Abstract

This bundled is aimed at producing undergraduate students final work/report at UFRGS/EE (Engineering School at the Federal University of Rio Grande do Sul), closely following ABNT rules (Brazilian Association for Technical Norms). It is composed of a main class, \texttt{ufrgscca}, and a set of auxiliary packages, some of which can be used independently.

Contents

1 Introduction ......................................... 2
1.1 Current Version ...................................... 2
2 \texttt{ufrgscca} Class .................................. 2
2.1 Class Options ...................................... 3
2.2 Class Declared Commands ............................ 4
2.3 Class Known Hooks .................................. 4
3 \texttt{ufrgscca-abnt} Package ......................... 4
3.1 Package Options ..................................... 4
3.2 Environments ........................................ 6
3.3 Tabular New Columns ................................ 6
3.4 enumitem Extra Keys ................................ 7
4 \texttt{ufrgscca-core} Package ......................... 9
4.1 Core Forms Commands ............................... 9
4.2 Core Global Commands ............................. 10
4.3 Core Specific Commands ............................. 10
5 \texttt{ufrgscca-cover} Package ......................... 10
5.1 Package Options ..................................... 10
5.2 Defined Commands ................................... 10
6 \texttt{ufrgscca-forms} Package ........................ 11
6.1 Forms Defined Commands ........................... 11
7 \texttt{ufrgscca-lists} Package ......................... 11
7.1 Environment ......................................... 11
7.2 Declared Commands ................................ 12
8 \texttt{ufrgscca-gen} Package (extended documentation) 12
8.1 Package Options ................................... 12
8.2 Defined Commands .................................. 12
1 Introduction

ABNT rules can be quite challenging some times (read: bibliography style/references) and sometimes just odd (line spacing, front matter, page layout), nevertheless it is a Brazilian Standard for typography whose students at UFRGS should grow cherished to follow.

In short, as of version 1.0.4 the bundle is composed of a class, \texttt{ufrgscca} (based on the standard \LaTeX report class), which pre-loads all other, as needed, packages: \texttt{ufrgscca-abnt}, \texttt{ufrgscca-core}, \texttt{ufrgscca-cover}, \texttt{ufrgscca-forms}, \texttt{ufrgscca-gen}, \texttt{ufrgscca-lists}, \texttt{ufrgscca-curr}, \texttt{ufrgscca-coord}, \texttt{ufrgscca-ppc}.

N.B.: This bundle requires a quite recent \LaTeX kernel, at least as recent as June 2022, which allows to declare package options using the new \texttt{key=value} system and declare commands with \texttt{\NewDocumentCommand}, out-of-the-box.

1.1 Current Version

For the sake of the “maintainer’s sanity”, since this is a bundle, all files are saved with the same version (bundle version), with two exceptions: \texttt{ufrgscca-curr.sty} \texttt{ufrgscca-ppc.sty} which are less tested than the others, and somewhat in what one would call ’beta’ state. Better said, all files are version 1.0.4, except \texttt{ufrgscca-curr} and \texttt{ufrgscca-ppc} whose versions are 1.0.4beta.

2 \texttt{ufrgscca} Class

The following packages are always pre-loaded: \texttt{etex}, \texttt{etoolbox}, \texttt{lmodern}, \texttt{fontenc} (\texttt{T1}), \texttt{inputenc} (\texttt{utf8}), \texttt{silence}, \texttt{ufrgscca-abnt}, \texttt{ufrgscca-gen}, \texttt{ufrgscca-cover}, \texttt{ufrgscca-core}, \texttt{hyperref} and (if it exists) a \texttt{local.tex} file.

Other set of auxiliary packages are also pre-loaded, depending on the class options used, and finally it loads (normally) the \texttt{report} class (the exception being if one uses the \texttt{dctools} option).

Being based on the report class, one can use all class options one would with a report, plus the ones listed below.
2.1 Class Options

**tocdepth** use: `tocdepth = ⟨number⟩`, whereas ⟨number⟩ indicates the deepest sectioning to appears in the Table of Contents (0 being the top section, which is \chapter for report based classes, 1 being \section, and so on.) The default value being 3 (\subsubsection).

**secdepth** use: `secdepth = ⟨number⟩`, whereas ⟨number⟩ indicates the deepest sectioning to be numbered. (0 being the top section, which is \chapter for report based classes, 1 being \section, and so on.) The default value being 4 (\paragraph).

**english** in some rare cases (to be further investigated) babel seems to get confused about which language is active, this “shouldn’t be necessary” but one can explicitly tell babel to use THIS language (which should, otherwise, be the default one).

**brazilian** the default language being Portuguese, this option changes locale to English.

**relnum** by default, figures, tables, etc. are numbered as a continuous series. With this switch, those lists are reset at each chapter, e.g. Figure 5.1 instead of Figure 23.

**openright** in case of printed material, this will assure that a \chapter always starts at an odd page, which is relevant in case of printing out (double sided) the document.

**oneside** in case the document will be printed in single side sheets, otherwise it’s assumed a two-sided printing.

**strict-abnt** to assure asymmetric margins, as defined by ABNT: inner ones greater than outer ones, which matters if you are going to print the doc and make a book of it, but makes it odd to look at in a computer screen, reason by which the current default setting is for symmetric margins (same text width).

**pretextontoc** “pre-text” elements, like “list of...” will be inserted in the “table of contents”.

**timesroman** will set the default font to Roman (using the obsolete mathptmx package, based on a free replacement of the proprietary Times New Roman (by Microsoft) and Times Roman (By Adobe)) instead of the default Latin Modern Roman font. As a side effect, the package microtype isn’t loaded (can’t be used), resulting in a worst overall layout. NB. The alternative (newer and maintained) packages newtxtext/newtxmath can’t be used due to some packages incompatibilities.

**repeatfields** in case of authors with multiple publications, their names will be repeated for each entry. In the default setting the author’s name is written only in the first entry, and replaced by underscores in the other entries.

**xlists** this will load the ufrgscca-lists package, for the definition of new floats/lists.

**xpacs** this will load a series of packages, which can be handy when writing Engineering reports: relsize, keyval, graphicx, mathtools, mathrsfs, amssfonts, amssymb, empheq, amsthm, extarrows, mathfixs, bigdelim, circuitikz, steimenz and tikz libraries: fit, math, calc, shapes.geometry, shapes.misc, shapes.multipart, graphs, 3d, positioning, shadows, babel. One is advised to look after each package documentation (ctan.org) for further information.

**report** in case the doc is just a class assignment with, possibly, many co-authors. It changes mainly the front matter, which is simplified (no referral page, for instance).

**internship** in case the doc is an internship report.

**forms** in the process of submitting a student final work/report, there is a series of forms to be submitted, this allows the customization of said forms in a simple way.

**chapternopagenum** to suppress the page numbers at chapters begin.

**nomicrotype** in some rare cases, microtype might hurt page layout, this allows the suppression of microtype.
showframes for layout proof only, it will draw frames around each page main parts.
showlabels it will put a reference mark in each label created, and print out it’s name.
nofontwarning in case of \texttt{ufrgscca-ppc} is loaded, it will suppress some font related warnings.
nolocal this will suppress the loading of any local\texttt{.tex} file, which would, otherwise, be loaded.
dctools this will change page layout and base class to article, it is meant to document the class itself.
texlive this is a reserved key, in case some workaround for \texttt{texlive} is needed.
overleaf this is a reserved key, in case some workaround for \texttt{overleaf} is needed.
miktex this is a reserved key, in case some workaround for \texttt{miktex} is needed.

\section{Class Declared Commands}

\begin{itemize}
  \item \texttt{\autonameref} \{\texttt{\langle sep\rangle}\} \{\texttt{\langle label\rangle}\} \{\texttt{\langle spc\rangle}\}
  \item \texttt{\annexref} \{\texttt{\langle label\rangle}\}
  \item \texttt{\autoannexref} \{\texttt{\langle sep\rangle}\} \{\texttt{\langle label\rangle}\} \{\texttt{\langle spc\rangle}\}
\end{itemize}

The \texttt{hyperref} package, sometimes, gets the \texttt{\autoref} name wrong (when referencing an annex), therefore the \texttt{\annexref} \{\texttt{\langle label\rangle}\} will assure the correct annex name is used.
\texttt{\autonameref} \{\texttt{\langle label\rangle}\} produces an entry of the form \texttt{\autoref} \{\texttt{\langle label\rangle}\} \{\texttt{\langle sep\rangle}\} \texttt{\nameref} \{\texttt{\langle label\rangle}\} \{\texttt{\langle spc\rangle}\}
\texttt{\autoannexref} \{\texttt{\langle label\rangle}\} produces an entry of the form \texttt{\annexref} \{\texttt{\langle label\rangle}\} \{\texttt{\langle sep\rangle}\} \texttt{\nameref} \{\texttt{\langle label\rangle}\} \{\texttt{\langle spc\rangle}\}
The default \texttt{\langle sep\rangle} being a comma, and the default \texttt{\langle spc\rangle} being empty space.

\section{Class Known Hooks}

\begin{itemize}
  \item \texttt{\miktexHack}
  \item \texttt{\overleafHack}
  \item \texttt{\livetexHack}
\end{itemize}

In case some workaround is needed due an unexpected error (when upgrading \LaTeX{}system) the class “knows” about those three hooks. They will be executed if, and only if, they are user defined and the corresponding package option is used, i.e., for example, for the hook \texttt{\miktexHack} to be used/called by the class \texttt{ufrgscca}, one has to: a) defined it and b) use the class option \texttt{miktex}.

\section{ufrgscca-abnt Package}

This package is the one that sets the page layout (using \texttt{geometry}, \texttt{titlesec}, \texttt{titletoc}) and adjusts the main float environments (figure, tables, captions). It can be used as a stand alone package, regardless of the underlying class.

The following packages are always pre-loaded: \texttt{biblatex}, \texttt{csquotes}, \texttt{geometry}, \texttt{appendix}, \texttt{titlesec}, \texttt{titletoc}, \texttt{enumitem}, \texttt{chngctr}, \texttt{caption}, \texttt{biblatex}, \texttt{microtype}, \texttt{array}, \texttt{nicematrix}, \texttt{contour} and \texttt{soul}.

Take note that \texttt{biblatex} is loaded with the \texttt{biber} option, to correctly handle ABNT biography style.

\subsection{Package Options}

\texttt{strict-abnt} to assure asymmetric margins, as defined by ABNT: inner ones greater than outer ones, which matters if you are going to print the doc and make a book
of it, but makes it odd to look at in a computer screen, reason by which the current default setting is for symmetric margins (same text width).

to suppress the page numbers at chapters begin.

by default, figures, tables, etc. are numbered as a continuous series. With this switch, those lists are reset at each chapter, e.g. Figure 5.1 instead of Figure 23.

in case of authors with multiple publications, their names will be repeated for each entry. In the default setting the author’s name is written only in the first entry, and replaced by underscores in the other entries.

in some rare cases, \textit{microtype} might hurt page layout, this allows the suppression of \textit{microtype}.

for layout proof only, it will draw frames around each page main parts.

use: \texttt{tocdepth = \langle \text{number} \rangle}, whereas (\text{number}) indicates the deepest sectioning to appears in the Table of Contents (0 being the top section, which is \texttt{chapter} for report based classes, 1 being \texttt{section}, and so on.) The default value being 3 (\texttt{\subsubsection}).

use: \texttt{secdepth = \langle \text{number} \rangle}, whereas (\text{number}) indicates the deepest sectioning to be numbered. (0 being the top section, which is \texttt{chapter} for report based classes, 1 being \texttt{section}, and so on.) The default value being 4 (\texttt{\paragraph}).

this will change page layout and base class to article, it is meant to document the class itself.

\keyword{\langle \text{keyword} \rangle}
This command can be invoked many times, it will construct a list of keywords to be used when printing out the abstract environment.

\sourcecitation{\langle \text{source} \rangle}
\note{\langle \text{text} \rangle}
When describing floating elements (like figure, tables, circuits) one always has to cite the source of it, and in some cases it might be necessary to add a special note. Those assure uniformity when doing that.

\nonum\chapter{\langle \text{chap.title} \rangle}
\notoc\chapter{\langle \text{chap.title} \rangle}
\notoc\section{\langle \text{sec.title} \rangle}

In some cases, it might be necessary to create a numberless chapters or sections. Those two commands can be used as a prefix to any sectioning command. Whilst \texttt{\nonum} will just suppress the sectioning number, the \texttt{\notoc} will also suppress it from the table of contents.

\LaTeX{} Code:

```
\nonum\chapter{some title} %this one will appear in the toc
\notoc\section{some other title} %this won't even appear in the toc
```

\tightul{\langle \text{text} \rangle}
This will underline a short text, take note that \langle \text{text} \rangle ‘can’t be broken’ (think paragraph justification), which can lead to text overflows and bad justification.

\LaTeX{} Code:

```
\tightul{Some text example}\% Some text example
```

\NewChapListEnv{\langle \text{envname} \rangle}{\langle \text{displayname} \rangle}
This is the command used to create those chapter like lists, like ‘List of Symbols’ or ‘List of acronyms’. With it, a new environment is created, (envname), with an associated ‘numberless’ chapter name (displayname). The newly created environment will implement a description like environment (thanks to enumitem) with an optional and a mandatory argument (see below).

\begin{itemize}
\item \texttt{\textbackslash def\listabbrvname{Lista de Abreviaturas}}
\item \texttt{\textbackslash NewChapListEnv{listoffabbrv}\{\listabbrvname\} % this is the actual code used in ufrgscca-abnt.sty}
\end{itemize}

The command \texttt{\textbackslash date} is redefined, to allow a separation between the many arguments (day), (month) and (year). If not called by the user it defaults to current month / year. \texttt{\textbackslash today} returns the current locale date, whilst \texttt{\textbackslash monthname} returns the locale name of the current month.

\subsection{Environments}

\begin{itemize}
\item \texttt{\textbackslash begin\{abstract\} [\langle\text{lang}\rangle] \{\langle\text{keywords}\rangle\}...\texttt{\textbackslash end\{abstract\} }}
\item \texttt{\textbackslash begin\{otherabstract\} [\langle\text{lang}\rangle] \{\langle\text{keywords}\rangle\}...\texttt{\textbackslash end\{otherabstract\} }}
\item \texttt{\textbackslash begin\{listoffabbrv\} [\langle\text{enum-opt}\rangle] \{\langle\text{ABBRV}\rangle\}...\texttt{\textbackslash end\{listoffabbrv\} }}
\item \texttt{\textbackslash begin\{listofsymbols\} [\langle\text{enum-opt}\rangle] \{\langle\text{SYMB}\rangle\}...\texttt{\textbackslash end\{listofsymbols\} }}
\item \texttt{\textbackslash begin\{appendix\}.... \texttt{\textbackslash end\{appendix\} }}
\item \texttt{\textbackslash begin\{annex\}.... \texttt{\textbackslash end\{annex\} }}
\end{itemize}

Both environments create a description like list preceded by a numberless (\texttt{\textbackslash nonum}) chapter. (enum-opt) is any enumitem list valid key. Whereas (ABBRV) / (SYMB) are just the ‘biggest’ abbreviation/symbol to be used as a tab reference.

\subsection{Tabular New Columns}

Thanks to \texttt{array} some new columns types are defined:
3.4 *enumitem* Extra Keys

Besides the *default* keys defined by the *enumitem* package a few others are defined for author’s convenience:

**ppc, tcc**  
*ppc* and *tcc* are alias of each other, and just assure that lists indentation will be the same as paragraphs default.

**parindent**  
With *parindent*, the list number/mark is aligned with paragraph indentation.

**noindent**  
*noindent* removes the label indentation.

**tight**  
*tight* allows for very tight lists (no indentation) to be used, for instance, inside quotes. N.B. don’t use it in normal paragraph mode, otherwise the labels will spill outside the default text window.

**miditemsep**  
*miditemsep* halves items separation, as an alternative to *noitemsep* from *enumitem*.

**arabic**  
That’s the *default* enumerate style. Arabic numbers, starting at 1, followed by a dot.
arabic) Label will be constructed as number followed by a parenthesis.
(arabic) Label will be enclosed by parenthesis.
arabic* (for secondary lists) Label will be constructed by the label of the outer list, this item number and a final dot.
arabic**) (for secondary lists) Label will be constructed by the label of the outer list, this item number and a final parenthesis.
roman This and below keys are the same as the arabic ones, but using lower case roman numbers.
roman) lower case roman number, followed by a parenthesis.
(roman) enclosed by parenthesis.
roman* preceding one followed by roman number and a final dot.
roman**) same, followed by a final parenthesis.
Roman This and below keys are the same as the arabic ones, but using upper case roman numbers.
Roman) upper case roman number, followed by a parenthesis.
(Roman) enclosed by parenthesis.
Roman* preceding one followed by roman number and a final dot.
Roman**) same, followed by a final parenthesis.
alpha This and below keys are the same as the arabic ones, but using lower case alpha numbers.
alpha) lower case alpha number, followed by a parenthesis.
(alpha) enclosed by parenthesis.
alpha* preceding one followed by alpha number and a final dot.
alpha**) same, followed by a final parenthesis.
Alpha This and below keys are the same as the arabic ones, but using upper case alpha numbers.
Alpha) upper case roman number, followed by a parenthesis.
(Alpha) enclosed by parenthesis.
Alpha* preceding one followed by roman number and a final dot.
Alpha**) same, followed by a final parenthesis.
\LaTeX\ Code: \LaTeX\ Result:
\begin{enumerate}[tcc,roman]
\item some A
\item some B
\item some C
\end{enumerate}

\begin{enumerate}[tcc,roman]
\item some A
\item some B
\begin{enumerate}[tcc,alpha*]
\item some A
\item some B
\item some C
\end{enumerate}
\item some C
\end{enumerate}

\begin{enumerate}[tcc,arabic]
\item some A
\item some B
\begin{enumerate}[tcc,roman*)]
\item some A
\item some B
\item some C
\end{enumerate}
\item some C
\end{enumerate}

\begin{itemize}[tcc,miditemsep]
\item some A
\item some B
\item some C
\end{itemize}

\begin{itemize}[tcc,bullet, miditemsep]
\item some A
\item some B
\item some C
\end{itemize}

for simple itemized lists, it will replace the default black dot by an ‘open bullet’ 

| \begin{itemize}[tcc,miditemsep]  |
| \item some A  |
| \item some B  |
| \item some C  |
| \end{itemize} |

\begin{itemize}[tcc,bullet, miditemsep]
\item some A
\item some B
\item some C
\end{itemize}

\begin{itemize}[tcc,bullet, miditemsep]
\item some A
\item some B
\item some C
\end{itemize}

4. ufrgscca-core Package

The \texttt{ufrgscca-core} package defines a set of commands for authors, students, advisors and examiners names and related info. It is needed by most/all of the tc bundled packages.

4.1 Core Forms Commands

\begin{itemize}
\item \texttt{tccbrief} \{(brief)\}
\item \texttt{tcccoadvisorbrief} \{(brief)\}
\item \texttt{ttccadvisorsreview} \{(brief)\}
\end{itemize}

Those commands are only of use when using \texttt{ufrgscca-forms}. \texttt{tccbrief} sets the work initial summary, \texttt{tcccoadvisorbrief} sets the justification for having
4.2 Core Global Commands

\texttt{\textbackslash location}

\texttt{\textbackslash location\{\texttt{\textbackslash (city)}\} \{\texttt{\textbackslash (state)}\}}

To redefine the default values of \texttt{\textbackslash (city)} and \texttt{\textbackslash (state)} (Porto Alegre and RS).

\texttt{\textbackslash TCCcoord}

\texttt{\textbackslash TCCcoord\{\texttt{\textbackslash (title)} full name\}\{\texttt{\textbackslash (gender)}\}}

\texttt{\textbackslash TCCcoordtitle}

\texttt{\textbackslash TCCcoordtitle\{\texttt{\textbackslash (coordinator denomination)}\}}

\texttt{\textbackslash coursecoord}

\texttt{\textbackslash coursecoord\{\texttt{\textbackslash (title)} full name\}\{\texttt{\textbackslash (gender)}\}}

\texttt{\textbackslash coursecoordtitle}

\texttt{\textbackslash coursecoordtitle\{\texttt{\textbackslash (course coordinator denomination)}\}}

\langle\texttt{\textbackslash (coordinator denomination)}\rangle and \langle\texttt{\textbackslash (course coordinator denomination)}\rangle are the full 'job title' of their position. \langle\texttt{\textbackslash (gender)}\rangle can be either 'm' or 'f'.

4.3 Core Specific Commands

The following commands are more or less self-explanatory, \langle\texttt{\textbackslash (ID)}\rangle is the student’s university ID. \langle\texttt{\textbackslash (Nproc)}\rangle is the process/request number. \langle\texttt{\textbackslash (gender)}\rangle can be either 'm' or 'f'.

\texttt{\textbackslash author}

\texttt{\textbackslash author\{\texttt{\textbackslash (last)}\} \{\texttt{\textbackslash (first)}\}\{\texttt{\textbackslash (gender)}\}}

\texttt{\textbackslash authorinfo}

\texttt{\textbackslash authorinfo \{\texttt{\textbackslash (Nproc)}\} \{\texttt{\textbackslash (ID)}\} \{\texttt{\textbackslash (email)}\}}

\texttt{\textbackslash student}

\texttt{\textbackslash student\{\texttt{\textbackslash (last)}\} \{\texttt{\textbackslash (first)}\}\{\texttt{\textbackslash (gender)}\}}

\texttt{\textbackslash studentinfo}

\texttt{\textbackslash studentinfo \{\texttt{\textbackslash (Nproc)}\} \{\texttt{\textbackslash (ID)}\} \{\texttt{\textbackslash (email)}\}}

\texttt{\textbackslash advisor}

\texttt{\textbackslash advisor\{\texttt{\textbackslash (title)}\} \{\texttt{\textbackslash (last)}\} \{\texttt{\textbackslash (first)}\}\{\texttt{\textbackslash (gender)}\}}

\texttt{\textbackslash advisorinfo}

\texttt{\textbackslash advisorinfo \{\texttt{\textbackslash (Institut)}\}\{\texttt{\textbackslash (title-info)}\} \{\texttt{\textbackslash (email)}\} \{\texttt{\textbackslash (phone)}\}}

\texttt{\textbackslash coadvisor}

\texttt{\textbackslash coadvisor\{\texttt{\textbackslash (title)}\} \{\texttt{\textbackslash (last)}\} \{\texttt{\textbackslash (first)}\}\{\texttt{\textbackslash (gender)}\}}

\texttt{\textbackslash coadvisorinfo}

\texttt{\textbackslash coadvisorinfo \{\texttt{\textbackslash (Institut)}\}\{\texttt{\textbackslash (title-info)}\} \{\texttt{\textbackslash (email)}\} \{\texttt{\textbackslash (phone)}\}}

\texttt{\textbackslash examiner}

\texttt{\textbackslash examiner\{\texttt{\textbackslash (title)}\} \{\texttt{\textbackslash (last)}\} \{\texttt{\textbackslash (first)}\}\{\texttt{\textbackslash (gender)}\}}

\texttt{\textbackslash examinerinfo}

\texttt{\textbackslash examinerinfo \{\texttt{\textbackslash (Institut)}\}\{\texttt{\textbackslash (title-info)}\} \{\texttt{\textbackslash (email)}\} \{\texttt{\textbackslash (phone)}\}}

\texttt{\textbackslash altexaminer}

\texttt{\textbackslash altexaminer\{\texttt{\textbackslash (title)}\} \{\texttt{\textbackslash (last)}\} \{\texttt{\textbackslash (first)}\}\{\texttt{\textbackslash (gender)}\}}

\texttt{\textbackslash altexaminerinfo}

\texttt{\textbackslash altexaminerinfo \{\texttt{\textbackslash (Institut)}\}\{\texttt{\textbackslash (title-info)}\} \{\texttt{\textbackslash (email)}\} \{\texttt{\textbackslash (phone)}\}}

5 \texttt{ufrgscca-cover} Package

This package is the one that sets the front pages, depending on the kind of 'report' being generated. The default being to generate 3 cover pages: an identification on, followed by presentation one, then an referral/approval one.

5.1 Package Options

\texttt{\textbackslash report}

in case the doc is just a class assignment with, possibly, many co-authors. It changes mainly the front matter, which is simplified (no referral page, for instance).

\texttt{\textbackslash internship}

in case the report is a internship one.

5.2 Defined Commands

\texttt{\textbackslash maketitle}

This is the only main command, which will typeset the front matter. It requires that all \textit{specific info} be already set up (like work title, author's name, affiliation, etc.)
In case some customization is needed, one can change them as needed. The default values are set for the control and automation course at UFRGS/EE.

6 ufrgscca-forms Package

This package defines just two user commands to generate specific forms needed at UFRGS/EE.

6.1 Forms Defined Commands

\tcforms \tcforms \tcemptyforms \tcemptyforms

\tcforms will generate the many forms (\formslist) using the information from local.tex, whilst \tcemptyforms will generate said forms with 'blanks' (to be fulfilled by hand, for instance).

\formslist is a csv list of any of:

reqform Registration requirement form.
coadvisor Coadvisor justification form.
boardsapproval Boards approval form.
advisorsapproval Advisors approval form.
receipts Receipts forms (one per board member).
examinersforms Grades and correction forms (per board member).
rectifyapprovalform Corrections approval form.

7 ufrgscca-lists Package

The following packages are always pre-loaded: newfloat, listings and xcolor. It defines a new floating environment. Combined with listings one can typeset exempts of code listing.

7.1 Environment

codelist \begin{codelist}...\end{codelist}
\caption{sample C code}
\label{code01}
\begin{lstlisting}[language=C]
struct i2c_msg
{
  __u16 addr; /* endereco do escravo */
  __u16 flags;
}
\end{lstlisting}
\sourcecitation{Garg:SMA-2000}}
\end{codelist}

7.2 Declared Commands

\listofcodelist
This will create the 'List of ...' associated with the previous environment.

\DeclareNewFloat{\langle env-name\rangle}{\langle file-ext\rangle}{\langle listname\rangle}{\langle listofname\rangle}
A new float environment, named \textit{env-name}, will be created. Captions will be associated (numbered) as \textit{listname num}. Finally, an associated command \texttt{listof...} will be defined, using \textit{listofname} as a numberless \texttt{chapter} title.

\LaTeX Code:
\def\listingname{Listing}\
\def\listlistingname{List of Listings}\
\DeclareNewFloat{\codelist}{lox}{\listingname}{\listlistingname}%% after that, one can do as in the previous example
\listofcodelist

8 \texttt{ufrgscca-gen} Package (extended documentation)

Just two set of commands are defined, one is kind of a 'command factory' aimed at creating macros in a standard way, while the other helps create 'case like' commands.

8.1 Package Options

\begin{itemize}
\item \texttt{family} sets the family name, defaults to \texttt{tcdef}.
\item \texttt{group} sets the group name, defaults to \texttt{gen}.
\end{itemize}

8.2 Defined Commands

\cmdfactory\ \texttt{\langle fam\rangle} \langle\langle grp\rangle\rangle \langle\langle cmd-list\rangle\rangle
\factory\ \texttt{\langle fam\rangle} \langle\langle grp\rangle\rangle \langle\langle cmd\rangle\rangle
\tcgen@cdef\ \texttt{\langle fam\rangle} \langle\langle grp\rangle\rangle \langle\langle cmd\rangle\rangle \langle\langle new-val\rangle\rangle

\cmdfactory is the actual command meant to be used (the other two are just auxiliary ones). \texttt{\langle cmd-list\rangle} is a csv list of commands. \texttt{\langle fam\rangle} is the command family (defaults to \texttt{tcdef}) and \texttt{\langle grp\rangle} is the family group (defaults to \texttt{gen}). The newly created commands will be based on \tcgen@cdef (the actual assignment command) having the form \texttt{\langle new-val\rangle}, accepting a single mandatory value. Internally \texttt{\langle new-val\rangle} will be stored in a macro likely named \texttt{\langle fam\rangle\langle grp\rangle\langle cmd\rangle}. 
\texttt{\textbackslash factory} is basically the same as \texttt{\textbackslash cmdfactory}, whilst to create just one new command (it is the command called by \texttt{\textbackslash cmdfactory} via \texttt{\textbackslash forcsvlist}).

\begin{verbatim}
\texttt{\textbackslash mkswitch \{\textbackslash default\}\{\textbackslash sw-name\}\{\textbackslash str-case\}\{\textbackslash code\}}
\texttt{\textbackslash addcase \{\textbackslash sw-name\}\{\textbackslash str-case\}\{\textbackslash code\}}
\end{verbatim}

\texttt{\textbackslash mkswitch} will create a command, \texttt{\textbackslash sw-name\{\textbackslash case\}}, which will behave like a switch/case in other programming languages. \texttt{\{\textbackslash default\}} is the code to be executed in case a switching value isn’t defined. \texttt{\textbackslash addcase} adds cases, one by one, to the switch. \texttt{\{\textbackslash str-case\}} can be any \texttt{\textbackslash csname} valid name. \texttt{\{\textbackslash code\}} is the code to be executed.

\texttt{\LaTeX} Code:

\begin{verbatim}
\texttt{\textbackslash mkswitch[\textbackslash gr\textbackslash depcut]\textbackslash gr\textbackslash case\textbackslash angle}
\texttt{\textbackslash addcase\textbackslash gr\textbackslash case\textbackslash angle{}{\texttt{\textbackslash def}\texttt{\textbackslash gr\textbackslash ANG\{0\}}}}
\texttt{\textbackslash addcase\textbackslash gr\textbackslash case\textbackslash angle\{A\}{\texttt{\textbackslash def}\texttt{\textbackslash gr\textbackslash ANG\{\textbackslash gr\textbackslash A\}}}}
%%
%% actual use of the switch
\texttt{\textbackslash gr\textbackslash case\textbackslash angle\{A\} } % this will result in \texttt{\textbackslash def}\texttt{\textbackslash gr\textbackslash ANG\{\textbackslash gr\textbackslash A\}}
\end{verbatim}

9 \texttt{ufrgscca-coord} Package (extended documentation)

This package defines a set of auxiliary commands meant to support the Professor coordinating students work. It will always pre-load the \texttt{longtable} and \texttt{ufrgscca-forms} packages. One can select the reports/forms to be generated using the package options or the command \texttt{\textbackslash setreports \{\textbackslash keys\}}

N.B. It might be also useful to use the commands defined at subsection 6.1, Forms Defined Commands.

9.1 Package/Report Options

\begin{itemize}
  \item \texttt{calendar} Calendar for the period.
  \item \texttt{checklist} a students check list.
  \item \texttt{report} a student control report.
  \item \texttt{reportxinfo} report additional info.
  \item \texttt{boards} exam board dates.
  \item \texttt{boarddates} exam board dates with highlighted dates.
  \item \texttt{studentlist} a simple student list.
  \item \texttt{revforms} per student reviews forms.
  \item \texttt{referral} per student referral letters.
  \item \texttt{cocertificate} per student coadvisor certificate letter (if any).
\end{itemize}

9.2 Defined Commands

The \texttt{report document} to be created is composed of 2 main parts:

1. A global preamble, where one sets
   1.a. the current semester, Course/TCC coordinator names, etc.,
   1.b. auxiliary data, like students \texttt{check list} items and
   1.c. students data.

2. A 'final part' whereas one set which reports are to be generated.
9.2.1 Global Commands I

One can (should) use the commands listed at subsection 4.2, Core Global Commands, and these below:

\tcccalendareventdate \langle date \rangle
\tcccalendareventdate \langle titleB \rangle
\boardsObs \langle obs \rangle
\TCCperiod \langle semester \rangle

Use \tcccalendareventdate to set the date of a given 'event' (the list of 'calendar events' are (might have been) set in the ufrgscca-ptBR-coord.def or ufrgscca-en-coord.def file). \boardsObs sets a 2nd title line for the 'boards schedule report'. \TCCperiod allows to add an observation ('obs') for the 'boards schedule report', finally, \TCCperiod \langle s \rangle sets the current semester value.

\tcceventAweek \langle week num. \rangle
\tcceventBweek \langle week num. \rangle

\tcceventJweek \langle week num. \rangle

Those macros allow to change the default week value for the calendar's events.

\checkdef \langle checkLC \rangle \langle check-item \rangle \langle check-text \rangle

Whereas one has a '4x5 alphabetic matrix', lines A to D, columns A to E. \checkdef \langle checkLC \rangle being one element of that matrix (from checkAA up to checkDE), \langle check-item \rangle is a free identifier (to be used with the \checklist), and \langle check-text \rangle the text to appear in the 'check list report'. So, for instance:

\checkdef \langle checkAA \rangle \langle tcc-part \rangle \langle Rel. Parcial \rangle
% this creates the 'check item' tcc-part and associates it with the AA position (first line, first column), display text 'Rel. Parcial'
\checkdef \langle checkBA \rangle \langle partOK \rangle \langle Aprov. Rel. Parcial \rangle
% this creates 'partOK' and associates it with BA position
\checkdef \langle checkAB \rangle \langle board \rangle \langle Banca def. \rangle
\checkdef \langle checkBB \rangle \langle board-date \rangle \langle Data defesa \rangle
% 'board-date' is associated with the BB position
\checkdef \langle checkAE \rangle \langle tcc-final \rangle \langle TCC final \rangle
% 'tcc-final' is associated with the EE position
\checkdef \langle checkBE \rangle \langle approval \rangle \langle Aprovação Correções \rangle
% 'approval' is associated with the EE position
\checkdef \langle checkDE \rangle \langle exam \rangle \langle Em Exame \rangle
% 'exam' (display 'Em Exame') is associated with the DE position

Be aware that, \checkdef can and should be only used at the preamble, whereas \checklist can only be used at the 'student data definition' context (meaning, inside the \NewStudent command).

9.2.2 Global Commands II

\NewStudent \langle studentname \rangle \langle code \rangle

This is the main command describing each \langle student \rangle associated work, advisor and exam board. In \langle code \rangle one should use the commands defined in subsection 4.3, Core Specific Commands, and subsubsection 9.2.3, Student Specific
Commands (although one can use any valid \LaTeX \$2\varepsilon$ preamble code, keep in mind those will be executed BEFORE \texttt{\begin{document}}, to describe a student work. So, for instance:

\LaTeX Code:

\begin{verbatim}
\NewStudent{Artur} {
  \student{last}{first}[m]
  \studentinfo[]{243716}{email@somewhere}
  \TCCtitle{work title}
  \advisor{de Amorin}{Heraldo José}[m]
  \coadvisor{Camargo Nardelli}{Vítor}[m]
  \examinergrades{9.2}{8.5}{9.2}
  \examiner{Götz}{Marcelo}[m]
  \examinergrades{10}{9.5}{9.5}
  \examiner{Comparsi Laranja}{Rafael Antônio}
  \examinergrades{8.5}{8.5}{8}
  \altexaminer{Ventura Bayan Henrique}{Renato}
  \timeslot[Teams]{12/11}{15:30}
}

\studentFate[Dismiss]  \%% FF or Dismiss ??
\end{verbatim}

N.B. Internally, \texttt{\NewStudent} will create a command named \texttt{\studentname}, with a \texttt{hook} named \texttt{\studentname.hook} (the dot is part of the hook’s name).

9.2.3 Student Specific Commands

\texttt{\studentFate} \[(fate)\]

This assigns the \texttt{\langle fate\rangle} of a student, for those cases that one cannot rely on the ‘calculated one’ (from examiners individual grades). \texttt{\langle fate\rangle} can be either C or D (in case a student got in exam), FF for those that haven’t finished the work or ‘Dismiss’ for those that, for whatever reason, got dismissed. The default is ‘do nothing’ (no \texttt{\langle fate\rangle} assigned).

\texttt{\studenttimeslot}\[\langle local\rangle\] {\langle date\rangle} {\langle time\rangle}

\texttt{\timeslot}\[\langle local\rangle\] {\langle date\rangle} {\langle time\rangle}

\texttt{\timeslot} is just an alias of \texttt{\studenttimeslot}. They set, for the Boards Report, the \texttt{\langle local\rangle}, \texttt{\langle date\rangle} and \texttt{\langle time\rangle} in which a student will have its work presented. Those commands are meant to be used ‘inside’ a \texttt{\NewStudent} command.

\texttt{\studentTCCtitle}\[\langle title\rangle\]

\texttt{\TCCtitle}\[\langle title\rangle\]

\texttt{\studentremark}\[\langle remark\rangle\]

\texttt{\TCCtitle} is also just an alias to \texttt{\studentTCCtitle} which just ’defines’ the current student “work’s title”. \texttt{\studentremark} just inserts a \texttt{\langle remark\rangle}, which will appear in the \texttt{report}’s report (...report option).

\texttt{\DistinctBoard}\[\DefaultBoard\]

Normally, the default, it’s assumed that the student’s advisor will also be a member of the student’s exam board. For the ones in which this doesn’t holds true, one should use the \texttt{\DistinctBoard} after informing a student’s name (via
\student) and before informing its advisor name (via \advisor). For instance:

\begin{verbatim}
\NewStudent{Artur}{
\student{last}{first}[m]
\studentinfo[]{243716}{email@somewhere}
\TCCtitle{work title}
\DistinctBoard
\advisor{de Amorin}{Heraldo José}[m]
\examiner{Götz}{Marcelo}[m] % He will be the 1st examiner
\examiner{Comparsi Laranja}{Rafael Antônio} % the 2nd examiner
\examiner{Ventura Bayan Henriques}{Renato} % the 3rd examiner
}
\examinergrades
\{\langle N1\rangle\} \{\langle N2\rangle\} \{\langle N3\rangle\}
\end{verbatim}

Quite obvious, this set the grades given by an examiner (the one defined by the 'last' \examiner before this).

\begin{verbatim}
\checklist\{\langle csv-checkitems\rangle\}
\end{verbatim}

\csv-checkitems is a csv list of valid 'items' (the ones defined by \checkdef) and it will 'mark' (check) the corresponding items for a given student.

\begin{verbatim}
\addtostudent\{(student)\} \{(code)\}
\end{verbatim}

\code will be appended to the command created with \NewStudent. \{student\} must be an already defined one, whilst \{code\} can be anything valid in the context of a \NewStudent as described in subsubsection 9.2.2, Global Commands II.

\begin{verbatim}
\setreports\{(rep-list)\}
\end{verbatim}

\rep-list is a csv list of keys as defined at subsection 9.1, Package/Report Options.

\begin{verbatim}
\setstudentlist\{(listID)\} \{(list)\}
\end{verbatim}

This command will define/create a list named \{listID\} composed of a csv \{list\} of student names (as defined by \NewStudent).

\begin{verbatim}
\tcreports\{(rep-list)\} \{(listID)\}
\end{verbatim}

This is the main command, to be used only once, at the end of the file. It will typeset the reports, as set by \setreports, using the student list identified by \{listID\}. \{rep-list\} is a csv list of keys as defined at subsection 9.1, Package/Report Options.

10 ufrgscca-ppc Package (beta) (extended documentation)

This contains a set of auxiliary commands to keep track of many indicators whilst writing a PPC document (which is going to be evaluated based on said indicators, though the track of those indicators themselves shall not appear in the final version of it). Keep in mind, when considering the use of it: “it works as is” but it hasn’t being properly debugged, and it might change “as needed locally”.

The packages longtable, pdfcomment, mdframed and ufrgscca-curr will always be pre-loaded.

10.1 Package Options

showind (for drafts) it will display the report indicators, of those indicators whose family
wasn’t set to hide. (for drafts) when displaying an indicator, the long version of them will be used. (for drafts) comments (created with the command \comment{()}) will be suppressed.

10.2 Defined Commands

\maketitle
\maketitle is redefined for the specifics of a PPC document.

The next few commands use a finite set of \langle status \rangle which are a pre-defined list of:

- \texttt{tbd} “To Be Done”
- \texttt{done} “Done”
- \texttt{review} “to be reviewed”
- \texttt{attention} Needs Attention
- \texttt{NSA} NSA (portuguese for “do not apply”)
- \texttt{noref} no references
- \texttt{EAD} EAD (portuguese for “distance learning”)
- \texttt{MDi} course ware (portuguese for “didactic material”)
- \texttt{DOCs} other DOCs
- \texttt{default} everything else

\declareindicator\*++ \langle status \rangle \langle fam \rangle \{\langle ID \rangle \} \{\langle text \rangle \}
\declareindicator\*+ \langle longdesc \rangle \{\langle extra \rangle \}
\declareindicator \langle text \rangle

\declareindicator*+ is the command to create/define a given “indicator”. \langle fam \rangle set’s its family group, \langle ID \rangle is the particular ID/term used to reference it (in a family of indicators), \langle text \rangle is a short text describing it (it is the text displayed when using the \indref command). \langle longdesc \rangle (long description) and \langle extra \rangle (extra long description) to a defined \declareindicator (it will add those text fields to the “last declared one”). \langle longdesc \rangle will also be displayed when using the \indref commands, but only if the \texttt{nolong} option was used. The \langle extra \rangle will only be used/displayed with the \PrintIndicators command. Finally, \indicatorText adds a remark \langle text \rangle, which will be also printed out when using \lstind (akin of an explanation/remark field.)

\indsetstatus \{\langle status \rangle \} \{\langle fam \rangle \} \{\langle ID \rangle \}
\indsetview \{\langle fam \rangle \}
\indsethide \{\langle fam \rangle \}

\indsetstatus sets the \langle status \rangle of a given indicator defined by \langle fam \rangle and \langle ID \rangle.
\indsetview and \indsethide \langle s \rangle set the visibility (or not) of a given “family” of indicators, meaning, if those indicators are going to be visible or not (command \indref, for instance) if the option showind is in use.

\lstind \{\langle seclvl1 \rangle \} \{\langle seclvl2 \rangle \} \{\langle fam \rangle \}

\lstind will produce a sectioning like list, \langle seclvl1 \rangle defaults to \section and \langle seclvl2 \rangle defaults to \subsection, those indicators marked with an * (when creating them) will be issued with \langle seclvl1 \rangle, those marked with an + will be issued with \langle seclvl2 \rangle. The indicator’s short text will be the sectioning title, whilst the indicator’s ‘text’ (the one assigned with \indicatorText will be the sectioning body.)

\PrintIndicators \{\langle fam \rangle \}
\PrintIndicators will produce a “list of contents” like list (with cross reference to all used \indref pages). It will either issue a list of all \declareindicator or just the ones belonging to \fam. \fam can be a csv list of families. Each entry will be composed by indicator’s “family”, “ID”, “short text”, “long text” and “extra description” but not the text issued with \indicatorText.

\helpindicators
This will just prints, middle text, a quick “help text” listing the few main “indicators related command” (to help out those less \LaTeX savvy writers.)

\ifshowind
\langle\text-ifshow\rangle
\langle\text-ifnot\rangle
\ifshowind
Just a helping command, based on the package options. If the option showind was used, (code-ifshow) is executed, otherwise (code-ifnot).

\textmark \langle\status\rangle \{\text\}
\comment \langle\status\rangle \{\title\} \{\text\}
Those are annotation, remark commands. The difference being that \textmark will just highlight the \text (using \langle\status\rangle “format”), whilst \comment will create a “remark box” (the same used when inserting an indicator’s reference, commands below). N.B. \comment is suppressed unless the option showind is used.

\indref\langle\status\rangle \{\fam\} \{\ID\} \{\comment\}
\indref*\langle\status\rangle \{\fam\} \{\ID\} \{\comment\}
\indref creates a box (TikZ based mdframed) of the indicator denoted by \fam and \ID. The family and IDs will be issued as the “frame title”, the current indicator’s \langle\status\rangle will be printed out (the whole box will be highlighted accordly), the short version of the indicator will be used (the long version will “appear” as a pdfcomment), finally any \langle\comment\rangle will be added to the text box. Each \indref box will have a link to the indicator’s list (issued with \PrintIndicators). If the optional argument \langle\status\rangle is used, the indicator’s status will be updated accordly. The star version also prints the indicator’s long text. 

\indreflst\langle\status\rangle \{\fam\} \{\IDlist\} \{\comment\}
\indreflst*\langle\status\rangle \{\fam\} \{\IDlist\} \{\comment\}
\indreflst behaves similarly, with the difference that \langle\IDlist\rangle is a csv list of IDs (same family), moreover, each item of said list can have the form either just \langle\ID\rangle or \langle\status:ID\rangle, in the last form, that ID will have its status updated, as well.

\fancyquote \langle\vspc\rangle \{\text\} \{\author\} \{\dateref\}
As quick “quote” hack, \fancyquote will typesets a \text (small size, italic text, in a minipage environment) followed by \author and \dateref. This is meant to be used after a \chapter or \section commands. \langle\vspc\rangle is to be used in case one has to adjust the vertical space between the sectioning command, and the quote one.

\labelhack \{\text\}
As the name implies, it is a hack. In some cases (which we haven’t manage to found why/what), hyperref would fail miserably when using the \nameref (in some cases getting the sectioning correct, but not the name!). This just assures that \nameref will use the correct sectioning name in those cases. For Example:

\section{this section}\labelhack{this section}\label{somelabel}

\acrodef\langle\acroID\rangle acronymlong
Those are yet another acronym hack. \acrodef “creates” an acronym, identified by \acroID, whose short (acronym) version is \acronym and the long version in \long. \acro just typesets the \acronym, \acrol the \long version. \acrols typesets the long version followed by the short (using a comma as separator). \acrosl prints the short version first. Finally, \acrofoot typesets the short version in text and the long as a footnote. \printacrolist creates a description list based on the \listofabbrv environment.

10.3 Environments

\begin{ppc.quote} ... \end{ppc.quote}
This is just a tailored “quote” environment, using almost all page width, just in a smaller font size.

\classdef defines a class, associating with a \langle topicID \rangle, \langle pos \rangle (for ufrgscca-curr-graph), \langle classID \rangle, number and type, \langle typ \rangle, of credits, \langle cred \rangle, a long name, \langle name \rangle and description, \langle desc \rangle. \classremark adds an extra remark to it. The following commands always refer to the “last defined” \classdef unless \setclass is used, which changes the “current class” for the following commands.

\depdef \altdef

\depdef inserts/creates a “class dependency” list. The highlight color (if used) is usually defined by the current class topic (informing \langle topicID \rangle changes the highlight color). \langle pos \rangle is used by ufrgscca-curr-graph to determine the incident line angle.

\altdef defines/start and alternate dependency list.

\bibdef

This is used to set a list of bibliographies, one per issued command. The default \langle type \rangle value is just \textit{bib}, possible values (as understood by ufrgscca-curr-tab) are \textit{bib}, \textit{basic} and \textit{comp}.

11.2 List Processing Commands

Those are the main loop commands that go through the lists.

\LstClass \LstDep \LstTopic

\LstClass \langle cmd \rangle \{\langle semID \rangle\}
\LstDep \langle cmd \rangle \langle ang \rangle \{\langle classID \rangle\}
\LstTopic \langle cmd \rangle \{\langle topicID \rangle\}

\langle cmd \rangle can be any command accepting a single argument. It will, in fact, be the one defining the way the data will be, effectively, be presented.

\LstClass will process \langle cmd \rangle over all classes associated with \langle semID \rangle.
\LstDep will process \langle cmd \rangle over all dependency classes associated with \langle classID \rangle.
\LstTopic will process \langle cmd \rangle over all classes associated with \langle topicID \rangle.

12 ufrgscca-curr-tab Package (alpha) (extended documentation)

This is truly a work in progress (based on some old ideas), not really tested. It shall be revised and, mostly sure, it will be changed (no compatibility guaranties). It always pre-load \texttt{ufrgscca-curr} and \texttt{longtable}.

12.1 Tabular Presentation Commands

\TabEtp \TabTopic

\TabEtp \langle type \rangle \{\langle sectioning \rangle\}c\{\langle semID \rangle\}
\TabTopic \langle type \rangle \{\langle topicID \rangle\}

\TabEtp will construct a longtable with all classes associated with \langle semID \rangle (including it’s dependencies and bibliography).
\TabTopic will construct a longtable with all classes associated with \langle topicID \rangle.

13 ufrgscca-curr-graph Package (alpha) (extended documentation)

13.1 Graph Presentation Command

Ironically, this is the “oldest” of the -curr- packages, but it is the less tested one, and the one whose code is more prone to fail in unexpected ways, be
advised: do not try to use it, unless you know the internal code well. It always pre-load \texttt{ufrgscca-curr} (N.B. it also depends on \texttt{tikz}).

\GraphSem \[(\text{type}) \{\text{semID}\} \]
It will create a dependency graph for a given (semId). N.B. to start with, it is highly dependent on the semester sequence, one shall start with first semester and go from there.