The `ist-report` class

Unofficial class for reports for the Instituto Superior Técnico

Daniel de Schiffart
https://github.com/ekspek/ist-relatorio

24th November 2018
v0.7.0

Abstract

This LaTeX class was developed as an unofficial template for reports developed for the Instituto Superior Técnico of Universidade de Lisboa, themed around the style the university has developed for its own internal documentation, complemented by using specifications defined by the university itself, all the while taking some creative liberties with the missing definitions.

This document details the class file, what it exactly contains, what its options are, and a concise documentation of its development, which will grow as the class file progresses. The class file itself, `ist-report.cls`, should be in the same directory as the `doc.tex` file.

Contents

1 Introduction

This project has the focus on creating a generic LaTeX report example for IST students to base their own reports upon, to streamline much of the typesetting end of things for an IST report so students and other users can focus on writing content instead of spending time getting LaTeX to work. Hopefully it can also work as a first dive for those who want to learn

LaTeX a tad more deeply than that. The final goal being having a working example using a custom class file that contains as many definitions as it can with the least possible amount of custom packages, with a couple of simple customisation options, so it can work as a simple template for those who need it. Obviously I'm taking some liberties with how it will end up looking, but hopefully it'll look pretty and professional and presentable enough.

Flexibility was also a major focus of this class file. The plan was to make something that could be used for most reports with little typographical requirements, taking those liberties and making something simple and easy to use but reusable and, most importantly, adaptable. The class files loads a number of packages and definitions which I felt were fitting to these ideas and to implement the style that I had in mind but as little as possible beyond that, leaving those extras to the user, which meant that some features were left out of the project entirely (I really really wanted to implement `minted` at some point).

The consequence of this flexibility and adaptability meant that the final product is very malleable and customizable to the user's preferences. Want to use a different font? Just load the proper package! Want to get rid of those hideous headers and footers (gee, thanks)? Include `\fancyhf{}` in your preamble! The definitions for all loaded packages can be (in theory, anyway) changed with the corresponding packages' macros and commands as if they weren't defined in the class file. This means that using this class file anyone can use the default settings and, with enough time, change it to fit their style. A world of possibilities.

This project also aims to be a learning experience for me. I've only scratched the surface of LaTeX apparently, and looking at examples from other universities and multiple other online resources made me want to see where I can go before giving up or losing my head. Lots of inspiration for working on this project also came from looking at other report and thesis class files from said universities around the world which have made their work available online, with some even including said work within TeX distributions. A particular favourite became the TUDelft's very own *Huisstijl* report class, available online at their website [2].

A lot of information for the writing and structuring of the class file itself was acquired from appendix A of *The LaTeX Companion* [3], which provided a solid base on how to build a class file based on the needs at hand. The documentation for all the implemented classes listed in section 3.4 is also relevant and should be mentioned, with most of the documentation being found online in the CTAN website [4] or within your preferred TeX distribution. The links to the CTAN page of any specific package are available by clicking said package anywhere in this document.

And of course, the TeX Stack Exchange page [1]. But that's a given.

2 Installation and Usage

The way to use this class file is quite simple, just place the class file `ist-report.cls` in the same folder as your TeX document, then within the document load the class using the `\documentclass` command in the beginning of your file, as in this next example.

```
\documentclass{ist-report}
```

The class file comes with some pre-defined optional arguments. These options can be chosen when loading the class file.

```
\documentclass[english]{ist-report}
```

In this example, the `english` option is loaded, to define the document's main language to english. You can also choose multiple options, like in this next example.

```
\documentclass[english,purist]{ist-report}
```

Here both the `english` and `purist` options are loaded. Some options will have priority over others (*i.e.*, are loaded first). This is yet to be fully documented.

The full list of options can be found in section 3.1.

2.1 IST Logos

The class file can make use of the Instituto Superior Técnico's logos for use in various places, namely the cover, but also some headers or footers. The logos are not included with the class file nor in the GitHub repo for reasons (I really don't want to risk it, I mean, it's not my work) but are available for download at the official IST website. The GitHub repo for this class file does however include placeholders to avoid any *file missing* errors and give an idea what files you should replace, but don't leave them in when you finish your document (highly recommended, I mean, really).

This class uses both the A and C versions. To make sure the logos are included in your document, place the files `IST_A_CMYK_POS.pdf` and `IST_C_CMYK_POS.pdf` in the same folder as your document file (and, consequently, the same folder as the class file). Both files can be found in the website linked previously.

3 Class File Documentation

3.1 Options

This section contains the full list of available class options and what argument to use to load them into your document.

Compiler detection The class detects the compiler being used and will react accordingly, defining certain settings to support the compiler's capabilities and shortcomings. Preferred compiler is XeLaTeX, to make full use of the class file[1].

Palatino Loaded using `palatino`. This option changes the main font to the palatino-inspired `newpxtext`, with math text using `eulervm` and typewriter text to `nimbusmono`. Off by default.

Portuguese or english Loaded using `portuguese` or using `english`. Both options refer to setting the document's main language to either portuguese or english. If using XeLaTeX, which loads `polyglossia`, both languages will be loaded, with the chosen language set as the main language. Otherwise, using `babel`, only the chosen language will be loaded. Currently, the `english` option has priority over `portuguese`, but `portuguese` is set by default.

Black and white Loaded using `bw`. This option activates black and white mode, which sets all custom colors of the document to black except for the logo colors, which cannot be changed via LaTeX. This is recommended for printing, as the cyan color used in a number of things may look faded in standard black and white printing.

Purist Loaded using `purist`. Remakes the document to match the IST thesis reference guide rules as close as possible. Currently, sets the font to Helvetica (pdfLaTeX) or TeX Gyre Heros (XeLaTeX), the line spacing to 1.5, and the margins to 2.5 cm. Takes priority over `minimal` and `palatino`. Off by default.

Minimal Loaded using `minimal`. Sets document margins to 1 cm, and the body of the document to a two-column mode.

While not fully tested, most options for the LaTeX `article` class (the class upon which the present class is based upon) can be used with this class as well, namely `twoside`. However, compatibility issues may arise for some options.

[1] XeLaTeX uses unicode input encoding, which means users can use unicode symbols in the input file without loading `inputenc`. Although not essential, this is very relevant for portuguese documents which make extensive use of unicode latin characters such as á or ç.

3.2 Cover Page

A customizable cover page was included in this version of the class file. This cover was designed by me and can be accessed using the command `\makecover{}`, which will create an entire page where it is called. It should be called preferably in the very beginning of your document, shortly after your `\begin{document}` call.

This cover makes use of the macros created in the class file. Commands have been defined to set these macros, and should be loaded before the `\makecover{}` command. Some of these macros have been loaded with default text. To empty one of these macros, simply call the command with an empty argument. The full list of these commands is available in table 1, and an example implementation is found in code listing 1.

Text field	Command	Default text	Default english text
Title	`\title`	Título de Exemplo	Sample Title
Subtitle	`\subtitle`		
Subject	`\subject`	Disciplina de Exemplo	Sample Subject
Course	`\course`	Curso de Exemplo	Sample Degree
Institution	`\institution`	Instituto Superior Técnico	
Group Number	`\groupno`		

Table 1: Document macros for covers, headers and footers.

```
\title{Relatório Laboratorial}
\subtitle{Ondas Estacionárias em Cordas Vibrantes}
\subject{Mecânica e Ondas}
\author{Daniel de Schiffart \\ 81479 \and Daniel Lopes \\ 81479}
\course{Mestrado Integrado em Engenharia Aeroespacial}
\groupno{5}
\date{Novembro de 2015}
\makecover{}
```

Listing 1: Example usage of `\makecover` and the macros in table 1.

This cover command is kinda shakey and very much in development, but I plan to implement more styles in the future. In the future. In the meantime, I suggest you make your own covers if you can. The one implemented here might work in a pinch, but the need to make everything macro-dependent does make implementation a bit harder. For anyone interested, I can make the cover code available if you contact me, or you can find it within the class file itself.

3.2.1 Title Segment

In alternative to a cover page, for more space-restricted documents, there's also a title command available, which can be compared to a cover that takes up less than half a page. Originally implemented for the `minimal` class option, it can be used in any document. Using the same macros as the `makecover` command, it takes a title, subtitle, subject and author, although the author macro in this scenario doesn't accept the `\and` command. When it is uploaded, check the example file for a basic implementation.

3.3 Headers and Footers

The class also carries some pre-designed header and footer styles designed by yours truly. These page styles are accessible by using the `\pagestyle{default}`, changing the `default` to your chosen page style. Aside from the package `fancyhdr` options `plain` and `empty`,

three styles, `default`, `style1` and `style2` have been included. Examples for all three, along with extra header and footer styles, should become available in the online repo linked in the cover within due time.

Setting a page style (style for headers and footers) is fairly straightforward, just the `\pagestyle` command with the respective page style, and TEX will set that style for the next page to be typeset.

```
\pagestyle{style1}
```

More information can be found in the documentation for the `fancyhdr` package in CTAN [4].

3.4 Packages

The class file loads some packages by default to make many options and configurations possible or, at the very least, much easier. Being loaded for the class file, you can use these packages on your own documents without calling them again. Even if you do call them again, TEX will ignore these second calls.

Some packages are loaded depending on the used engine to make full use of your engine of choice, while remaining compatible with the rest. The dependencies are listed in table 2. These packages were listed for the defualt class (*i.e.*, no optional arguments) in order of loading as best as I could, and one-engine packages were listed with their counterpart whenever such a definition applies.

pdfLaTeX	XeLaTeX
etoolbox	
ifxetex	
ifpdf	
mathtools	
geometry	
graphicx	
hyperref	
inputenc	
fontenc	
	fontspec
babel	polyglossia
microtype	
tgheros	
inconsolata	
lmodern	
xcolor	
metalogo	
fancyhdr	
footmisc	
caption	
tikz	

Table 2: Packages loaded by the class file with no optional arguments given.

The documentation for all packages listed here can be found at CTAN [4].

4 Changing the Design

If there's any part of the design you want to change, or just reformulate the whole thing entirely, no problem. One of the ideas behind this class was compatibility and minimalism,

which means it was implemented using as little packages as possible for defining the look of the class, and without changing any major TEX or LATEX definitions. This means you can, for the most part, change any class definition by changing the properties of the package loaded for the purpose, or importing new classes to set things your way. As an example, you can change or create a page style by using the commands listed in the `fancyhdr` package documentation, the package used to implement the class page styles. The margins can be redefined using the `geometry` package. The list of packages can be found in section 3.4, and the documentation for each of them can be found at CTAN [4].

Fonts, however, can be a bit trickier. For the most part, the fonts found at The LATEX Font Catalogue can be used as-is, by implementing them in your preamble. If you're using XELATEX, however, changing the font is more difficult because of `fontspec`. While I write down the details, you can read `fontspec`'s documentation for help.

5 Problems, Errors and Contributions

There are problems with the package, be that in actual engine errors, design mismatches or outdated or obsolete packages. I still have a long list of things I want to implement, and that might still take a while. So in the meantime, if you come across any problems or inconveniences, have a suggestion, or need some help, you can simply contact me whatever way you prefer, be that via email at `daniel.de.schiffart@gmail.com` or `daniel.de.schiffart@tecnico.ulisboa.pt`, or via GitHub at the repo page itself, which I've linked to in the cover page.

If you want to contribute directly, with code, documentation, page style or cover designs, you're more than welcome! I'm not all too familiar with the inner workings of GitHub, but you can try forking or opening a pull request on the repo, or even mailing me any contributions you like. I'll be sure to give you the proper credit.

6 License

This project is licensed using the LATEX Project Public License v1.3. What this exactly entails I don't exactly know, I'm not a lawyer, but the gist of it is that you can use and distribute this stuff as much as you want, send it to friends and family, but you can only modify and distribute it if you change the file name. More details are available online, but that's about as concise as I can make it here.

Don't be discouraged to distribute the class file if you are pleased with it and feel like other people could use it, I'd very much appreciate it! Don't be discouraged to modify and experiment with it either, I'm always open for feedback!

7 Changelog

v0.1.0 First working release. Included header and footer, 2.5 cm margins. Implemented `palatino` option to separate font choices (default is `lmodern`), implemented TEX conditionals to detect XETEX or pdfTEX.

v0.2.0 Added `portuguese` and `english` options to change report macro language, included IST colors and applied them to `hyperref`. Option `portuguese` runs by default and doesn't need to be included.

v0.2.1 Added black and white option (except logos), added IST logo crop margins.

v0.3.0 Reworked `twoside` option to work with already implemented features. Implemented `etoolbox` package for TEX conditionals.

v0.4.0 Added the `\makecover` command with a simple cover example. Added the first version of the placeholder logos.

v0.5.0 Separated covers and header and footer styles into separate files (might undo later for release). Headers and footers are now bound to the `\pagestyle` command. By default, the `default` page style is loaded. Fixed wrong IST logo crops.

v0.5.1 Added `basic` option. Added one more style.

v0.6.0 Added `purist` option, only compatible with non-unicode engines.

v0.6.1 First working version of example cover. Fixed cover to use the `article` class `\author{}` command. Changed default style. Fixed group number command and implemented it into cover. Small code cleanup.

v0.6.2 Removed `basic` option.

v0.6.3 Added compatibility with unicode engines for the `purist` option and changed typewriter text to `inconsolata`.

v0.7.0 Resolved conflict between options, defined option priority. Added `minimal` option.

References

[1] *TEX – LATEX Stack Exchange*. URL: https://tex.stackexchange.com/ (visited on 11/17/2018).

[2] *Huisstijl TUDelft*. URL: https://www.tudelft.nl/huisstijl/ (visited on 11/17/2018).

[3] Frank Mittelbach et al. *The LATEX Companion*. 2nd ed. Addison-Wesley, 2004. ISBN: 0-201-36299-6.

[4] CTAN Team. *The Comprehensive TEX Archive Network*. URL: https://ctan.org/ (visited on 11/17/2018).