The *Toptesi* package

Typesetting a PhD thesis with *LaTeX*

Claudio Beccari

Politecnico di Torino

[GU]IT meeting 2019
The purpose of this tutorial is not to teach how to use \LaTeX.

It is assumed the the attendants have already a sufficient acquaintance with this language and are already capable of processing their .\text{tex} files with any of the T\TeX{} system typesetting engines \texttt{pdflatex}, \texttt{lualatex} and, maybe, \texttt{xelatex}.
Plan of this tutorial

In this tutorial we examine how to typeset a PhD thesis that fulfils the specifications issued by the ScuDo doctoral School of Politecnico di Torino.

The *TOPtesi* bundle was created to typeset any kind of final work report, especially for Politecnico di Torino (bachelor, master, and PhD theses) and other final reports not only for this University.

Here we enter into the details of the ScuDo doctoral theses.
The *TOPtesi* bundle is a collection of class and extension packages that may configure the structure and the title page of each report.

The various theses have different requirements for their very nature, and for the specifications issued by the various universities. Most of these university specifications concern what has to appear on the title page; the contents structure is more related to the thesis type.
The *toptesi* bundle has two different documentation files that are already part of the \TeX{} system installation; the user can read them with the commands (issued in a command line window): either

```
texdoc toptesi.pdf
```

or

```
texdoc toptesi-it
```

They are complementary to one another; the first is completely in English and documents also the code; the second is partially in Italian and partially in English and deals mostly on how to write a customised thesis with *toptesi*.
Invoking the \textit{toptesi} class

The thesis \texttt{.tex} source main file shall start by declaring the document class \texttt{toptesi}.

The start of the main source file

\begin{verbatim}
\documentclass[\langle options\rangle]{toptesi}
\end{verbatim}
Many options may be specified among which the following are the most important.

- **corpo**=(normal font size)
- **twoside**
- **tipotesi**=(thesis type)

The default font size is 10 pt; the user may chose any, even fractional, size but it is better not to exceed 12.5 pt or, exceptionally, 13.5 pt.

For the ScuDo PhD theses the *(thesis type)* is, very predictably: **scudo**.
Next comes the metadata list: such information is necessary in order to produce an ISO compliant archivable PDF file.

We say more about metadata from frame 61 on, but since we are talking about the preamble of the thesis main file, here is where we should put the metadata; each one is preceded by a descriptor that in general is self explanatory. But the important point is that they must be inserted within a special environment.

**Metadata environment**

```latex
\begin{filecontents*}{\jobname.xmpdata}
<metadata>
\end{filecontents*}
```
If we want to produce an ISO compliant PDF archivable file we have to typeset the thesis using the \texttt{lualatex} typesetting engine but we have to load before anything else the special package \texttt{pdfx} with its option \texttt{a-1b}:

\begin{verbatim}
ISO PDF compliant package
\usepackage[a-1b]{pdfx}
\end{verbatim}

This package provides the many other metadata required by ISO regulations; it loads the \texttt{hyperref} package with a specific option to guarantee that hyperlinks are ISO compliant; it also patches many commands to be compliant with the ISO regulations. The fact that \texttt{hyperref} is loaded by \texttt{pdfx} implies that we cannot load it again with our preferred options, otherwise we get an Option Clash error.
We have to specify the fonts to use while typesetting with \texttt{lualatex}. By default it uses the Latin Modern ones together with the corresponding Latin Modern Math fonts. Here we make an example with the Libertinus font collection.

\begin{Verbatim}
\usepackage{fontspec}
\defaultfontfeatures{Ligatures=TeX}
\setmainfont{Libertinus Serif}
\setsansfont{Libertinus Sans}
\setmonofont{Libertinus Mono}\%
[Scale=MatchLowercase]
\usepackage{unicode-math}
\setmathfont{Libertinus Math}
\end{Verbatim}
If the thesis contains an index, remember that package *imakeidx* has already been loaded by the class; you should not reload it. But in order to make it work for collecting the information you want to index, you have to turn on this feature by using the \makeindex command.

Since *imakeidx* can collect material to produce more than one index, each index must be activated with its own \makeindex command.

### Configuring \makeindex

\makeindex[name=\textit{index name},
title=\textit{index title},
intoc,
\langle other options\rangle]
Collecting material to index

Of course in the body of the thesis it is necessary to mark which words or short phrases are to be sent to which index.

For this purpose the `\index` command requires, besides the material to be sent to one of the indices, the `<index name>` information with this syntax:

```
The \texttt{\index} command
\index[<index name>]{<lemma for the index>}
```
The words or short phrases collected by the `\index` commands are sent to a raw index file that needs to be properly processed by a program named `makeindex`, not to be confused with the command `\makeindex`.

Without using the `imakeidx` package, this processing should be done by hand by the user through suitable commands written in the command line window.

Thanks to the properties of the TeX system typesetting programs, such operation is done with the `-shell-escape` functionality of `imakeidx` so that the user should not care to run again and again the typesetter, then `makeindex`, then the typesetter.

Each time the user typesets the thesis, the indices get automatically updated.
The `toptesi` class with the `tipotesi=scudo` option by default produces the bibliography with the style of the IEEE Transactions; this is a sober style with numeric reference and citation schemes.

The user should only create a bibliographic database in a special format, See frames from 56 on; then the user should run the program `biber` that extracts from the database only the records that have been referenced/cited within the thesis. If the thesis body contains the command `\nocite{*}`, all the records, even if they have not been referenced, will be extracted and typeset in the thesis bibliography.

Nothing else is required.
Customised bibliography

For theses that deal with disciplines that have nothing to do with the publications of the engineering profession institutions, it is possible to customise the bibliography the way the user prefers. To do so it is necessary to do the following.

1. The *mybibliography* option should be added to the class options.

2. The settings to customise the bibliography should be specified by the user.

The example file *toptesi-scudo-example.tex* shows an example, but the user is completely free to do what s/he likes best. It is his/her responsibility to find the proper information by reading the pertinent documentation and by experimenting.
The `hyperref` package may be invoked again without any option in order to avoid the Option clash error. It may be done for example in this way.

Customising `hyperref`

```latex
\usepackage{hyperref}
\hypersetup{%
    pdfpagemode={UseOutlines},
    bookmarksopen,
    pdfstartview={FitH},
    colorlinks,
    linkcolor={blue},
    citecolor={blue},
    urlcolor={blue}}
```
Before issuing the command `\begin{document}` the user may load other packages and set other settings.

Actually there should be nothing else to do; but every thesis is a *unicum*, therefore it possible that the default `tipotesi=scudo` settings are insufficient for certain theses.

The user is only warned to avoid any package or any setting that changes the thesis graphical design; its design may or may not encounter the user’s aesthetical sensibility, but it is the design decided by the ScuDo doctoral school, therefore it is sort of a brand.

At the same time the user must remember that the thesis is valid for its contents, not for its typographical design.
After the command \texttt{\begin{document}} the thesis body begins.

In general it is recommended to split the thesis body in several major chunks; for example: the front matter (title and legal page, table of contents, lists of tables and of figures, and the like), the single chapters and numbered appendices, the back matter (one unnumbered appendix, bibliography, possible glossaries or nomenclature lists, possible indices, and the like).

These major chunks may be conditionally typeset one at the time if before \texttt{\begin{document}} the \texttt{\includeonly} list is specified.
The code to specify the `\includeonly` list should look like this.

```latex
\includeonly{%
  Chapter1/chapter1,%
  Chapter2/chapter2,%
  Chapter3/chapter3,%
  Appendix1/appendix1,%
  Appendix2/appendix2,%
}
```

The folders should all be subfolders of the one where the thesis main file has been saved.

Of course the folders names and the files names might be more meaningful.
The thesis starts with the front matter; the first instance of a starred or unstarred `chapter` command automatically starts the main matter.

\begin{ThesisTitlePage}
<title page data>\end{ThesisTitlePage}

<special pages>
\tablespagetrue% optional
\figurespagetrue% optional
\allcontents
The \textit{special pages} are explained in a few slides. Both commands \texttt{\tablespagetrue} and \texttt{\figurespagetrue} are optional switches that control if the following command \texttt{\allcontents} should generate also the list of tables and the list of figures.
The *special pages* are the optional ones for the acknowledgements and the dedication. If they should start on an odd page preceded by a blank page, they should be inserted after a `\emptypage` command.

**Acknowledgements and dedication**

```
\emptypage\acknowledgements
I would like to acknowledge ...
\emptypage
\begin{dedication}
I would like to dedicate ...
\end{dedication}
```
Acknowledgements and dedication

The **acknowledgements** are required only if some support or assistance has been obtained from people outside Politecnico, otherwise they are completely superfluous. In particular it is not correct to acknowledge the support of any university tutor or supervisor: they just acted as institutional officers.

The **dedication** is always superfluous, at least in Italy, but it is not forbidden to dedicate the thesis to someone. In some other countries the dedication is almost mandatory.
In the ScuDo doctoral thesis you may specify up to two university logos.

The university logos

\PhDs@schoolLogo{\% ScuDoLogo, \% mandatory \%
OtherLogo \% optional
}\

It is possible to chose among three ScuDo logos, but the user must fetch them in the special site of the ScuDo Web site dedicated to PhD students.

The other optional logo is the one of Università di Torino, or of the INRIM, or of the INFN institutions, and only in case of theses developed in conjunction with one of these external research institutions.
Next come the identification data.

Identification and personal data

\ProgramName{\langle\textit{discipline}\rangle}
\CycleNumber{\langle\textit{cycle ordinal number}\rangle}
\author{\langle\textit{your name}\rangle}
\title{\langle\textit{Thesis title}\rangle}
\subtitle{\langle\textit{Thesis subtitle}\rangle}

The \langle\textit{discipline}\rangle is the official name of the particular scientific area your thesis is on.

Every PhD program is ordered in “cycles”; it is needed to specify the cycle ordinal number, but no specific format is required; you may chose the uppercase roman numeral without superscript (XXV), the Arabic numeral with superscript (35\textsuperscript{o}), the German style ordinal (35.o), the English style (35th).
The supervisor(s) number must be indicated together with their list:

\SupervisorNumber\{\langle number\rangle\}
\SupervisorList\{\langle
Prof. \langle name(s) surname(s)\rangle \ Supervisor\}
Prof. \langle name(s) surname(s)\rangle \ Co-supervisor\}\}

If \langle number\rangle equals zero, no label is printed over the list of names; if it equals 1 a singular label is used and just one supervisor must be indicated; if it is greater than 1 a plural label is used.
The ScuDo title page may contain the examiners list and other data.

**Final title page data**

\ExaminerList{%
  Prof. \textit{name(s) surname(s)}, Referee, University of . . . \ \ \\
  Prof. \textit{name(s) surname(s)}, Referee, University of . . . \ \ \\
  Prof. \textit{name(s) surname(s)}, University of . . . \ \ \\
  . . . \ \\
  Prof. \textit{name(s) surname(s)}, University of . . . }

\ExaminationDate{\textit{date}}

The \textit{date} may have any format, but the month full name is preferred.
The Scudo thesis format requires a copyright or legal page that contains both the copyright notice and the label of the Creative Commons licence that covers the thesis contents.

This page contains also a disclaimer where the author declares and undersigns in original that the contents is his/her personal work and that s/he paid all possible attention to attribute the citations to the original authors.
Please read carefully the disclaimer text; you need to undersign it, therefore you are going to declare that what you signed is true.

So please, double check that every quotation is correctly attributed; that any material that you copied and pasted into your source files, is correctly attributed. This includes any picture you got from the Internet; before including that picture, control its license and, if possible, discover the original author and correctly cite him/her. And so on.

Missing to do so means infringing the law, and it may cause you more trouble than you can imagine.
Writing your Doctoral Thesis with \LaTeX

This document is an example of what you can do with the TOPtesi class

Mario Rossi

Supervisors
Prof. A.B., Supervisor
Prof. C.D. Co-supervisor

Doctoral Examination Committee:
Prof. A.B., Referee, University of ...
Prof. C.D., Referee, University of ...
Prof. E.F., University of ...
Prof. G.H., University of ...
Prof. I.J., University of ...

Politecnico di Torino
February 29, 2123

This thesis is licensed under a Creative Commons License, Attribution-Noncommercial-NoDerivative Works 4.0 International: see www.creativecommons.org. The text may be reproduced for non-commercial purposes, provided that credit is given to the original author.

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

 Mario Rossi
 Turin, February 29, 2123
The body of the main file continues with a “summary”, which is not an abstract; it may contain several pages to be printed separately so that, even if the examiners received a printed copy or a PDF file of the thesis, it might be useful for them to have also an extended summary.

At the same time this summary might be visible by the librarians so that when the archived copy is looked for, the interested persons can read what the thesis is really about: the title and the subtitle might not be enough.

Summary

\summary
\begin{summarytext}
The main matter of the thesis may be started explicitly with the command `\mainmatter`. But any starred or unstarred `\chapter` command switches the thesis compilation settings to those requested by the main matter.

In particular the folios restart from 1 if the front matter had roman folios; the class option `romannumerals` sets this unusual style, that was very common when typesetters worked with metal types, but now it is very outdated. Chapters start only on the odd pages, sectional divisions are all numbered until the level of subsections and are all listed in the table of contents.
In the main matter every sectional division should be numbered. It is not prohibited to create an introductory or a conclusive short unstructured chapter, but it creates some problems to the user, because s/he cannot use the `\chapter` command, not even the starred version. In facts the starred version inhibits the chapter numeration, but it does not send any information either to the table of contents, or to the page headers.
If the user wants absolutely to introduce the thesis with an unnumbered chapter and/or wants to conclude it with an unnumbered chapter s/he has to use the following code.

```
\chapter*{\textit{title}}
\addtocontents{toc}{0}{{\textit{title}}}
\markboth{\textit{title}}{\textit{title}}
```

The paper on the use of *TOPtesi* in this *GU It* Meeting proceedings shows a possibly smarter way to achieve the same result.
At this point, having specified in the preamble the files to be conditionally typeset, it is necessary to list what to include. We refer to the same example list used in frame 19.

Command `\appendix` declares that from that point on the chapters must be labeled “Appendix”, and must be numbered with capital letters.
Now there are the final commands,

**Finishing commands**

```latex
\backmatter% here begins the back matter
\printbibliography[heading=bibintoc]
\printnomencl% if any
\printindex% if any
\end{document}
```

The \texttt{\backmatter} changes some settings, in particular chapters are not numbered any more, and may start on pages of any parity.
Now that the thesis main file is ready, it is necessary to fill up all the chapters, possible appendices, certainly the bibliography, a possible nomenclature list, a possible index.

First we examine the material that is part of chapters: of course text, then figures, tables, equations, code lists, and so on.
Read and practice

Remember to read the article in the proceedings; read also the package documentation with \texttt{texdoc toptesi.pdf} and/or \texttt{texdoc toptesi-it}. Play experiments with the \texttt{toptesi-scudo-example.tex}: copy this file to a working folder; change its name, so as to avoid touching the original file; then comment or uncomment certain commands; change the options; at the end you might empty your example file and fill it up with your own thesis contents.
Remember that the specific module for the ScuDo theses already loads some useful, almost necessary, packages; among which *amsmath*; remember that when you use *lualatex* the symbol macros provided by package *amssymb* are already available without any need to load it. The module loads also *amsthm*.

Therefore you have available all the machinery needed to typeset advanced mathematics.

Nevertheless math expressions are large objects that cannot be floated; this might cause some little problems.
Some users feel uncomfortable to code any mathematical expression with the low level LaTeX syntax and look around for graphical interfaces in order to ease assembling such expressions; actually simple mathematical expressions are simple to code; complicated mathematical expression are simple but require much more attention.

All equation editors have the usual disadvantage of any graphical interface: in order to immediately show the result on the screen they do only what they were programmed to do, nothing else. Therefore you cannot use non standard math packages, you cannot define your own math commands, you cannot fulfill the ISO regulations for physics and applied sciences.
On my Mac I have available the \texttt{LaTeXiT.app} application. Typesetting the formula for the Laplace transform is as follows:

$$F(s) = \int_{0}^{\infty} f(t)e^{-st}dt$$
As it is possible to notice, there are some glitches, even if the formula is beautifully typeset:

- if the variables are quantities, as it is usual in engineering, the formula must fulfil the ISO regulations, and the differential sign should be in roman font;
- the number $e = 2.718281828 \ldots$ is a mathematical constant, not a physical one, and should be typeset in roman font;
- $dt$ is not spaced from the preceding exponent; it is not noticeable but if the exponential function preceded $f(t)$, the glitch would be very evident.
GUI for equations

It is not always like that, but what is the usefulness of a program if you need to write the \( \TeX \) code in the lower pane, in order to see the result in the upper pane?

Yes, it would help in getting an almost immediate view of the result before inserting the code into the \( \TeX \) source.

But, for example, the default computer Modern math fonts don’t have the symbol for the Cauchy integral principal value; then you create yourself a macro where you struggle with math kerns in order to properly place the bar across a normal integral sign; you’d need to replicate the same code over and over again because neither \texttt{LaTeXiT.app} nor any other Equation Editor, to my best knowledge, interprets and expands user macros.
The second degree equation

\[ x^2 + 2ax + b = 0 \]

has the following solutions:

\[ x_{1,2} = \begin{cases} 
-a \pm \sqrt{a^2 - b} & \text{if } a^2 > b \\
-a & \text{if } a^2 = b \\
-a \pm j\sqrt{b - a^2} & \text{if } a^2 < b 
\end{cases} \]

As it can be seen, even simple formulas such as these ones occupy a lot of vertical space.
Simple math expressions in display occupy the equivalent of three lines of text.

In general they are higher than three lines. Such large objects cannot be broken across pages, therefore they get moved to the next page, and the previous page is adjusted by inserting vertical space between paragraphs, and before and after other large objects on the page.

The result is an Underfull vbox warning, and a page that may be rather bad looking.

I’d suggest to typeset the whole thesis with the \raggedbottom declaration; I prefer a shorter page once on a while, rather than pages with uneven spacing between their components.
The above problem is really annoying with vertical alignments of math expressions.

\textit{amsmath} offers several environments to handle such vertical alignments. By default they don’t break across pages. But \textit{amsmath} defines two commands, one for breaking a single alignment, and the other to globally set this breaking facility to all alignments.

This does not avoid the fact that delimited matrices are not breakable at all, and the problem of underfull boxes may show up anyway.

\texttt{\textbackslash raggedbottom}, according to my taste, is the solution.
All figures are large objects, no matter if they are graphical objects drawn with the various drawing packages available, or are photographs or scanned images.

All of them should be floated where the typesetting engine finds enough space to output them.

This implies that every figure must have a numbered caption and any reference to each figure must be done through the use of a \texttt{\textbackslash label\{}\langle label\rangle\texttt{\}} and commands such as \texttt{\textbackslash ref\{}\langle label\rangle\texttt{\}} or \texttt{\textbackslash pageref\{}\langle label\rangle\texttt{\}}; the wording must be adapted accordingly.
Figures drawn by means of the available packages are embedded in the code of the source file of each chapter and are generally vectorial objects.

Photographs and scanned graphic files can have only the formats PDF, EPS, JPG, and PNG. The first two formats may be truly vectorial; the second two formats are bitmapped ones.

Other bitmapped formats should be transformed into the acceptable ones by means of any image processing program; among the free ones, there is **GIMP** (GNU Image Manipulation Program) that is available for all main platforms.
Float positioning

Give up the idea of putting any image “right here”; float them and use a numbered caption.

Nevertheless the typesetting program floats the large objects where it is compatible with the settings embedded in the class used to format the document.
TOPtesi is conceived to be very tolerant for what concerns float placement. But the user can customise his/her preferences for single floats by using the optional positional codes: t (at the top of the page), b (at the bottom), h (here if possible), and p (in a page of floats); moreover s/he can relax some positioning constraint by means of the position code !. Never ever use the position code h alone! Always add at least another code, otherwise the float stack might get saturated and some floats may be lost.
Tables are large objects and generally must be floated.

What has been said about figure positioning applies also to tables.

Tables may result to be too wide, too high, too high and too wide.

It is necessary to plan in advance a table so as to avoid the above difficult situations.

If a table is slightly too wide and/or too high it is possible to adjust the problem by typesetting it with a smaller sized font; passing from the normal size to $\texttt{\small}$ may reduce the table dimensions by about 10%.
When a table is only moderately too wide, it is possible to set it centered within a box so as to let it protrude evenly in both margins.

When a table is wider than \textwidth but not wider than \texttheight, and not higher than, say, 90\% of \textwidth, it is possible to output it in a page of floats, by rotating it, together with its caption, 90° anticlockwise.
If a table is too long to fit in one page, but it is not too large, it is possible to type it by means of the `longtable` or `supertabular` environments (they require to load the homonymous packages).

If a long table is wider than \textwidth but not wider than \textheight, it is possible to use the \texttt{lscape} package, especially conceived to rotate long tables that span several pages.

\texttt{lscape} is preloaded by the \texttt{toptesi-scudo} module.
The module *toptesi-scudo* loads the *nomencl* package to create a nomenclature list if the user wants or needs to use one.

This package is assumed to be easier to use than *glossaries*.

Of course it is necessary to read its documentation to use it at its best.

The *toptesi-scudo-example.tex* file shows a simple example, where a few nomenclature categories are defined and each category is populated with some entries.

The module provides the necessary actions to create the typeset nomenclature list without any user intervention.
As already mentioned, in order to create a bibliography it is necessary to store all the relevant information in a database with the name `\jobname.bib`. The `toptesi-scudo` module takes care of all stylistic aspects, unless you want to customise your own bibliography. Everything has been already described in frames 14 and 15.

Here we show just a few examples of the bibliographic records that have to be inserted in the database.
This record describes the basic book by Leslie Lamport:

```latex
@book{book:Lamport2,
    Address = {Reading, Mass.},
    Author = {Lamport, Leslie},
    Edition = {2},
    Publisher = {Addison Wesley},
    Title = {A document preparation system -- \LaTeX{} -- User's guide and reference manual},
    Year = {1994}}
```

Descriptors and fields may be entered in any order, except the citation key (in red) that must always be in the first position.
This record describes an article with many authors:

@article{art:accessibility,  
Author = {Ahmetovic, D. and Armano, T. and Berra, M. and Bernareggi, C. and Capietto, A. and Coriasco, S. and Murru, N. and Ruighi, A.},  
Journal = \{\Ars\},  
Month = {10},  
Number = {26},  
Pages = {50-54},  
Title = {Axessibility: creating PDF documents with accessible formulae},  
Year = {2018}
This is a typical example for a manual record.

A manual record

@manual{man:NIST,
    Author = {Ambler Thompson and Taylor, Barry N.},
            special-publication-811-extended-contents}},
    Organization = {NIST - National Institute of Standards and Technology},
    Title = {Guide for the Use of the International System of Units (SI)},
    Year = {2008}}

Of course urls must not be split across lines; here the url is split because of the frame short measure.
Indices require some recommendations.

The `\index` command must be attached to the word to index, otherwise the page reference might be incorrect.

If the word to index is to be typeset to the index, without any modification, a situation such as `word\index{word}` is acceptable, and a service macro can be defined to shorten the input.

In the `ArsTEXnica` article dealing with `TOPtesi` a solution is given that employs the advanced commands defined by package `xparse`; on that model many other similar macros may be defined by the user.
This topic is a hot one; on one side universities insisted on asking archivable theses in a compliant PDF/A format. On the other side no free software existed to solve this problem, especially for analysing the conformity with the ISO standard.

Eventually in 2018 lualatex was upgraded so as to produce at least PDF/A-1b compliant archivable files. But the result was not guaranteed if the document incorporated external material that infringed the regulations.

At the same time a free verification program appeared, the development of which was supported by the European Union; its name is veraPDF and can be searched and downloaded from the Internet.
In order to produce a valid PDF/A compliant file it is necessary that it contains the mandatory metadata required by the ISO regulations.

Most of these metadata are already inserted by package \textit{pdfx}; of course the package cannot insert the specific metadata that concern the thesis.

Such metadata must be entered in a special format in a file named \texttt{\jobname.xmpdata}, as it was anticipated in frame 8.
The metadata format

Their format is as follows.

Metadata

\begin{filecontents*}{\jobname.xmpdata}
\Author{\textlangle name(s) surname(s)\textrangle}
\Title{\textlangle thesis title\textrangle}
\Subject{\textlangle thesis discipline\textrangle}
\Keywords{\textlangle first keyword\textbackslash sep second keyword\textbackslash sep ...

\penultimate keyword\textbackslash sep last keyword\textrangle}
\Publisher{Politecnico di Torino}
\end{filecontents*}
Using `lualatex` the default encoding is UNICODE or its translation UTF-8.

Therefore all metadata are already correctly encoded.

It was already said that the `filecontents`* environment must be placed before loading `pdfx`, and with the facilities of the `tipotesi-scudo.sty` module, they can be put after the `\documentclass` statement and regularly updated at every compilation.

If there are no problems with colors, hyperlinks, fonts, and imported files, the PDF thesis file should be PDF/A-1b compliant.

It is necessary to verify this compliance.
With the Adobe Acrobat Pro program it is possible to verify the compliance; but since this program is a commercial one, I describe the verification through the free software **veraPDF**.

I assume that the user has already downloaded and installed this verification program and has available the **bash** or the **bat** file, depending on the fact the s/he is using a UNIX compliant platform or a Windows one.
The user opens a terminal or a command line window; and gives the command

```
verapdf-gui
```

The program opens a pane where the user enters the name of the file to check. Then s/he clicks the button [Execute], in order to launch the verification process.

After a short time the pane communicates whether the file to check is or is not PDF/A compliant.
Please specify input PDF, Validation Profile and press "Execute"

Choose PDF
Choose Profile
Auto-detection
Execute

PDF file is compliant with Validation Profile requirements

Save XML
View XML
Save HTML
View HTML
PDF/A Conformance Checker

Settings   About

Please specify input PDF, Validation Profile and press "Execute"

Choose PDF

Please, specify validation profile

Choose Profile

Report type: Validation Fix metadata

Auto-detection

Execute

PDF file is not compliant with Validation Profile requirements

Report

<XML> Save XML View XML

<HTML> Save HTML View HTML
When the verification process is finished the user can click the **XML** or **HTML** line buttons to read the response; possibly the user saves the file so as to keep memory of the result.

If the response is negative, the response tells why; the user then corrects the infringing element(s), recompiles the corrected thesis source files, and checks again the compliance.

Generally the cause might be a wrong color model or an invalid color profile in some imported file. Sometimes it is possible to correct the imported file glitches, sometimes it is very difficult; in case the user may redraw an invalid graph or redo the imported image. Some times the user is forced to look for and use a different imported file.
With this tutorial you have not become a \TeX{} guru, but you have an idea of the many things you can do with the \texttt{toptesi-scudo} module.

Your displayed thesis and your printed one have a professional typographical look, and possibly you end up asking yourself why many people keep writing important documents by means of the usual word processors.
Accept my best wishes for the conclusion of your PhD degree course and a very satisfactory defence of your thesis.
Happy TEXing with \LaTeX and TOPtesi