";`
  Example 2: `{\prfilename*[Pierre]{Martin}}, "Pierre Martin, Pr.";`

- `{\drfilename[von]{first name}{last name}}`
  `{\drfilename*[von]{first name}{last name}}`
  format the specified people name components according to the document standards for Doctor title. By default, the format first von last is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: `{\drfilename{Pierre}{Martin}}, "Dr. Pierre Martin";`
  Example 2: `{\drfilename*[Pierre]{Martin}}, "Pierre Martin, Dr.";`

- `{\phdfilename[von]{first name}{last name}}`
  `{\phdfilename*[von]{first name}{last name}}`
  format the specified people name components according to the document standards for Philosophiae-Doctor title. By default, the format first von last is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: `{\phdfilename{Pierre}{Martin}}, "Ph.D. Pierre Martin";`
  Example 2: `{\phdfilename*[Pierre]{Martin}}, "Pierre Martin, Ph.D.";`

- `{\scdfilename[von]{first name}{last name}}`
  `{\scdfilename*[von]{first name}{last name}}`
  format the specified people name components according to the document standards for Scientiae-Doc- tor title. By default, the format first von last is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: `{\scdfilename{Pierre}{Martin}}, "Sc.D. Pierre Martin";`
  Example 2: `{\scdfilename*[Pierre]{Martin}}, "Pierre Martin, Sc.D.";`
• \mdname[von][first name][last name]
  \mdname*[von][first name][last name]
  format the specified people name components according to the document standards for Medicinæ-Doctor title. By default, the format first von last is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: \mdname[Pierre][Martin], “M.D. PIERRE MARTIN”;
  Example 2: \mdname*[Pierre][Martin], “PIERRE MARTIN, M.D.”;

• \pengname[von][first name][last name]
  \pengname*[von][first name][last name]
  format the specified people name components according to the document standards for Professional/Chartered Engineer title. By default, the format first von last is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: \pengname[Pierre][Martin], “CEng. PIERRE MARTIN”;
  Example 2: \pengname*[Pierre][Martin], “PIERRE MARTIN, CEng.”;

• \iengname[von][first name][last name]
  \iengname*[von][first name][last name]
  format the specified people name components according to the document standards for Incorporated Engineer title. By default, the format first von last is used. The star-ed version is post-fixed, the non-star-ed version is prefixed.
  Example 1: \iengname[Pierre][Martin], “IEng. PIERRE MARTIN”;
  Example 2: \iengname*[Pierre][Martin], “PIERRE MARTIN, IEng.”.

5.15/ Symbols

5.15.1/ Symbols in Text Mode

The package upmethodology-fmt provides several symbols in text mode, and described inside the table 5.3

| \arakhneorg | Arakhnêorg |
| \copyright | © |
| \trademark | ™ |
| \regmark | ® |
| \smalltrade | ® |
| \smallreg | ™ |
| \smallcopy | © |
| \ust | st |
| \und | nd |
| \urd | rd |
| \uth | th |

Table 5.3: List of symbols

5.15.2/ Symbols in Math Mode

The package upmethodology-fmt provides several symbols in math mode, and described inside the table 5.3
5.16. BIBLIOGRAPHY

The package `upmethodology-fmt` provides a set of macros allowing to manage the bibliography. The default bibliography style is `abbr`.

- \bibliographystyle{style}
  set the bibliography style to use.
  Example: \bibliographystyle{alpha};

- \bibliography{file}
  set the Bu\TeX\ file to use.
  Example: \bibliography{mybib};

- \bibsize{size}
  set the font size used for the bibliography section.
  Example: \bibsize{\Huge};

5.17. THEOREMS AND MATHEMATIC ENVIRONMENTS

The package `upmethodology-fmt` defines several environments and macros that are based on the theorem or the math API of \LaTeX.

5.17.1. DEFINITION OF A NEW THEOREM ENVIRONMENT

If you want to create a new theorem environment based on the style provided by this package, you could invoke `\declareupmtheorem`:

```
\declareupmtheorem[name of the style]{name}{label}{title of the list}
```

This macro defines:

- the environment with the given name, and
- the macro `\listof(name)s`.

The name of the style is the name of the theorem style to be used. This style is defined by `\newtheoremstyle`. By default, it is `upmdefinition`. The label is the text to put in the theorem header. The title of the list is used by the macro `\listof(name)s` as the title of the chapter.
Some features provided by this package depend on the version of the package thmtools. We recommend to use and install the version 2012/05/04, or later.

The macro \declareupmtheorem can be used only inside the preamble of your document.

Example: The following code define the environment mytheorem:

\documentclass{upmethodology-document}
\declareupmtheorem{mytheorem}{My Theorem}{List of my Theorems}
\begin{document}
\begin{mytheorem}[Theorem of Everything]
This is the theorem of Everything.
\end{mytheorem}
\end{document}

gives the result:

My Theorem 1: Theorem of Everything
This is the theorem of Everything.

5.17.2/ DEFINITION

The package upmethodology-fmt defines the environment definition to put a definition in your document. This environment is based on the standard theorem environment. The definition takes one optional parameter: the name of the definition.

Example: The following \LaTeX code:

\begin{definition}[Name of the definition]
Text of the definition.
\end{definition}

produces:

Definition 1: Name of the definition
Text of the definition.

Change the colors of the definition: You could change the colors of the definition environment by redifining the colors below with one of the macros \definecolor or \colorlet:

- definitionbackground is the color of the background of the definition;
- definitionborder is the color of the frame;
- definitionheaderforeground is the color of the text in the header of the definition;
- definitiontextforeground is the color of the text in the body of the definition.
**Example of color redefinition:** The following \LaTeX code:

\begin{verbatim}
\definecolor{definitionheaderforeground}{rgb}{.3,.5,.8}
\colorlet{definitionbackground}{gray!20}
\colorlet{definitionborder}{red}
\begin{definition}[Name of the definition]
Text of the definition.
\end{definition}
\end{verbatim}
produces:

<table>
<thead>
<tr>
<th>Definition 2: Name of the definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text of the definition.</td>
</tr>
</tbody>
</table>

**5.18. EMPHAZING BOX**

If you want to create a text that is emphasized with a box, you could use the environment: \\begin{emphbox}[width] text \end{emphbox}

**Example:** The following \LaTeX code:

\begin{verbatim}
\begin{emphbox} [.7\linewidth]
This is an emphasized text.
\end{emphbox}
\end{verbatim}
produces:

This is an emphasized text.

**Change the colors of the emphasizing box:** You could change the colors of the \texttt{emphbox} environment by redefining the colors below with one of the macros \texttt{definecolor} or \texttt{colorlet}:

- \texttt{emphboxbackground} is the color of the background of the environment;
- \texttt{emphboxborder} is the color of the frame;
- \texttt{emphboxtext} is the color of the text in the body of the environment.

**Example of color redefinition:** The following \LaTeX code:

\begin{verbatim}
\colorlet{emphboxbackground}{gray!20}
\colorlet{emphboxborder}{red}
\begin{emphbox}
This is an emphasized text.
\end{emphbox}
\end{verbatim}
produces:

This is an emphasized text.
5.19/ Framed Boxes or Mini Pages

Standard \LaTeX\ distribution provides the minipage environment. This environment allows you to put a small piece of page inside your document. The package upmethodology-fmt provides two framed extensions of the original minipage environment: framedminipage and framedcolorminipage.

The prototypes of these two new environments are, respectively:

- \begin{framedminipage}{width} \ldots \end{framedminipage}
- \begin{framedcolorminipage}{width}{border_color}{background_color} \ldots \end{framedcolorminipage}

Example of \texttt{framedminipage}  The following \LaTeX\ code:

\begin{framedminipage}{.75\linewidth}
This is a text inside a framed minipage.
\end{framedminipage}

produces: This is a text inside a framed minipage.

Example of \texttt{framedcolorminipage}  The following \LaTeX\ code:

\begin{framedcolorminipage}{.75\linewidth}{red}{yellow}
This is a text inside a framed minipage with colors.
\end{framedcolorminipage}

produces: This is a text inside a framed minipage with colors.

5.20/ Message Boxes

The package upmethodology-fmt provides a set of environment to put emphasis message boxes in the text. Three types of boxes are supported: caution, information, and question.
\begin{upmcaution}[width]
This is an example of a caution message. This text must be rendered with enough height (usually 2 lines of text) to avoid intersection between the caution icon and the box frame.
\end{upmcaution}

\begin{upminfo}[width]
This is an example of an information message. This text must be rendered with enough height (usually 2 lines of text) to avoid intersection between the caution icon and the box frame.
\end{upminfo}

\begin{upmquestion}[width]
This is an example of a question message. This text must be rendered with enough height (usually 2 lines of text) to avoid intersection between the caution icon and the box frame.
\end{upmquestion}

5.21/ ADDITIONAL MACROS FOR THE TABLE OF CONTENT

The macro \texttt{newpageintoc} makes it possible to insert a page break inside the table of contents (toc). It may be used to avoid orphan titles in the toc.

5.22/ ADDITIONAL DOCUMENT SECTIONING MACROS

The package \texttt{upmethodology-fmt} provides several macros that permit to create special sections.

5.22.1/ NON-NUMBERED PART IN TABLE OF CONTENT

If you want to add a document part that has no part number but appearing inside the table of content, the classical \LaTeX macros \texttt{part} and \texttt{part*} are inefficient. Indeed, \texttt{part} is adding a numbered part inside the table of content, and \texttt{part*} is adding an unnumbered part but not inside the table of content.

To add an unnumbered part inside the table of content, you could use one of the macros:
\begin{verbatim}
\parttoc[toctitle][title]
\parttoc*[toctitle][title]
\end{verbatim}

The macros \texttt{parttoc} and \texttt{parttoc*} have the same effect except that \texttt{parttoc*} aligns the part’s title to the other numbered parts’ titles; and \texttt{parttoc} not.
5.22.2/ **Non-numbered Chapter in Table of Content**

If you want to add a document chapter that has no chapter number but appearing inside the table of content, the classical \LaTeX\ macros \texttt{\chapter{}} and \texttt{\chapter*{}} are inefficient. Indeed, \texttt{\chapter{}} is adding a numbered chapter inside the table of content, and \texttt{\chapter*{}} is adding an unnumbered chapter but not inside the table of content.

To add an unnumbered chapter inside the table of content, you could use one of the macros:

\begin{verbatim}
\chaptertoc[toctitle]{title}
\chaptertoc*[toctitle]{title}
\end{verbatim}

The macros \texttt{\chaptertoc{}} and \texttt{\chaptertoc*{}} have the same effect except that \texttt{\chaptertoc*{}} aligns the chapter’s title to the other numbered chapters’ titles; and \texttt{\chaptertoc{}} not.

5.22.3/ **Non-numbered Section in Table of Content**

If you want to add a document section that has no a section number but appearing inside the table of content, the classical \LaTeX\ macros \texttt{\section{}} and \texttt{\section*{}} are inefficient. Indeed, \texttt{\section{}} add a numbered section inside the table of content, and \texttt{\section*{}} adds an unnumbered section but not inside the table of content.

To add an unnumbered section inside the table of content, you could use one of the macros:

\begin{verbatim}
\sectiontoc[toctitle]{title}
\sectiontoc*[toctitle]{title}
\end{verbatim}

The macros \texttt{\sectiontoc{}} and \texttt{\sectiontoc*{}} have the same effect except that \texttt{\sectiontoc*{}} aligns the section’s title to the other numbered sections’ titles; and \texttt{\sectiontoc{}} not.

5.22.4/ **Non-numbered Subsection in Table of Content**

If you want to add a document subsection that has no subsection number but appearing inside the table of content, the classical \LaTeX\ macros \texttt{\subsection{}} and \texttt{\subsection*{}} are inefficient. Indeed, \texttt{\subsection{}} is adding a numbered subsection inside the table of content, and \texttt{\subsection*{}} is adding an unnumbered subsection but not inside the table of content.

To add an unnumbered subsection inside the table of content, you could use one of the macros:

\begin{verbatim}
\subsectiontoc[toctitle]{title}
\subsectiontoc*[toctitle]{title}
\end{verbatim}

The macros \texttt{\subsectiontoc{}} and \texttt{\subsectiontoc*{}} have the same effect except that \texttt{\subsectiontoc*{}} aligns the subsection’s title to the other numbered subsubsections’ titles; and \texttt{\subsectiontoc{}} not.

5.22.5/ **Non-numbered Subsubsection in Table of Content**

If you want to add a document subsubsection that has no subsubsection number but appearing inside the table of content, the classical \LaTeX\ macros \texttt{\subsubsection{}} and \texttt{\subsubsection*{}} are inefficient. Indeed, \texttt{\subsubsection{}} is adding a numbered subsubsection inside the table of content, and \texttt{\subsubsection*{}} is adding an unnumbered subsubsection but not inside the table of content.

To add an unnumbered subsubsection inside the table of content, you could use one of the macros:

\begin{verbatim}
\subsubsectiontoc[toctitle]{title}
\subsubsectiontoc*[toctitle]{title}
\end{verbatim}

The macros \texttt{\subsubsectiontoc{}} and \texttt{\subsubsectiontoc*{}} have the same effect except that \texttt{\subsubsectiontoc*{}} aligns the subsubsection’s title to the other numbered subsubsections’ titles; and \texttt{\subsubsectiontoc{}} not.
5.22. ADDITIONAL DOCUMENT SECTIONING MACROS

5.22.6/ CHAPTER with different labels in TOC, HEADERS and DOCUMENT

If you want to control the labels in the table of contents (TOC), the headers and the document for a chapter, the classical \LaTeX macros \texttt{\chapter} and \texttt{\chapter*} are inefficient.

To control these labels, you could use the macro:
\texttt{\chapterfull\{toctitle\}\{title\}\{headertitle\}}

The macro create a chapter with the given label \texttt{title} in the core part of the document, with the given label \texttt{toctitle} in the table of contents, and with the label \texttt{headertitle} in the headers.

5.22.7/ SECTION with different labels in TOC, HEADERS and DOCUMENT

If you want to control the labels in the table of contents (TOC), the headers and the document for a section, the classical \LaTeX macros \texttt{\section} and \texttt{\section*} are inefficient.

To control these labels, you could use the macro:
\texttt{\sectionfull\{toctitle\}\{title\}\{headertitle\}}

The macro create a section with the given label \texttt{title} in the core part of the document, with the given label \texttt{toctitle} in the table of contents, and with the label \texttt{headertitle} in the headers.
The package upmethodology-document provides base functions to manage document information (project, subproject, authors...).

6.1/ DOCUMENT INFORMATION AND DECLARATION

The informations associated to an UP document are:

- \texttt{theupmproject} is the name of the project for which the document was produced;
- \texttt{theupmsubproject} is the name of the sub-project for which the document was produced;
- \texttt{theupmdocname} is the name of the document;
- \texttt{theupmdocref} is the reference number of the document;
- \texttt{theupmfulldocname} is the complete name of the document (composing by the project, subproject and name of the document).

You could declare the information about your document with one of the following functions:
\texttt{declaredocument[project][name][ref]}
\texttt{declaredocumentex[project][subproject][name][ref]}
where the parameters are:

- \texttt{project} is the name of the project the document belongs to;
- \texttt{subproject} is the name of the sub-project the document belongs to;
- \texttt{name} is the name of the document;
- \texttt{ref} is the reference number of the document.

6.2/ ABSTRACT AND KEY-WORDS

You are able to declare the abstract and the key-words for your document. Both are basically used by the back page package.
6.2.1/ Declarations

The macro `\setdocabstract` is for entering the document’s abstract:
```
\setdocabstract[lang]{abstract_text}
```
where `abstract_text` is the text of your abstract and `lang` designates for which language the abstract text is for. If the language is not specified, this macro uses the current document language.

The macro `\setdockeywords` is for entering the document’s key-words:
```
\setdockeywords[lang]{keywords}
```
where `keywords` is the list of key-words and `lang` designates for which language the key-words are for. If the language is not specified, this macro uses the current document language.

6.2.2/ Rendering

The macro `\theupmdocabstract` is expanded with the abstract text:
```
\theupmdocabstract
```

The macro `\theupmdockeywords` is expanded with the key-words:
```
\theupmdockeywords
```

6.3/ Document Summary

You can obtain a document summary with the macro `\upmdocumentsummary[width]` which produces:

<table>
<thead>
<tr>
<th>Document Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
</tr>
<tr>
<td>Document</td>
</tr>
<tr>
<td>Reference</td>
</tr>
<tr>
<td>Version</td>
</tr>
<tr>
<td>Last Update</td>
</tr>
</tbody>
</table>

6.4/ Change Icons

By default, this package uses the logo of *Arakhnē* as icons. You could change them with the macros:

- `\defupmsmalllogo{filename}` defines the small logo used in the headers for instance;
- `\defupmlogo{filename}` defines the logo used on the front page for instance.

The logos’ filenames are accessible with the functions `\theupmsmalldoclogo` and `\theupmdoclogo`.

6.5/ Document Authors

An author is someone who participates to the writing of the document. You could register author identities with:

```
\addauthor[\email]{firstname}{name}
\addauthor*[\email]{firstname}{name}{comment}
\addauthorvalidator[\email]{firstname}{name}
\addauthorvalidator*[\email]{firstname}{name}{comment}
```

The list of the authors is accessible by two means:
6.6. DOCUMENT VALIDATORS

- \authorlist is a comma-separated list of the authors’ names;
- \upmdocumentauthors produces an array of all the authors (see below for an example).

<table>
<thead>
<tr>
<th>Authors</th>
<th>Comments</th>
<th>Emails</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stéphane Galland</td>
<td>Original Author</td>
<td><a href="mailto:galland@arakne.org">galland@arakne.org</a></td>
</tr>
<tr>
<td>Frans Van Dunné</td>
<td>Reviewer</td>
<td></td>
</tr>
</tbody>
</table>

You could test if a string is the name of the author with:

- \ifdocumentauthor\lowercasename\then\else; the first parameter must be lower case. If the \lowercasename is the name of one of the authors, then the \then clause is expanded, otherwise the \else clause is expanded.

<table>
<thead>
<tr>
<th>Authors</th>
<th>Comments</th>
<th>Emails</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stéphane Galland</td>
<td>Original Author</td>
<td><a href="mailto:galland@arakne.org">galland@arakne.org</a></td>
</tr>
<tr>
<td>Frans Van Dunné</td>
<td>Reviewer</td>
<td></td>
</tr>
</tbody>
</table>

6.6/ DOCUMENT VALIDATORS

A validator is someone who participates to the validation of the document. You could register validator identities with:
\addvalidator\email\firstname\name
\addvalidator*\email\firstname\name\comment
\addauthorvalidator\email\firstname\name
\addauthorvalidator*\email\firstname\name\comment

The list of the validators is accessible by two means:

- \validatorlist is a comma-separated list of the validator’s names;
- \upmdocumentvalidators produces an array of all the validators (see below for an example).

<table>
<thead>
<tr>
<th>Validators</th>
<th>Comments</th>
<th>Emails</th>
<th>Initials</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stéphane Galland</td>
<td>Original Author</td>
<td><a href="mailto:galland@arakne.org">galland@arakne.org</a></td>
<td></td>
</tr>
</tbody>
</table>

6.7/ INFORMED PEOPLE

An informed person is someone who receives the document to be informed about its content. You could register informed people identities with:
\informed\email\firstname\name
\informed*\email\firstname\name\comment

The list of the informed people is accessible by two means:

- \informedlist is a comma-separated list of the informed people’s names;
- \upmdocumentinformedpeople produces an array of all the informed people (see below for an example).
6.8/ Copyright and Publication Information

Package `upmethodology-document` provides several macros to define the copyright owner and the publication informations required to generate a publication page.

6.8.1/ Setting Information

The Copyright holder(s) are person(s) or institution(s), that own the copyright on the document. The following macro allows you to set the identity of the copyright holder in all parts of the documents:

\setcopyrighter{name}

Publisher is the people or the institution, or both, which is publishing the document. Basically it is the same the copyrighter (see above):

\setpublisher{name}

Some times, copyright laws depend on the location where the document is printed. The following macro allows you to put a message in the publication page which is indicating where the document is printed:

\setprintingaddress{address}

Publications may be identifier by international identifiers. Package `upmethodology-document` supports ISBN, ISSN and DOI:

\setisbn{number}

\setissn{number}

\setdoi{number}

6.8.2/ Retrieving Information

The information set by the macros described in the previous section may be retrieved with the following macros:

\theupmcopyrighter
\theupmpublisher
\theupmprintedin
\theupmisbn
\theupmissn
\theupmdoi

6.8.3/ Publication Page

The package `upmethodology-document` provides the `\upmpublicationpage` macro which is displaying a empty page with publication informations and optionally set the page number (default value is −1). Figure 6.1 illustrates the publication page of this document.

6.9/ Localization

The current language is defined in the macro `\upmcurrentlang`.

For testing the current language, you could use the macro `\ifuplang{lang_id}{then macros}{else macros}`. This macro tests if the given `lang_id` corresponds to the value expended by the macro
\upmcurrentlang. If it is true, the macros specified in the "then macros" are expanded. Otherwise, the macros specified in the "else macros" are expanded.

The following macros defines some localized strings used by \upmethodology\-document:

- \upm@lang@project: Project;
- \upm@lang@document: Document;
- \upm@lang@docref: Reference;
- \upm@lang@lastupdate: Last Update;
- \upm@lang@document@summary: Document Summary;
- \upm@lang@document@authors: Authors;
- \upm@lang@document@validators: Validators;
- \upm@lang@document@names: Names;
- \upm@lang@document@emails: Emails;
- \upm@lang@document@initials: Initials;
- \upm@lang@document@abstract: Abstract;
- \upm@lang@document@keywords: Key-words.
This document describes the \LaTeX\ Packages for Unified Process Methodology project.

\LaTeX\ is a trademark of the American Mathematical Society.
tex-upmethodology is owned by Stéphane Galland, Arakhné.org, France.

This document was realised with \LaTeX\ and tex-upmethodology.
Copyright © 2017-2019 Stéphane GALLAND.

This document is published by the Arakhné.org Group. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publishers.

Reference : UPM-2019-01

Figure 6.1: Example of Publication Page generated with \texttt{\upmpublicationpage}
7

PACKAGE UPMETHODOLOGY-FRONTPAGE

Version: 2017/08/08

The upmethodology-frontpage package provides a front page for the UP documents. This package does not provide any public function. It is based on all the previous packages.

7.1/ DISPLAY THE FRONT PAGE

The front cover is displayed by invoking one of the following macros:
\maketitle
\makefrontcover

7.2/ CHANGE FRONT PAGE LAYOUT

It is possible to change the layout of the front page with the macro:
\setfrontlayout[layout_name]
where layout_name must be one of:

- classic: classic front page layout with title and logo;
- modern: front page layout with title and logo and background picture.

The figure 7.1 illustrates the different layouts.

7.3/ CHANGE ILLUSTRATION PICTURE

It is possible to insert an illustration picture on the front page. You could specify the image with the macro:
\setfrontillustration[width_factor][filename]
where:

- width_factor is the scaling factor of the picture according to the line width. If you specifies 1 the image will not be scaled, for .5 the image will be the half of its original width...
- filename is the name of picture to use as the illustration.
7.4/ **Define a Front Page in Extensions**

The `upmethodology-frontpage` package is able to use a page layout defined in a document extension (see chapter 9 for details on document extension).
A \LaTeX{} macro must be defined in the \texttt{upmext-NAME.cfg} file of the extension. The name of this macro (for example \texttt{mylayout}) must be set with the \texttt{\set macro} in the same file: \texttt{\set \{frontpage\} \{mylayout\}}

### 7.5/ Localization

The following macros defines some localized strings used by \texttt{upmethodology-frontpage}:

- \texttt{\upm@lang@front@authors}: Authors;
The package upmethodology-backpage provides a back page for the UP documents. This package does not provide any public function. It is based on all the previous packages.

8.1/ Display the back page

The back cover is displayed by invoking the following macro:
\makebackcover

8.2/ Change back page layout

It is possible to change the layout of the back page with the macro:
\setbacklayout{layout_name}
where layout_name must be one of:
- none: no back page.

8.3/ Small text before the back page

It is possible to insert a text at the bottom of the page just before the back page (usually the inner page of the cover for a two sided document). You must set the macro \backcovermessage with the \Set macro:
\Set{backcovermessage}{text}

8.4/ Define a back page in extensions

The upmethodology-backpage package is able to use a page layout defined in a document extension (see chapter 9 for details on document extension).
A \LaTeX macro must be defined in the upmext-NAME.cfg file of the extension. The name of this macro is backpage, and it must be set with the \Set macro in the same file:
\Set{backpage}{\LaTeX macros}
9

Package upmethodology-extension

Version: 2017/08/08

The package upmethodology-extension provides tools to create layout and rendering extensions. It is possible to write an extension to the upmethodology-document package. An extension is able to override several values from the default upmethodology-packages or may be used by the other suite’s packages. For example, the Systems and Transport laboratory\\[1](11) extension is providing laboratory’s icons, publisher’s name and page layouts.

9.1/ Load a Document Extension

To load and use a document extension, you must invoke the macro:
\UseExtension{extension_name}
where extension_name is the identifier of the extension to load. The extension’s files must be inside your \LaTeX{} search path.

9.2/ Write a Document Extension

A document extension could be written and described inside a file named upmext-NAME.cfg, where NAME is the name of the extension. This file must be put in your \LaTeX{} search path.

The upmext-NAME.cfg file is a \LaTeX{} file in which a set of definition macros are put. These macros must respect the \LaTeX{} syntax.

The \DeclareCopyright macro enables you to declare additional copyright information about the extension:
\DeclareCopyright[lang]{extension_name}{year}{copyrighter}{trademark and copyright information}

This macro declares the copyright value which contains the copyright text (for this documentation \"\").

This macro also declares the trademarks value which contains the trademark and other related informations about the extension (for this documentation \"\").

Additional macros are provided to redefine the upmethodology-document constants:
\Set[lang]{variable_name}{value}

The variable_name is the name of the value to override. It must be taken in one of the names listed in table [7]. The lang parameter is a language identifier. It is used to restrict the definition to a specific language. If not given, the default language is used instead. The image_name and image_scale are the name of the image file and the scaling factor respectively.
<table>
<thead>
<tr>
<th>Value Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>logo</td>
<td>the filename of the picture which must be used as a large logo.</td>
</tr>
<tr>
<td>smalllogo</td>
<td>the filename of the picture which must be used as a small logo.</td>
</tr>
<tr>
<td>copyrighter</td>
<td>the name of the authors or the institution which own the copyright on the document.</td>
</tr>
<tr>
<td>publisher</td>
<td>the name of the document’s publisher. The <code>\lang</code> parameter is supported.</td>
</tr>
<tr>
<td>printedin</td>
<td>the location/address where this document is printed.</td>
</tr>
<tr>
<td>frontillustration</td>
<td>the image to use as illustration. The <code>\lang</code> parameter is ignored.</td>
</tr>
<tr>
<td>frontpage</td>
<td>the name of the front page style — not the <code>\LaTeX</code> macros — to layout the front page. OR the front page illustration.</td>
</tr>
<tr>
<td>backpage</td>
<td>the <code>\LaTeX</code> macros to layout the back page. OR the back page illustration.</td>
</tr>
<tr>
<td>cfrontpage</td>
<td>the <code>\LaTeX</code> macros — not the name of the front page style — to layout the front page.</td>
</tr>
</tbody>
</table>

Table 9.1: List of overiddable value names

The `\Get` macro allows you to retrieve the value defined by a `\Set`:

\Get{variable_name}

The `\Unset` macro allows you to remove the definition of a value:

\Unset{variable_name}

The `\Ifnotempty` macro allows you to expand the `\LaTeX` macros if the given text is not empty:

\Ifnotempty{text}{\latex_code}

The `\Ifempty` macro allows you to expand the `\LaTeX` macros if the given text is empty:

\Ifempty{text}{\latex_code}

The `\Ifelsedefined` macro allows you to expand the `\LaTeX` macros in `then code` if a value with the given name was defined, or to expand the `\LaTeX` macros in `else_code` if no value with the given name was defined:

\Ifelsedefined{value_name}{\then_code}{\else_code}

The `\Put` macro is an extension of the standard picture `\put` macro. It takes into account the joint margin applied in two sided documents when it is used on page’s backside (eg. the back page of the document):

\Put{x,y}{macros}

This macro must be used inside a `\picture` environment in place of the standard `\put` macro.
The \LaTeX package upmethodology-task provides a set of macros to define project’s tasks. During \LaTeX compilation this package could log the message “Project Task(s) may have changed. Rerun to get cross-references right” when some task information was not found or due to cross-references on them.

10.1/ Task Definition

The definition of a task could be made only inside one of the following environments:
\begin{taskdescription}[id]\end{taskdescription}
\begin{taskdescription*}[id]\end{taskdescription*}
where id is the identifier of the task.

The environment taskdefinition displays the task’s description with a call to \texttt{\thetaskdescription[id]}. On the other hand, taskdefinition* never displays the task’s description.

Inside one of the task’s definition environment above, you could use one of the following macros to define the task’s attributes:

- \texttt{\taskname[name]} to define the name of the task;
- \texttt{\tasksuper[id]} indicates that the current task is a sub-task of the task identified by the given identifier;
- \texttt{\taskcomment[text]} to describe the task’s purposes and goals (will be shown in the description box of the task’s description);
- \texttt{\taskprogress[percent]} to set the percentage for task achievement;
- \texttt{\taskstart[date]} to set the starting date of the task (real or predicted);
- \texttt{\taskend[date]} to set the finished date of the task (real or predicted);
- \texttt{\taskmanager[name]} to add a task’s manager into the list of the managers;
- \texttt{\taskmember[name]} to add a task’s member into the list of the members;
• \taskmilestone[\text{date}][\text{comment}]
  to add a milestone into the task for the given date and described by the given comment.

10.2/ Task Reference

You could reference any information about the defined tasks in your document. In case you used cross-references this package could log the message “Project Task(s) may have changed. Rerun to get cross-references right” to complain about rebuilding of our document.

The following macros are available:

• \thetasksuper[\text{id}]
  replies the identifier of the parent task corresponding to the task identified by \text{id};

• \thetaskname[\text{id}]
  replies the name of the the task identified by \text{id};

• \thetaskcomment[\text{id}]
  replies the description for the the task identified by \text{id};

• \thetaskprogress[\text{id}]
  replies the achieving percent for the the task identified by \text{id};

• \thetaskstart[\text{id}]
  replies the starting date for the the task identified by \text{id};

• \thetaskend[\text{id}]
  replies the ending date for the the task identified by \text{id};

• \thetaskmanagers[\text{id}]
  replies the managers' list for the the task identified by \text{id};

• \thetaskmembers[\text{id}]
  replies the members' list for the the task identified by \text{id};

• \thetaskmilestones[\text{id}]
  replies the list of milestone’s dates for the the task identified by \text{id};

• \thetaskmilestonecomment[\text{id}][\text{date}]
  replies the comment of the given milestone for the the task identified by \text{id};

• \thetaskdescription[\text{width}][\text{id}]
  replies the complete description of the the task identified by \text{id}.

10.3/ Localization

The following macros defines some localized strings used by upmethodology-task:

• \upm@task@lang@task: Task;

• \upm@task@lang@escription: Description;

• \upm@task@lang@startat: Start at;

• \upm@task@lang@endat: End at;

• \upm@task@lang@archieved: Achieved;

• \upm@task@lang@managers: Managers;
10.3. LOCALIZATION

- \texttt{upm@task@lang@members}: Members;
- \texttt{upm@task@lang@Milestones}: Milestones;
- \texttt{upm@task@lang@subtask}: Sub-task of.
The \LaTeX\ package \texttt{upmethodology-code} provides a set of macros for source code formatting. The supported source codes are UML, Java and C++.

You could load the package with the following options:

\begin{verbatim}
\uml \ use the UML notation (default value)
\java \ use the Java notation
\cpp  \ use the C++ notation
\end{verbatim}

You could also change the notation language with the macro:
\texttt{\textbackslash upm\textbackslash codelang\{upm\}|java|cpp}\textbackslash

The provided macros are listed in the following table:

\begin{tabular}{|l|l|l|l|}
\hline
macro & UML & Java & C++ \\
\hline
\texttt{\jclass[TheClass]} & \texttt{\textsc{TheClass}} & \texttt{\textsc{TheClass}} & \texttt{\textsc{TheClass}} \\
\texttt{\jinterface[TheInterface]} & \texttt{\textsc{TheInterface}} & \texttt{\textsc{TheInterface}} & \texttt{\textsc{TheInterface}} \\
\texttt{\jpackage[ThePackage]} & \texttt{\textsc{ThePackage}} & \texttt{\textsc{ThePackage}} & \texttt{\textsc{ThePackage}} \\
\texttt{\jfunc[FunctionName]} & \texttt{\textsc{FunctionName}} & \texttt{\textsc{FunctionName}} & \texttt{\textsc{FunctionName}} \\
\hline
\hline
\texttt{\jclass} & \texttt{\textsc{class}} & \texttt{\textsc{Class}} & \texttt{\textsc{class}} \\
\texttt{\jvoid}  & \texttt{\textsc{void}}  & \texttt{\textsc{void}}  & \texttt{\textsc{void}}  \\
\texttt{\jboolean} & \texttt{\textsc{boolean}} & \texttt{\textsc{boolean}} & \texttt{\textsc{bool}}   \\
\texttt{\ jint}  & \texttt{\textsc{integer}} & \texttt{\textsc{int}}   & \texttt{\textsc{int}}   \\
\texttt{\jlong} & \texttt{\textsc{long integer}} & \texttt{\textsc{long}} & \texttt{\textsc{long}} \\
\texttt{\jfloat} & \texttt{\textsc{float}} & \texttt{\textsc{float}} & \texttt{\textsc{float}} \\
\texttt{\jdouble} & \texttt{\textsc{double}} & \texttt{\textsc{double}} & \texttt{\textsc{double}} \\
\texttt{\jchar} & \texttt{\textsc{character}} & \texttt{\textsc{char}} & \texttt{\textsc{char}} \\
\texttt{\jstring} & \texttt{\textsc{string}} & \texttt{\textsc{string}} & \texttt{\textsc{string}} \\
\texttt{\jarray[T]} & \texttt{\textsc{array of Ts}} & \texttt{\textsc{Collection <T>}} & \texttt{\textsc{std::vector <T>}} \\
\texttt{\jcollection[T]} & \texttt{\textsc{collection of Ts}} & \texttt{\textsc{Set <T>}} & \texttt{\textsc{std::set <T>}} \\
\texttt{\jset[T]} & \texttt{\textsc{set of Ts}} & \texttt{\textsc{Set <T>}} & \texttt{\textsc{std::set <T>}} \\
\hline
\end{tabular}
### Constants

<table>
<thead>
<tr>
<th>Macro</th>
<th>UML</th>
<th>Java</th>
<th>C++</th>
</tr>
</thead>
<tbody>
<tr>
<td>jtrue</td>
<td>TRUE</td>
<td>TRUE</td>
<td>TRUE</td>
</tr>
<tr>
<td>jfalse</td>
<td>FALSE</td>
<td>FALSE</td>
<td>FALSE</td>
</tr>
</tbody>
</table>

### Operations

<table>
<thead>
<tr>
<th>Macro</th>
<th>UML</th>
<th>Java</th>
<th>C++</th>
</tr>
</thead>
<tbody>
<tr>
<td>jcode{source code}</td>
<td>source code</td>
<td>source code</td>
<td>source code</td>
</tr>
<tr>
<td>jcall{fct}(params)</td>
<td>fct(params)</td>
<td>fct(params)</td>
<td>fct(params)</td>
</tr>
<tr>
<td>jop{operator}</td>
<td>operator</td>
<td>operator</td>
<td>operator</td>
</tr>
</tbody>
</table>
Copyright © 2017-2019 Stéphane Galland

This program is free library; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 3 of the License, or any later version.

This library is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this library; see the file COPYING. If not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.