

The MRTbundle

L^AT_EX Templates for the MRT, University of Bayreuth

from

Jonathan P. Spratte

Karl-Theodor zu Guttenberg

27th July 2018

Contents

	Seite
1 Introduction	1
1.1 Feature Requests and Bug Reports	1
1.2 Individual Versions	1
2 The MRTthesis class	2
2.1 Options	2
2.1.1 Load time options	2
2.1.2 Setup options	3
2.2 Macros	7
2.3 Dependencies	9
3 The MRTbeam class	10
3.1 Random chatter	10
3.2 Frame contents	10
3.3 Options	11
3.3.1 Footnote related	12
3.3.2 Bibliography related	13
3.4 Macros	13
3.4.1 Footnote related	14
3.4.2 Bibliography related	15
3.5 Dependencies	16
4 The MRTalone class	17
4.1 Options	17
4.1.1 Load time options	17
4.1.2 Setup options	17
4.2 Macros	18
4.3 Dependencies	18
5 The MRTtab package	19
5.1 The MRTtabular environment	19
5.1.1 Known Bugs	20
5.2 The \MRTcline macro	20
5.3 The MRTtable environment	22
5.4 Explicit head rows	22
5.5 Other package macros	23
5.6 Options	23
5.6.1 Setup Options	23
5.6.1.1 longtable related options	26
5.7 Dependencies	26
6 The MRTif package	27
6.1 Macros	27
6.2 Dependencies	28

7 The MRTwuline package	29
7.1 Options	29
7.2 Macros	29
7.3 Dependencies	29
8 The MRTsfacc package	30
8.1 Dependencies	31

1 Introduction

This bundle provides two \LaTeX classes, one for theses and one for presentations, which both aim to match the corresponding MS Office templates of the Chair of Measurement and Control Engineering (Lehrstuhl für Mess- und Regeltechnik; MRT) of the University of Bayreuth, hence the name. Along the two major classes `MRTthesis` and `MRTbeam` there are minor auxiliary packages contained in this distribution.

This bundle makes no claim to be complete, comprehensive, or correct. For formatting errors I don't take any responsibility. Each author takes full liability for his work and its formatting.

You're allowed to share this work with fellow students working at the MRT, though official distribution channels might be better suited as they assure up to date versions.

I'd feel guilty distributing this bundle without saying the following: I'm not responsible for the overall look of this. I tried to match the Word template of the institution where possible and as a result, this is non-optimal typography, in my humble opinion.

Of course this documentation is created with one of the provided classes, namely `MRTthesis`, in use.

If you're not yet familiar with \LaTeX you should stop reading at this point (meaning the end of this paragraph) and either read a *good* and *up-to-date* introduction to \LaTeX and afterwards read on or use MS Word for your thesis. Personally I think the time reading an introduction in order to use \LaTeX is well spend, but there certainly are different opinions on that – unfortunately opinions are prone to be biased, mine is no exception. A viable introduction is `lshort` which is available in several languages at the following link: <https://www.ctan.org/pkg/lshort>

1.1 Feature Requests and Bug Reports

You can request features or report bugs at gitlase: <https://gitlase.de/jonathan/MRTbundle>

Alternatively you can request features or report bugs if you find some via email: mrt_depp@yahoo.de. Please use a descriptive subject containing "MRTbundle" (e.g. "MRTbundle – bug report").

1.2 Individual Versions

Package	Date	Version
<code>MRTthesis</code>	2018/07/26	0.0.3b
<code>MRTbeam</code>	2018/06/01	0.0.1
<code>MRTalone</code>	2018/06/03	0.0.1
<code>MRTtab</code>	2018/07/26	0.0.2b
<code>MRTif</code>	2018/06/01	0.0.1
<code>MRTwuline</code>	2018/06/01	0.0.1
<code>MRTsfacc</code>	2018/06/01	0.0.1

2 The MRTthesis class

MRTthesis provides the template to write a thesis at the MRT. It sports a layout which looks confusingly similar to the MS Word template provided by the chair. Of course there are some minor differences and the typesetting algorithm of TeX should create better line breaking than Word's but if one doesn't know on what to pay attention or for an untrained eye the distinction won't be possible (at least I hope so, as that was the goal in the first place).

2.1 Options

2.1.1 Load time options

The class features a few load time options, none of which do take arguments.

<u>longtable</u>	–NoArgument– Is forwarded to MRTtab see its description in section 5.6.
<u>mathsizes</u>	–NoArgument– Opposite of <code>no mathsizes</code> . If used (which it by default is) the maths sizes are set according to the MS Word template. Note that those weren't set by <code>mrtarbeit</code> and if you alter the default font size won't be set.
<u>minimal</u>	–NoArgument– If this option is passed some packages are not loaded and therefore related configurations not set. See section 2.3.
<u>no geometry</u>	–NoArgument– If this option is passed the <code>geometry</code> package is not loaded (and of course the page dimensions passed to <code>geometry</code> otherwise are not set).
<u>no mathsizes</u>	–NoArgument– Opposite of <code>mathsizes</code> . If used the maths sizes are not changed from the defaults of <code>scrreprt</code> .
<u>tikzunderline</u> <u>tUline</u>	–NoArgument– This option is forwarded to MRTwuline.
<u>british</u> <u>english</u> <u>UKenglish</u>	–NoArgument– If used the document will be using the <code>british</code> definition of <code>babel</code> . Many strings used in the package will be in English, but some might be missed out. If you find any of which you think should be translated, please contact me as described in section 1.1. English simplified (US) is not supported by the class.

Every other given option will be passed on to `scrreprt`.

2.1.2 Setup options

The following options are accessible with \MRTthesisSetup.

<u>advisor</u>	= $\langle name \rangle$ Sets the name of the advisor of this thesis. One typical value could be Dipl.-Ing. Alice Fischerauer
<u>author</u>	= $\langle name \rangle$ Sets the name of the author or authors as the macro \author does. Separate authors with \and. You can give the surname first followed by a comma and the given name, in which case the parsing for the abbreviation works better (especially with name affixes). The following two options are fine: author={Duck, Donald \and Mouse, Mickey} or author={Donald Duck \and Mickey Mouse} and should both result in the abbreviation D. Duck, M. Mouse. Another example would be zu Guttenberg, Karl-Theodor or Karl-Theodor zu Guttenberg. Here the parsing would result in K.-T. zu Guttenberg or K.-T. z. Guttenberg – the first one seems correct, the second one fails. Remember to surround the argument with braces if you use a comma.
<u>degree</u>	= $\langle degree \rangle$ The degree you aim to achieve with the thesis. If you don't use this option it is tried to be guessed from the type of thesis you can specify with the thesis key. An error is thrown if the degree can't be guessed. If you don't want to achieve any degree, use the option no degree. Typical values would be Bachelor of Science or Master of Science.
<u>caption above</u>	–NoArgument– Is forwarded to MRTtab. See subsection 5.6.1.
<u>caption below</u>	–NoArgument– Is forwarded to MRTtab. See subsection 5.6.1.
<u>examiner</u>	= $\langle name \rangle$ The examiner of the thesis. Initial value is Univ.-Professor Dr.-Ing. Gerhard Fischerauer.
<u>logoL</u>	= $\langle file \rangle$ The image file for the left logo on the titlepage. Initial value is MRTthesis_logo_UBT2.pdf. If $\langle file \rangle$ is an empty argument no left logo will be used.
<u>logoL height</u>	= $\langle dimen \rangle$ The height the left logo is displayed in. Initial value is 10.85mm.

<u>logoR</u>	= $\langle file \rangle$ The image file for the right logo on the titlepage. Initial value is MRTthesis_logo_MRT2.pdf. If $\langle file \rangle$ is an empty argument no right logo will be used.
<u>logoR height</u>	= $\langle dimen \rangle$ The height the right logo is displayed in. Initial value is 11.9mm.
<u>no advisor</u>	= $\langle bool \rangle$ If true no advisor will be displayed on the title page. Default is true, initially is false.
<u>no citation</u>	= $\langle bool \rangle$ If true no citation indications are displayed at the bottom of the title page. Default is true, initially is false.
<u>no degree</u>	= $\langle bool \rangle$ If true no degree will be displayed on the title page. Default is true, initially is false. Also the paragraph corresponding to the degree in the affidavit will be left out.
<u>no examiner</u>	= $\langle bool \rangle$ If true no examiner will be displayed on the title page. Default is true, initially is false.
<u>no chair</u>	= $\langle bool \rangle$ If true no chair will be displayed on the title page. Default is true, initially is false.
<u>no logos</u>	–NoArgument– If used logoL={},logoR={} is used, which results in no logos on the title page.
<u>no thesis</u>	= $\langle bool \rangle$ If true no thesis type will be displayed on the title page. Default is true, initially is false.
<u>no usage</u>	= $\langle bool \rangle$ If true no usage rights are given to the MRT in the affidavit text. Default is true, initially is false. If you need a custom paragraph and don't want to leave it out completely you should redefine \affidavitttext@usagerights.
<u>number</u>	= $\langle number \rangle$ The MRT report number displayed in the citation indications. Initially is empty.

The typical pattern of these numbers is something like: TT-yy-mm-n with TT the thesis type, e. g. BA or MA, yy the last two digits of the year, mm the month, and n the number of the thesis in this month.

`pos figure` = $\langle placement \rangle$

The $\langle placement \rangle$ of floats of type figure.

`pos float` = $\langle placement \rangle$

The $\langle placement \rangle$ of floats of both types, figure and table. Initially set to `tbp`.

`pos MRTtable` = $\langle placement \rangle$

The $\langle placement \rangle$ of floating MRTtables, forwarded to MRTtab's option `pos`. See subsection 5.6.1.

`pos table` = $\langle placement \rangle$

The $\langle placement \rangle$ of floats of type table.

`short advisor` = $\langle abbreviation \rangle$

`sadvisor` The abbreviated name of the advisor. This is needed for the citation indications and not parsed automatically from the `advisor`, as the name contains academic titles, but the abbreviation should not and the parsing would be hard to do correctly.

`short author` = $\langle abbreviation \rangle$

`sauthor` The abbreviated name or names of the author or authors. If you don't use this option it is tried to parse those automatically. If the parsing does something wrong you'll have to use this option giving the correct abbreviations with each name separated with commas from the others, e. g. `short author={D. Duck, M. Mouse}`.

`short examiner` = $\langle abbreviation \rangle$

`sexaminer` The abbreviated name of the examiner. This is needed for the citation indications and not parsed automatically from the `examiner`, as the name contains academic titles, but the abbreviation should not and parsing would be hard to do correctly. Initially value is `G. Fischerauer`.

`sign height` = $\langle dimen \rangle$

The height reserved for each signature below the affidavit text. Initial value is 9mm.

`sign seperation` = $\langle dimen \rangle$

`sign sep` If `sign width max` is not given (or 0pt) the maximum width is calculated from the text width and the width of the date and location. The minimum distance

from the date and location to the signature lines is then enforced to be at least $\langle \textit{dimen} \rangle$. Initial value is 2em.

`sign width max` = $\langle \textit{dimen} \rangle$

You can enforce a maximum width for the signature lines below the affidavit using this option. If it is not used, the maximum width is calculated.

`sign width min` = $\langle \textit{dimen} \rangle$

You can enforce a minimum width for the signature lines using this option. Initially this is set to 7cm.

`stretch caption` = $\langle \textit{float} \rangle$

`stretch cap` Uses `\setkomafont` to enforce a specific line spread using `\setstretch` for captions.

`stretch tabular` = $\langle \textit{float} \rangle$

`stretch tab` Is forwarded to MRTtab's option `stretch tab`. See subsection 5.6.1.

`stretch text` = $\langle \textit{float} \rangle$

Uses `\setstretch` to set a specific line spread in the document.

`stretches` = $\langle \textit{float} \rangle$

Sets `stretch cap`, `stretch tab`, and `stretch text` in one go. Initially set to 1.408.

`subtitle` = $\langle \textit{title} \rangle$

The title page might include a subtitle. If you really want to use it, you'd have to use with `subtitle`. You can also use `\subtitle` to set it.

`thesis` = $\langle \textit{thesis type} \rangle$

Sets the $\langle \textit{thesis type} \rangle$. Typical arguments would be `Bachelorarbeit` or `Bachelor Thesis` (the former if you're writing in German, the latter if you're writing in English).

`title` = $\langle \textit{title} \rangle$

Sets the title of the thesis. You might also use `\title` to set this.

`toc ChapIndent` = $\langle \textit{dimen} \rangle$

Sets the indentation of chapter entries in the table of contents. Initially set to 0.01em.

`toc SecIndent` = $\langle \textit{dimen} \rangle$

Sets the indentation of section entries in the table of contents. Initially set

to 1.32em. The width is also used for entries in the list of figures and list of tables.

`toc sSecIndent` = $\langle dimen \rangle$

Sets the indentation of subsection entries in the table of contents. Initially set to 3.38em.

`toc ssSecIndent` = $\langle dimen \rangle$

Sets the indentation of subsubsection entries in the table of contents. Initially set to 6.38em.

`with subtitle` = $\langle bool \rangle$

If true a subtitle can be used on the title page. Default is true, initially is false.

2.2 Macros

The following macros are provided:

`\ifNoWidthTF` Usage: `\ifNoWidthTF{ $\langle arg \rangle$ }{ $\langle true \rangle$ }{ $\langle false \rangle$ }`

Typesets the argument in a box (so the code is actually executed). If the produced box has a width of 0pt the $\langle true \rangle$ branch is executed, else the $\langle false \rangle$ branch.

`\vfillmult` Usage: `\vfillmult{ $\langle num \rangle$ }`

Same as if you'd $\langle num \rangle$ instances of `\vfill`.

`\hfillmult` Usage: `\hfillmult{ $\langle num \rangle$ }`

Same as if you'd $\langle num \rangle$ instances of `\hfill`.

`\MRTafterhyperref` Usage: `\MRTafterhyperref{ $\langle content \rangle$ }`

Places $\langle content \rangle$ after `hyperref` is loaded. This is important for the relative few packages that need to be loaded after `hyperref`. So if you have one of these, you should use something like `\MRTafterhyperref{\usepackage{cleveref}}`. This macro has to be used prior to `\begin{document}`.

`\DeclareTOCStyleEntryMRTChapterLike` Usage: `\DeclareTOCStyleEntryMRTChapterLike[$\langle indent \rangle$]{ $\langle entry-layer \rangle$ }`

See the description of `\DeclareTOCStyleEntryMRTSectionLike`. The difference is that this sets the entries how the chapters are formatted. Also the $\langle indent \rangle$ defaults to the one of chapters.

`\DeclareTOCStyleEntryMRTSectionLike` Usage: `\DeclareTOCStyleEntryMRTSectionLike[$\langle indent \rangle$]{ $\langle entry-layer \rangle$ }`

The macro calls the KOMA macro `\DeclareTOCStyleEntry` and sets the

entry-layer	macro name	default length
chapter	<code>\l_MRTthesis_toc_chapter_indent_tl</code>	0.01em
section	<code>\l_MRTthesis_toc_section_indent_tl</code>	1.32em
subsection	<code>\l_MRTthesis_toc_subsection_indent_tl</code>	3.38em
subsubsection	<code>\l_MRTthesis_toc_subsubsection_indent_tl</code>	6.38em
table	<code>\l_MRTthesis_toc_section_indent_tl</code>	1.32em
figure	<code>\l_MRTthesis_toc_section_indent_tl</code>	1.32em

Table 2-1: Indents of different ToC entry types and the macros they are stored in. Use the options described in subsection 2.1.2 to change the values.

options how they are used for the section entries in the table of contents. `<indent>` defaults to the indent length of section entries. It is possible to use an `<indent>` but give more options in that optional argument afterwards (comma separated). Table 2-1 shows an overview of the class's default indentations.

`\MRTthesisSetup`

Usage: `\MRTthesisSetup{<options>}`

You can use this macro to set the options listed in subsection 2.1.2.

`\sethead`

Usage: `\sethead{<content>}`

Sets the head marks for both sides to `<content>`. It is the same as `\markboth{<content>}{<content>}`. You might use this (or any similar macro provided by KOMA script) to manually set the head marks, e. g. if your section title gets too long.

`\affidavit`

Usage: `\affidavit`

Prints a chapter "Eidesstattliche Erklärung" and the affidavit text (as stored in `\affidavit@text`) and the location and date, followed by a signature line for each author. `\maketitle` has to be used prior to it, else the lines won't be printed. This is a bug I might fix in the future.

`\mainpart`

Usage: `\mainpart`

`\mainmatter`

Switches the formatting from the one at the beginning to the one used in the main part of the document. Should be used after `\tableofcontents`, `\listoffigures`, and `\listoftables`.

`\appendix`

Usage: `\appendix`

Switches the formatting to the one used in the appendix. This includes switching to alphabetically numbered sections and setting the option `no float` in `\MRTtabSetup`.

Additionally the macros `\author`, `\title`, and `\subtitle` have been redefined to internally use `\MRTthesisSetup` to set the corresponding options.

2.3 Dependencies

As this class is based on `scrreprt`, it depends on that class and all of its dependencies, of course. Additionally the following packages are loaded (used options given in brackets). Those are quite some but unfortunately most of these are required (or help a lot) to achieve certain formattings in order to match the MS Word template of the MRT best.

Some of the used packages are not necessarily needed to match the MS Word template, but provide useful features – e.g. `hyperref` which allows the use of `\autoref` and cross linking but is not needed to match any specific formatting.¹

- `expl3`
- `xparse`
- `MRTif`
- `MRTsfacc`
- `MRTtab`
- `MRTwuline`
- If \LaTeX or \LuaTeX are used
 - `fontspec` and `\defaultfontfeatures` `{Ligatures=TeX}`
- else
 - `fontenc` `[T1]`
 - `inputenc` `[utf8]`
- `lmodern`
- `helvet`
- `babel` `[ngerman]` or if `british` is used with `[main=british, ngerman]`
- `scrlayer-scrpage` `[singlespacing=true]`
- `geometry` (with correct options)
- `setspace`
- `xcolor`
- `graphicx`
- `enumitem`
- `mathtools`
- `mathastext` `[italic,defaultmathsizes]`
- `isomath`
- `hyperref`
- if the `minimal` option is not used:
 - `siunitx` `[detect-all, per-mode=reciprocal-positive-first]`
If `babel`'s `british` is used `[locale=UK]` will be used, if `ngerman` `[locale=DE]`. Additionally the `range-phrase` will be set to either `to` or `bis` with spaces around it.
 - `biblatex` `[backend=biber, natbib=true, citestyle=numeric, bibstyle=numeric, sorting=none, giveninits=true, sortcites]`
 - `csquotes`

¹Don't remove it though, the current code for section headings relies on it.

3 The MRTbeam class

The MRTbeam class is a class build upon beamer. It should mimic the style of the MS Powerpoint template of the MRT which was in use when I held my Bachelor's presentation. I heard the requirements to match a specific template are less strict today, but at least I'll still use this template.

Many of the features described here are also available if one uses `\usetheme{MRTbeam}` within a document using the `beamer` class. There however is no dedicated documentation for that possibility provided. You're encouraged to also use the corresponding MRTbeam class if your using the eponymous theme.

If there is a new institution template which should be matched that doesn't match this beamer template please contact me as described in section 1.1.

3.1 Random chatter

The creation of a presentation using beamer is not everyone's cup of tea. Refer to the beamer manual to get a basic idea of how to use it, as MRTbeam only adds some stuff that is not basic beamer stuff. The main idea of creating a presentation remains that of beamer.

MRTbeam doesn't follow the way beamer does things everywhere. As a result some stuff may not work out as you expect if you're used to beamer. Especially the customization might require you to actually read the sources of MRTbeam and its beamer themes. Also MRTbeam is only for presentation mode as of now.

Special thanks are due to the TeX.SX user samcarter, who helped me ram my head through beamer's walls in order to get my will.

3.2 Frame contents

The class builds up frames as shown in Figure 3-1 (not true to scale).

You can specify the used logos using `\uselogo`. The default is the UBT logo on the left, no logo in the centre, and the MRT logo on the right side.

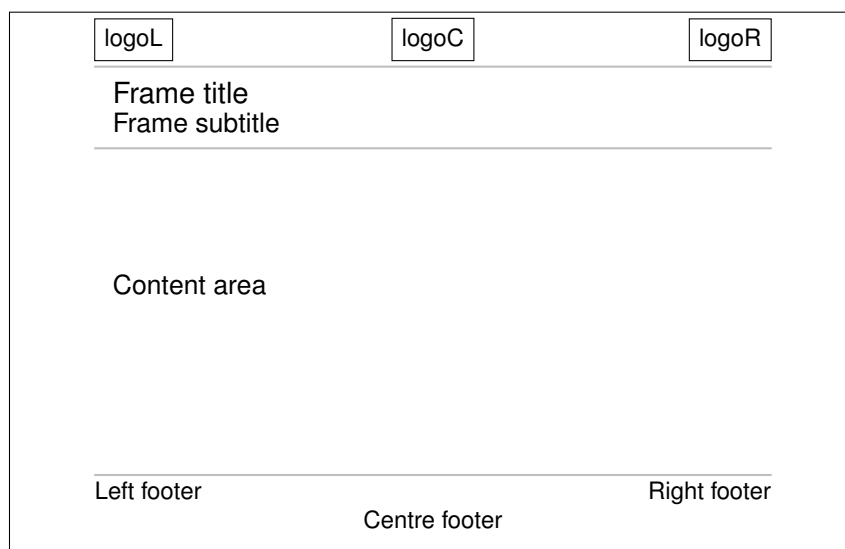


Figure 3-1: The basic layout of a frame in MRTbeam

If you specify no or an empty frame title the current section is used (with its numbering). The frame's subtitle can be prepended with the current subsection (with its numbering) followed by a colon. This depends on the current value of `\ifPrependSubsections`.

The left footer contains the `occasion`, the `shorttitle`, and the `shortauthor`. If no short title or no short author is given, the title and the author, respectively, are used instead. If you give a `*` for the short author or the short title, they are left out (e.g. with using `\title[*]{foo}`).

The centre footer contains the frame number and if you want a progress bar. The progress bar is shown if `\ifProgressBar` is true.

In the right footer the following is displayed atop of each other: Persistent MRT footnotes, volatile MRT footnotes, citation MRT footnotes, normal footnotes. The right footer has enough space for three entries. If you need more they are scaled to the available vertical space. MRT footnotes might be displayed in a tabular manner with the labels in the left and the actual notes in the right column. This depends on the value of `\ifTabularNotes`.

Neither of the footers is restricted in horizontal size. As a result they might overlap if you specify really long contents.

3.3 Options

The class passes all options given to it on to `beamer`. There are still some class specific options which you can set with some macros. The macros in this section are only provided to set specific options, other macros are described in section 3.4.

<u><code>\advisor</code></u>	<p>Usage: <code>\advisor{*}[\langle title \rangle]{\langle name \rangle}</code></p> <p>Sets <code>\langle name \rangle</code> as the current advisor. It also redefines itself, any consecutive call will not take any arguments but return the <code>\langle name \rangle</code>. The <code>\langle title \rangle</code> shall be the title used on the title frame defaulting to 'Betreuerin' if the starred version is used, else it defaults to 'Betreuer'.</p>
<u><code>\occasion</code></u>	<p>Usage: <code>\occasion{\langle occasion \rangle}</code></p> <p>Defines the occasion of the presentation. If used the occasion will be displayed in the left footer.</p>
<u><code>\uselogo</code></u>	<p>Usage: <code>\uselogo{\langle pos \rangle}[\langle options \rangle]{\langle file \rangle}</code></p> <p>Specifies the logo used at the position <code>\langle pos \rangle</code>. There are l, c, and r available. The <code>\langle file \rangle</code> is included using <code>\includegraphics</code> with the specified <code>\langle options \rangle</code> (defaulting to <code>height=0.056\paperwidth</code>). If <code>\langle file \rangle</code> is an empty argument there is no logo used at the specified position. By default <code>MRTbeam_logo_UBT2.pdf</code> is used for the left logo and <code>MRTbeam_logo_MRT2.pdf</code> for the right one. The centre logo is initially empty.</p>
<u><code>\ShowGrid</code></u>	<p>Usage: <code>\ShowGrid[\langle options \rangle]</code></p> <p>Globally activates a <code>TikZ</code> grid displayed in the background of the frames. You can specify the <code>TikZ</code>-style used for the grid with <code>\langle options \rangle</code>. The default is: <code>xstep=.05\paperwidth, ystep=.1\paperheight, help lines</code>.</p>

\HideGrid Usage: \HideGrid<*>

Globally deactivates the background grid and restores the package's default options for that grid. If the starred version is used, the options are not reset.

\ifPrependSubsections Usage: \ifPrependSubsections

\PrependSubsectiontrue If set true each frame's subtitle is prepended by the current subsection.

\PrependSubsectionsfalse

\ifOnlyOneTopRule Usage: \ifOnlyOneTopRule

\OnlyOneTopRuletrue If set true in each frame the title and subtitle will not be displayed and the lower top rule will be omitted, significantly enlarging the content area. If you use \OnlyOneTopRuletrue or \OnlyOneTopRulefalse \contentheight will be adjusted.

\OnlyOneTopRulefalse

\ifProgressBar Usage: \ifProgressBar

\ProgressBartrue If set true a progress bar will be shown in the middle of the slides foot at the frame number. You can customize the progress bar shown using \SetProgressBar or \ProgressBarStyle.

\ProgressBarfalse

\SetProgressBar Usage: \SetProgressBar<*>{<align>}{<length>}{<height>}{<voffset>}

This changes the default values of \ProgressBar. If the starred version is used the changes are made locally, else they are applied globally. Take a look at the description of \ProgressBar for an explanation what each of the parameters mean. If you use a * as one of the arguments the corresponding default value will remain unchanged.

\ProgressBarStyle Usage: \ProgressBarStyle<*>{<style>}

This sets the progress bar options to a predefined <style> using the unstarred version of \SetProgressBar. If the starred version of \ProgressBarStyle is used \ProgressBartrue is issued.

style	align	length	height	voffset	description
default	c	30pt	font size	-1.65ex	a thick and relatively short bar around the frame number
Spratte	c	\paperwidth	2pt	3pt	A thin line spanning the whole page width at the bottom of the frame

3.3.1 Footnote related

\ifTabularNotes Usage: \ifTabularNotes

\TabularNotetrue If set true the MRT footnotes will be displayed in a tabular manner with two columns. MRT footnotes are those footnotes set with the footnote related macros in subsection 3.4.1.

\TabularNotesfalse

<hr/> <code>\ColumnsTabularNotes</code>	Usage: <code>\ColumnsTabularNotes{<specification>}</code> With this macro you can specify the column specifications used by MRT footnotes. Your definition should contain two columns.
---	---

3.3.2 Bibliography related

<hr/> <code>\ifExplicitCiteOnce</code> <code>\ExplicitCiteOncetrue</code> <code>\ExplicitCiteOncefalse</code> <hr/>	Usage: <code>\ifExplicitCiteOnce</code> If set to true for every used key the citation is in an explicit manner only once. For each following citation of the same key only the number is used.
<hr/> <code>\ifNoExplicitCite</code> <code>\NoExplicitCitetrue</code> <code>\NoExplicitCitefalse</code> <hr/>	Usage: <code>\ifNoExplicitCite</code> If set to true there will never be an explicit citation at the frame, only the citation number will be used.

3.4 Macros

<hr/> <code>\PlaceAt</code> <hr/>	Usage: <code>\PlaceAt{<*>}<pos>[<node options>]{<content>}</code> The starred version differs fundamentally from the unstarred one. The unstarred one places <code><content></code> at the specified position <code><pos></code> in the background inside a <code>TikZ</code> node with the optionally specified <code><node options></code> . The coordinates default to multiples of <code>\pagewidth</code> and <code>\pageheight</code> for <code>x</code> and <code>y</code> , respectively. You can use anything <code>TikZ</code> understands as coordinates for <code><pos></code> . The starred version places the <code>tikzpicture</code> where you currently are. It uses <code>remember picture</code> and <code>overlay</code> as options. The <code><pos></code> must match the pattern <code><(x),<y)></code> . <code><x></code> is in multiples of <code>\pagewidth</code> and <code><y></code> in multiples of <code>\pageheight</code> and you can't change that. The node still gets <code><node options></code> . In both cases <code>(0,0)</code> is the bottom left corner of the slide.
<hr/> <code>\AddToPlaced</code> <hr/>	Usage: <code>\AddToPlaced{<tikz code>}</code> Adds the specified <code><tikz code></code> to the background of the current slide. <code>(0,0)</code> is the bottom left corner of the slide. Coordinates are by default in multiples of <code>\pagewidth</code> and <code>\pageheight</code> . It uses the same <code>tikzpicture</code> as <code>\PlaceAt</code> and is stored in the same macro.
<hr/> <code>\ProgressBar</code> <hr/>	Usage: <code>\ProgressBar[<align>]<length>>[<height>][<voffset>]</code> Prints a progress bar. <code><align></code> is the horizontal alignment as you would pass it to a <code>\makebox</code> , the initial default is <code>c</code> . <code><length></code> is the overall length the progress bar should have (defaulting to 30pt), <code><height></code> its height, defaulting to the current font size. <code><voffset></code> allows you to offset the progress bar vertically. With positive values the shift is downwards, the default is <code>-1.65ex</code> . The progress bar uses the <code>xcolor</code> colours <code>progressed</code> and <code>noprogress</code> . All arguments are optional.

<u>\StartOfProgress</u> <u>\EndOfProgress</u>	<p>Usage: \StartOfProgress</p> <p>Denotes the start and end of the progress bars gauge. The frame after the call of \StartOfProgress is the first frame filling the gauge, the frame prior to \EndOfProgress is the first frame in which the gauge is fully filled. The macros should be used outside of the frame environment. If \StartOfProgress is not used the first frame starts filling the gauge, if \EndOfProgress is not used the last frame is the only one with a completely filled gauge.</p>
<u>\contentwidth</u> <u>\contentheight</u>	<p>Usage: \contentwidth</p> <p>These are lengths which are set to match the height and width of the content block of a frame (the space between the bottom rule and the lower top rule). \textwidth should match \contentwidth if you're outside of a minipage or similar, but \textheight will most likely not match the actual height of the content area.</p>
<u>\UseAndIfEmptyTF</u>	<p>Usage: \UseAndIfEmptyTF[⟨pre⟩]{⟨arg⟩}{⟨true⟩}{⟨false⟩}</p> <p>The ⟨arg⟩ is expanded inside a box. If that box has a width not equal 0pt ⟨pre⟩ is used followed by the contents of the box. Then the ⟨false⟩ branch is executed. If the box's width equals 0pt the ⟨true⟩ branch is used instead and neither ⟨pre⟩ is used nor the box containing ⟨arg⟩ placed.</p>
<u>\cursec</u>	<p>Usage: \cursec⟨*⟩</p> <p>If the current section is starred or you used the optional * for \cursec, this macro inserts the current sections name, else the name is prepended by the current sections number.</p>
<u>\curssec</u>	<p>Usage: \curssec⟨*⟩</p> <p>This macro is very similar to \cursec. If you used the starred version of it or the current subsection is starred, this macro inserts the current subsections name, else the name is prepended by the current subsections number.</p>
<u>whiteframes</u>	<p>Usage: \begin{whiteframes} ... \end{whiteframes}</p> <p>In this environment \ifwhiteframes is set true.</p>

3.4.1 Footnote related

The term footnotes relates to the special MRT footnotes in this section.

<u>\AddToRightFoot</u>	<p>Usage: \AddToRightFoot⟨*⟩⟨+⟩⟨overlay⟩[⟨pre⟩]{⟨note⟩}</p> <p>This macro adds stuff to the right footer. If ⟨*⟩ is given, the content is added to the persistent footnotes, else if ⟨+⟩ is given added to the cite related footnotes, else to the ordinary ones. ⟨overlay⟩ is used for any overlay specifications using \uncover. ⟨pre⟩ is added left to ⟨note⟩. If tabular footnotes are used ⟨pre⟩ is in the left, ⟨note⟩ in the right column. If tabular footnotes are not</p>
------------------------	--

used the distance between $\langle pre \rangle$ and $\langle note \rangle$ is 0.5\tabcolsep . The starred variant should only be used outside of the `frame` environment. If you get strange errors during compilation a `\noexpand` in front of some macros (e.g. stuff like `\href`) you give as arguments might help.

\ClearRightFoot Usage: `\ClearRightFoot<*>`

Clears the footnotes. If the `*` is given only the volatile footnotes are cleared, else all of them.

3.4.2 Bibliography related

\cite Usage: `\cite<overlay>>[<opt1>][<opt2>]{<key>}`

$\langle overlay \rangle$ is handled by `\uncover`, which affects only the footnote not the footnote mark. The usage of the two optional arguments and $\langle key \rangle$ match those known from biblatex's `\cite`. The citation's contents are dependent on `\ifNoExplicitCite` and `\ifExplicitCiteOnce`, an explicit citation contains the citation number, authors' names, the journal, and the year.

\framecite Usage: `\framecite<*><overlay>>[<pre>]{<key>}[<post>]`

Places a citation in the footnotes, if the starred version is used using the persistent footnotes, else the volatile non-cite related footnotes. Use the starred version prior to the `frame` it should first be shown in. $\langle overlay \rangle$ specifications are interpreted by `\uncover`. $\langle pre \rangle$ is put in front of the citation with a distance of `\`, (unaffected by tabular footnotes options), $\langle post \rangle$ with a distance of `\`, after the citation. The citation contains the authors' names, the journal, and the year.

\bibliographyframe Usage: `\bibliographyframe<*>[<bibfont>]{<title>}{<subtitle>}`

Prints the bibliography. The starred variants uses `\whiteframestrue`. The $\langle bibfont \rangle$ defaults to `\small`, you might give any font related commands here. Both $\langle title \rangle$ and $\langle subtitle \rangle$ are optional though delimited by curly braces. $\langle title \rangle$ defaults to 'Quellen', $\langle subtitle \rangle$ is initially empty. The `\bibliographyframe` is printed using `allowframebreaks`.

\inlinecite Usage: `\inlinecite[<opt1>][<opt2>]{<key>}`

Gives the citation which would be placed in the in the text by `\cite` without any `\textcolor`. In fact `\cite` uses this internally.

\insertcite Usage: `\insertcite{<key>}`

Gives the citation which would be placed in the footnote by `\cite`. In fact `\cite` uses this internally.

\insertframecite Usage: `\insertframecite{<key>}`

Gives the citation which would be placed in the footnote by `\framecite`. In

fact `\framecite` uses this internally.

3.5 Dependencies

The class uses `beamer` as its basis. Additionally the following packages are loaded:

- `helvet`
- `xparse`
- `biblatex`
- `MRTsfacc`
- `TikZ`

`biblatex` uses `biber` as its backend.

4 The MRTalone class

The standalone version of MRTthesis. The aim is to provide a class to produce simple L^AT_EX based images which match the look of MRTthesis.

4.1 Options

4.1.1 Load time options

The class features a few load time options, none of which do take arguments.

<u>longtable</u>	–NoArgument– Is forwarded to MRTtab see its description in section 5.6.
<u>mathsizes</u>	–NoArgument– Opposite of no mathsizes. If used (which it by default is) the maths sizes are set according to the MS Word template. Note that those weren't set by mrtarbeit and if you alter the default font size won't be set.
<u>minimal</u>	–NoArgument– If this option is passed some packages are not loaded and therefore related configurations not set. See section 2.3.
<u>no mathsizes</u>	–NoArgument– Opposite of mathsizes. If used the maths sizes are not changed from the defaults of scrreprt.
<u>tikzunderline</u> <u>tUline</u>	–NoArgument– This option is forwarded to MRTwuline. See its description in section 7.1.
<u>british</u> <u>english</u> <u>UKenglish</u>	–NoArgument– If used the document will be using the british definition of babel. Many strings used in the package will be in English, but some might be missed out. If you find any of which you think should be translated, please contact me as described in section 1.1. English simplified (US) is not supported by the class.

Every other given option will be passed on to standalone.

4.1.2 Setup options

The following options are accessible with \MRTaloneSetup.

<u>caption above</u>	–NoArgument– Is forwarded to MRTtab and its \MRTtabSetup. See its description in subsection 5.6.1.
----------------------	---

<hr/> <code>caption below</code> <hr/>	<code>–NoArgument–</code> Is forwarded to <code>MRTtab</code> and its <code>\MRTtabSetup</code> . See its description in subsection 5.6.1.
<hr/> <code>stretch caption</code> <code>stretch cap</code> <hr/>	<code>= <float></code> Currently does nothing.
<hr/> <code>stretch tabular</code> <code>stretch text</code> <hr/>	<code>= <float></code> Is forwarded to <code>MRTtab</code> and its <code>\MRTtabSetup</code> . See its description in subsection 5.6.1.
<hr/> <code>stretch text</code> <hr/>	<code>= <float></code> Uses <code>\setstretch</code> to set a specific line spread in the document.
<hr/> <code>stretches</code> <hr/>	<code>= <float></code> Sets <code>stretch cap</code> , <code>stretch tab</code> , and <code>stretch text</code> in one go. Initially set to 1.408.

4.2 Macros

<hr/> <code>\MRTaloneSetup</code> <hr/>	Usage: <code>\MRTaloneSetup{<options>}</code> You can use this macro to set the options listed in subsection 4.1.2.
---	--

4.3 Dependencies

The class is based on `standalone`, therefore it naturally depends on that and all its dependencies. Additional dependencies are:

- `expl3`
- `xparse`
- `MRTtab` for which `in text sep` is set to `0pt` and the option `no float` is set. Take a look at subsection 5.6.1 to see what those do.
- `MRTwuline`
- `MRTsfacc`
- If \XeTeX or \LuaTeX are used
 - `fontspec` and `\defaultfontfeatures{Ligatures=TeX}`
- `lmodern`
- `helvet`
- `babel` [`ngerman`] or if `british` is used with [`main=british, ngerman`]
- `setspace`
- `enumitem`
- `mathtools` with the `fleqn` option
- `mathastext` with `italic` and `defaultmathsizes`
- `isomath`
- if the `minimal` option is not used:
 - `siunitx` [`detect-all, per-mode=reciprocal-positive-first`]
If `babel's british` is used [`locale=UK`] will be used, if `ngerman` [`locale=DE`]. Additionally the `range-phrase` will be set to either `to` or `bis` with spaces around it.

5 The MRTtab package

The MRTtab provides means to typeset tables in a style similar to the ones in the scripts of the MRT. This includes:

- delimited by horizontal rules on both ends
- head rows are light grey and delimited by horizontal rules
- all horizontal rules have the same thickness
- no vertical rules (though not enforced)

The package provides an environment similar to `tabular` (section 5.1), an enhanced version of `\cline` (section 5.2), and an environment to typeset displayed tables with many options available (section 5.3).

5.1 The MRTtabular environment

The MRTtabular environment calls a patched `tabular` environment. The following differences exist:

- a hook is provided at the beginning and the end of each line
- above and below of it a `\hline` is placed
- it has an additional optional argument specifying the number of rows to be formatted as head rows.
- you can access the current row number
- automatic application of a stretch factor based on the `stretch` `tabular` key in subsection 5.6.1.

Any `tabular` environments inside of an MRTtabular are ordinary `tabulars` which neither have hooks nor row numbers. They might be affected by an outer `\rowcolor` or similar, though.

An ordinary description as done with other environments in this documentation:

MRTtabular	Usage: <code>\begin{MRTtabular}[\langle valign \rangle][\langle head rows \rangle]{\langle preamble \rangle} ...</code> <code>\end{MRTtabular}</code>
-------------------	--

The first optional argument as well as the mandatory argument match the ones of a regular `tabular` environment. `\langle head rows \rangle` specifies how many rows at the beginning of the environment should be formatted as head rows. If `\langle head rows \rangle` is not specified, no head row will be formatted. No further markup is required for this formatting to take place. You should end your rows only with `\\` to make the hook mechanism work (on which the head row markup relies).

\head	Usage: <code>\head{\langle num \rangle}</code>
--------------	--

Additionally to the optional argument of `MRTtabular` to set the first n rows as head rows, you can use `\head` to set the next `\langle num \rangle` rows as head rows. This does not only work at the beginning of the environment but anywhere you want. Alternatively you can use the macros described in section 5.4.

\MRTtabAddtoBoLHook	Usage: <code>\MRTtabAddtoBoLHook{\langle content \rangle}</code>
----------------------------	--

You can add `\langle content \rangle` to the Begin-of-Line hook with this macro. Bear in

mind that the $\langle content \rangle$ should be fully expandable and not produce any text, if you want to use stuff like `\multicolumn`, `\rowcolor`, or `\cline` at the beginning of the line – as this hook will be executed prior to that and `\noalign` and `\omit` won't work in that case. If you need something unexpandable you can enclose it in `\noalign`. The addition is made locally.

<code>\MRTtabClearBoLHook</code>	Usage: <code>\MRTtabClearBoLHook</code>
	Clears the Begin-of-Line hook locally.

<code>\MRTtabAddtoEoLHook</code>	Usage: <code>\MRTtabAddtoEoLHook{\langle content \rangle}</code>
	You can also add $\langle content \rangle$ to the End-of-Line hook. Here it should not matter whether the contents are expandable or not, as it is impossible that something follows in the same row which can't follow something unexpandable. The addition is made locally.

<code>\MRTtabClearEoLHook</code>	Usage: <code>\MRTtabClearEoLHook</code>
	Clears the End-of-Line hook locally.

<code>\MRTtabCurrentRow</code>	Usage: <code>\MRTtabCurrentRow</code>
	Returns the current row in an <code>MRTtabular</code> expandably.

5.1.1 Known Bugs

Currently only one bug is known: If after the last head row there is only one additional row the bottom `\hline` will only be drawn if you end that last row with `\\`. If you have more rows following the last head row, it won't matter whether you end the last row with `\\` or not.

5.2 The `\MRTcline` macro

<code>\MRTcline</code>	Usage: <code>\MRTcline[\langle color \rangle]{\langle * \rangle[\langle color \rangle]\langle <[\langle left skip \rangle] \rangle >[\langle right skip \rangle] \rangle \langle cols \rangle}</code>
------------------------	---

Sets something like a `\cline` in the specified $\langle cols \rangle$.

In the mandatory argument the only mandatory element is the affected $\langle cols \rangle$.

The mandatory argument can include a comma separated list in which you can repeat every optional argument you like as many times as you like. Additionally you can enclose the $\langle cols \rangle$ in curly braces and give another comma separated list there which then can only contain column specifications and none of the optional arguments using the optional arguments specified before that list. A valid column specification is a single column, or a column range separated by a `-`, so something like $\langle start-end \rangle$.

Both $\langle color \rangle$ arguments have the same effect, but the first applies to every specification in the list, while the second only affects the current list item. The $\langle color \rangle$ doesn't change the color of the line, but the color of the optional fill arguments. It defaults to either `tabulargray` if used inside the scope of head rows, or `white` else. If you give a $\langle * \rangle$ the current list item will be completely in

the specified `<color>`.

You can introduce a small skip on the left side if you specify a `<<` which defaults to `.5\tabcolsep`, with the optional `<left skip>` you can customize that length. A small skip to the right can be introduced with `>`, again of customizable width using `<right skip>`.

You should only use one `\MRTcline` per line and specify every column you want in that.

I hope you got that rather cryptic description (if you can supply a better description, message me as noted in section 1.1).

Here are a few examples of usage with comparison to a correct `\cline` usage. The source of each table is printed below it. The last example of `\MRTcline` is not possible with the standard `\cline` as far as I know.

a	b	c
d	e	f
g	h	i
j	k	l

```
\begin{MRTtabular}{lll}
a & b & c\\
\MRTcline{1-2}
d & e & f\\
g & h & i\\
j & k & l\\
\end{MRTtabular}
```

a	b	c
d	e	f
g	h	i
j	k	l

```
\begin{MRTtabular}{lll}
a & b & c\\
\cline{1-2}
\clineReveal
d & e & f\\
g & h & i\\
j & k & l\\
\end{MRTtabular}
```

a	b	c
d	e	f
g	h	i
j	k	l

```
\begin{MRTtabular}[] [2] {lll}
a & b & c\\
\MRTcline{1-2,*3}
d & e & f\\
g & h & i\\
j & k & l\\
\end{MRTtabular}
```

a	b	c
d	e	f
g	h	i
j	k	l

```
\begin{MRTtabular}[] [2] {lll}
a & b & c\\
\MRTcline{<>1-2,*3}
d & e & f\\
g & h & i\\
j & k & l\\
\end{MRTtabular}
```


a	b	c
d	e	f
g	h	i
j	k	l

```

\begin{MRTtabular}[] [2] {111}
  a & b & c\\
  \cline{1-2}
  \arrayrulecolor{tablegray}
  \cline{3-3}
  \arrayrulecolor{black}
  \clineReveal
  \rowcolor{tablegray}
  d & e & f\\
  g & h & i\\
  j & k & l\\
\end{MRTtabular}

```

5.3 The MRTtable environment

The MRTtable environment is a wrapper around an MRTtabular inside of a table environment. There might be a severe difference in the implementation of the long version, but as of now it is not implemented.

MRTtable Usage: `\begin{MRTtable}[\langle key=value \rangle] \dots \end{MRTtable}`
MRTtable sets its contents in an MRTtabular environment. It features several $\langle key \rangle$ s you are encouraged to use.
All available $\langle key \rangle$ s are listed in subsection 5.6.1.

5.4 Explicit head rows

It is possible to mark head rows explicitly. For this the following macros are provided:

\headS Usage: `\headS`
Start of the head rows. Sets a `\hline` above the current row except if the current row is the first row in a MRTtabular environment. Additionally the current row is coloured with `\rowcolor{tablegray}`.

\headR Usage: `\headR`
An additional head row should be started with this macro. It sets the current row's colour to `tablegray`.

\headE Usage: `\headE`
The end of the head rows. Should be used after the last row of the table's head but prior to the next row (immediately after `\\`).

<hr/> <code>\MRTtabDeclareHeadMacros</code> <hr/>	<p>Usage: <code>\MRTtabDeclareHeadMacros</code></p> <p>By default the above macros are only available inside of <code>MRTtabular</code> and in the body of <code>MRTtable</code>. You can make them locally available with <code>\MRTtabDeclareHeadMacros</code>.</p>
---	---

5.5 Other package macros

<hr/> <code>\MRTtabSetup</code> <hr/>	<p>Usage: <code>\MRTtabSetup{<key=value>}</code></p> <p>This is the interface to set the options listed in subsection 5.6.1 outside of <code>MRTtable</code>.</p>
<hr/> <code>\clineReveal</code> <hr/>	<p>Usage: <code>\clineReveal</code></p> <p>As you can see in section 5.2 the macro <code>\clineReveal</code> is used. This is done because a <code>\cline</code> doesn't take up any vertical space (by issuing <code>\noalign{\vskip-\arrayrulewidth}</code>) as opposed to a <code>\hline</code>. This is done so that multiple <code>\clines</code> can be used in the same row. As a result the spacing is inconsistent and a <code>\cline</code> is overlapped by a following <code>\rowcolor</code> or <code>\cellcolor</code>. <code>\clineReveal</code> does introduce a vertical skip which reveals the lines (issuing <code>\noalign{\vskip\arrayrulewidth}</code>). It is also used by <code>\MRTcline</code>.</p>

5.6 Options

The package only features one load time option, which is `longtable`. If it is specified the `longtable` package is loaded and some more options of `MRTtable` become available which are focused around the usage of `longtable` inside of `MRTtable`.

5.6.1 Setup Options

The following options are available for `\MRTtabSetup` and `MRTtable`.

<hr/> caption above above <hr/>	<p>–NoArgument–</p> <p>If specified the caption will be put above the <code>MRTtabular</code> in <code>MRTtable</code>. If <code>\KOMAOPTIONS</code> is available the KOMA option <code>captions=tableheading</code> is used.</p>
<hr/> caption below below <hr/>	<p>–NoArgument–</p> <p>If specified the caption will be put below the <code>MRTtabular</code> in <code>MRTtable</code>. If <code>\KOMAOPTIONS</code> is available the KOMA option <code>captions=tablesingature</code> is used.</p>
<hr/> bare <hr/>	<p>= <code><bool></code></p> <p>If set to true the potential caption and the tabular like environment in <code>MRTtable</code> are neither surrounded by a <code>minipage</code> nor a <code>figure</code>. Only a <code>\centering</code> is issued.</p>
<hr/> BoL <hr/>	<p>= <code><content></code></p> <p>Sets the <code>MRTtabular</code> Begin-of-Line hook using <code>\MRTtabAddtoBoLHook</code></p>

<u>EoL</u>	= $\langle content \rangle$ Sets the MRTtabular End-of-Line hook using <code>\MRTtabAddtoEoLHook</code>
<u>caption</u> <u>cap</u>	= $\langle caption \rangle$ Specifies the content of the caption in an MRTtable. If it is blank, no caption will be used.
<u>columns</u> <u>col</u>	= $\langle preamble \rangle$ Specifies the MRTtabular preamble (the column specifications). Defaults to first column l, others c.
<u>env</u>	= $\langle name \rangle$ Uses the tabular like environment $\langle name \rangle$ instead of MRTtabular. If an empty argument is provided, no inner environment will be used. This is useful if you want to use an environment that grabs its contents and has to be explicitly used, e.g. tabularx can only be used like this.
<u>env begin</u>	= $\langle begin \rangle$ Uses $\langle begin \rangle$ as the start of the tabular like environment. This way you can specify some options. Note that any outer braces are stripped. If you want to use an environment you have to include <code>\begin</code> in the argument. Note that if the argument you provide is not empty, the column specification as defined with columns is inserted in braces after $\langle begin \rangle$.
<u>env end</u>	= $\langle end \rangle$ Uses $\langle end \rangle$ as the end of the tabular like environment. This way you can specify some options. Note that any outer braces are stripped. If you want to use an environment you have to include <code>\end</code> in the argument.
<u>float</u>	= $\langle bool \rangle$ If set true (the default and initial value) the MRTtable floats.
<u>head rows</u> <u>head</u>	= $\langle num \rangle$ The number of rows which should be formatted as head rows as in MRTtabular. In each MRTtable it is initially 1 – this is different from a stand alone MRTtabular which by defaults to 0 rows.
<u>in text sep</u>	= $\langle skip \rangle$ This controls the vertical space around a non-floating MRTtable. It is initially set to <code>\intextsep</code> . If it is equal to Opt the <code>\vskip</code> is not issued.
<u>label</u>	= $\langle label \rangle$ If caption is used the MRTtable will get the specified $\langle label \rangle$.

<u>no float</u>	= $\langle bool \rangle$ The opposite of <code>float</code> . If set true the <code>MRTtable</code> will not float which is the default (but not initial) value.
<u>no inner env</u>	–NoArgument– Same result as <code>env begin={}</code> , <code>env end={}</code> , so no tabular like environment is used at all.
<u>pos</u>	= $\langle pos \rangle$ The placement of a floating <code>MRTtable</code> . Initially <code>tbp</code> .
<u>post tab</u> <u>post</u>	= $\langle content \rangle$ A hook which is executed right after the <code>\end</code> of the inner <code>MRTtabular</code> .
<u>pre tab</u> <u>pre</u>	= $\langle content \rangle$ A hook which is executed right before the <code>\begin</code> of the inner <code>MRTtabular</code> .
<u>short caption</u> <u>short cap</u> <u>scap</u>	–NoArgument– If caption and this option are used the list of tables will get this short caption instead of the caption.
<u>stretch tabular</u> <u>stretch tab</u> <u>stretch</u>	= $\langle float \rangle$ Sets the stretch in <code>MRTtabular</code> to the specified $\langle float \rangle$ using <code>\setstretch</code> .
<u>stretch caption</u> <u>stretch cap</u> <u>cstretch</u>	= $\langle float \rangle$ Sets the stretch in the caption using <code>\setkomafont</code> and <code>\setstretch</code> . Doesn't work if KOMA script is not used but issues a warning in that case.
<u>striped</u>	= $\langle bool \rangle$ If set to true the inner <code>MRTtabular</code> will be striped with <code>stripe color 1</code> and <code>stripe color 2</code> , beginning in line <code>stripe start</code> . It uses <code>\rowcolors</code> internally.
<u>stripe color 1</u> <u>stripe 1</u> <u>scolor 1</u> <u>scolor1</u>	= $\langle color \rangle$ Defines the $\langle color \rangle$ of the first color argument of <code>\rowcolors</code> if <code>striped</code> is true. Initially set to <code>tablegray!50</code> .
<u>stripe color 2</u> <u>stripe 2</u> <u>scolor 2</u> <u>scolor2</u>	= $\langle color \rangle$ Defines the $\langle color \rangle$ of the second color argument of <code>\rowcolors</code> if <code>striped</code> is true. Initially set to <code>white</code> .
<u>stripe invert</u> <u>sinvert</u>	–NoArgument– Exchanges the current values of <code>stripe color 1</code> and <code>stripe color 2</code> .

stripe start	= $\langle row \rangle$	
sstart		Defines the starting row of a potentially striped MRTtabular. Initially set to 2.

5.6.1.1 longtable related options

The following options are only available if the `longtable` was used during package load time.

long	= $\langle bool \rangle$	
		If set true the MRTtable uses <code>longtable</code> internally. It doesn't float and gets page breakable. You should specify the columns of MRTtable manually as the automatic detection might fail terribly in conjunction with <code>longtable</code> .

continue caption	= $\langle caption \rangle$	
cont cap		If specified following pages use this $\langle caption \rangle$ instead of the <code>short caption</code> or the normal caption.

continue with caption	= $\langle bool \rangle$	
cont with cap		If set true, the following pages use the <code>caption</code> and not the <code>short caption</code> or <code>continue caption</code> . Defaults to true and initially is set to false.
cont w cap		

5.7 Dependencies

The package requires the following packages and their dependencies:

- | | | |
|---|---|--|
| <ul style="list-style-type: none"> • <code>expl3</code> • <code>xparse</code> | <ul style="list-style-type: none"> • <code>array</code> • <code>setspace</code> | <ul style="list-style-type: none"> • <code>xcolor</code> with option <code>table</code> • <code>potentially longtable</code> |
|---|---|--|

6 The MRTif package

The MRTif package provides a number of expandable tests. In the following macros *TF* is used to specify that the macros exist with the endings T, F, and TF. The T ending stands for a *<true>* branch, F for the *<false>* branch.

If a macro name contains a G prior to *TF*, it strips any outermost groups prior to the test using *\MRTifGroupTF*. An N denotes that the first token in the argument is expanded once prior to any test. If a macro which takes two arguments ends with NN prior to the *TF* in both arguments the first token is expanded once, Nn and nN mean that only for the first and second argument, respectively, an expansion is made.

6.1 Macros

<i>\MRTifEmptyTF</i>	Usage: <i>\MRTifEmptyTF{<arg>}{<true>}{<false>}</i>
<i>\MRTifEmptyGTF</i>	
<i>\MRTifEmptyNTF</i>	Tests if <i><arg></i> is completely empty.
<i>\MRTifEmptyGNTF</i>	

<i>\MRTifGroupTF</i>	Usage: <i>\MRTifGroupTF{<arg>}{<true>}{<false>}</i>
<i>\MRTifGroupNTF</i>	Tests if <i><arg></i> is a single group no matter what the contents of that group are. It ignores spaces around the group.

<i>\MRTifGroupNoSpacesTF</i>	Usage: <i>\MRTifGroupNoSpacesTF{<arg>}{<true>}{<false>}</i>
<i>\MRTifGroupNoSpacesNTF</i>	Tests if <i><arg></i> is a single group no matter what the contents of that group are. It doesn't ignore spaces around the group.

<i>\MRTifStringsMatchTF</i>	Usage: <i>\MRTifStringsMatchTF{<string1>}{<string2>}{<true>}{<false>}</i>
<i>\MRTifStringsMatchNNTF</i>	Tests if <i><string1></i> and <i><string2></i> match.
<i>\MRTifStringsMatchNnTF</i>	
<i>\MRTifStringsMatchnNTF</i>	
<i>\MRTifStringsMatchGTF</i>	
<i>\MRTifStringsMatchGNTF</i>	
<i>\MRTifStringsMatchGNnTF</i>	
<i>\MRTifStringsMatchGnNTF</i>	

<i>\MRTifOneTokenTF</i>	Usage: <i>\MRTifOneTokenTF{<arg>}{<true>}{<false>}</i>
<i>\MRTifOneTokenGTF</i>	Tests if <i><arg></i> is only a single token.
<i>\MRTifOneTokenNTF</i>	
<i>\MRTifOneTokenGNTF</i>	

<i>\MRTifLetterTF</i>	Usage: <i>\MRTifLetterTF{<arg>}{<true>}{<false>}</i>
<i>\MRTifLetterGTF</i>	Tests if <i><arg></i> is a letter, meaning of category code 11.
<i>\MRTifLetterNTF</i>	
<i>\MRTifLetterGNTF</i>	

<code>\MRTifTokensMatchTF</code>	Usage: <code>\MRTifTokensMatchTF{<arg1>}{<arg2>}{<true>}{<false>}</code>
<code>\MRTifTokensMatchNNTF</code>	Tests if <code><arg1></code> and <code><arg2></code> are single tokens and if so compares them whether both tokens match. The variants without G test if one of the arguments is contained in a group. If that's the case the <code><false></code> branch is executed.
<code>\MRTifTokensMatchNnTF</code>	
<code>\MRTifTokensMatchnnTF</code>	
<code>\MRTifTokensMatchGTF</code>	
<code>\MRTifTokensMatchGNNTF</code>	
<code>\MRTifTokensMatchGNnTF</code>	
<code>\MRTifTokensMatchGnNNTF</code>	

6.2 Dependencies

MRTif loads the `pdftexcmds` package to make the pdfTeX primitive `\pdfstrcmp` available as `\pdf@strcmp` for LuaTeX.

7 The MRTwuline package

The package provides a MS Word like looking line breakable underlining. It does so by using `ulem` or `stackengine`.

7.1 Options

<code>tUline</code>	–NoArgument–
<code>tikzunderline</code>	If this option is passed <code>TikZ</code> will be added as a required package and an alternative underlining macro defined called <code>\tUline</code> , see its description in section 7.2.

7.2 Macros

<code>\WUline</code>	<p>Usage: <code>\WUline[⟨height⟩]{⟨text⟩}</code></p> <p>This sets <code>⟨text⟩</code> and underlines it in a way that looks like MS Word underlining – at least in the headings. It is usable both in math mode and in text mode. Though in math mode you should use <code>\underline</code>.</p> <p>In text mode the <code>ulem</code> package is used for the underline. In math mode <code>stackengine</code> is employed. In both cases you can use <code>⟨height⟩</code> to change the default height of the underlining. In text mode and math mode the needed <code>⟨height⟩</code> to achieve the same height of the line differs quite a lot. By default in math mode <code>0.21ex</code> is used, in text mode <code>-0.42ex</code>.</p>
<code>\tUline</code>	<p>Usage: <code>\tUline[⟨height⟩][⟨overhang⟩][⟨thickness⟩]{⟨text⟩}</code></p> <p>This macro can be used to underline bigger portions of text. You should never need it, I guess. Just use <code>\WUline</code> instead. If you need it, you'll have to use the package option <code>tUline</code>.</p> <p>If you think you can use this one instead: It underlines <code>⟨text⟩</code> at the given <code>⟨height⟩</code> (default <code>-0.35ex</code>) with the given <code>⟨thickness⟩</code> (default <code>0.185ex</code>). You can specify <code>⟨overhang⟩</code> (default <code>0pt</code>) which is the width the line should be wider than a text line on each side. If you let any optional argument empty, the default is used. It is assumed that the lines are equally separated with <code>\baselineskip</code> – so if your material does stretch the baseline skip, you can't use <code>\tUline</code>. It needs at least two runs to be displayed correctly.</p>

7.3 Dependencies

- `expl3`
- `xparse`
- `stackengine`
- `scalerel`
- `MRTif`
- `ulem` with the `normalem` option
- if the `tUline` option is used:
 - `TikZ`
 - `tikzpagenodes`
 - The `TikZ` library `calc`

8 The MRTsfacc package

This package is provided to remedy an issue related with sans serif maths, to be more precise to fix the placement of `\mathaccentV`, which is internally used by macros such as `\hat` and `\dot`. It is therefore loaded by both, `MRTthesis` and `MRTbeam`. The `beamer` class provides a fix for the same issue which is unfortunately only working for `beamer`'s default font by fixing the font metrics.

`MRTsfacc` has a different approach by patching `\mathaccentV` to move the accent horizontally depending on the height of the accented character. Furthermore it is tested whether the character is an alphabetic one by checking the category code. If it is not an alphabetic character the shift isn't applied.

You can specify the shift based on the multiple of the height as the package option. This is the only option the package takes and is not checked to be a valid float. The default is 0.25. Alternatively you can use `\MRTsfaccSet` to set the coefficient.

One can use `*` to enforce the shift and `!` to enforce the omitting of that shift. Consider the following example: `\hat!{m}` produces \hat{m} , `\hat*{m}` produces \hat{m} , and `\hat{m}` produces \hat{m} , which is the same as `\hat*{m}` since `m` has by default the category code of an alphabetic character.

The tokens `*` and `!` must not be enclosed by braces if you want to specify the behaviour of `\mathaccentV`, so `\hat{*}\hat{!}` results in $\hat{*}\hat{!}$.

The resulting `\mathaccentV` macro is not expandable and therefore created `\protected`. You can take a look at the results in Table 8-1.

original	shifted	original	shifted
\hat{a}	\hat{A}	\hat{n}	\hat{N}
\hat{b}	\hat{B}	\hat{o}	\hat{O}
\hat{c}	\hat{C}	\hat{p}	\hat{P}
\hat{d}	\hat{D}	\hat{q}	\hat{Q}
\hat{e}	\hat{E}	\hat{r}	\hat{R}
\hat{f}	\hat{F}	\hat{s}	\hat{S}
\hat{g}	\hat{G}	\hat{t}	\hat{T}
\hat{h}	\hat{H}	\hat{u}	\hat{U}
\hat{i}	\hat{I}	\hat{v}	\hat{V}
\hat{j}	\hat{J}	\hat{w}	\hat{W}
\hat{k}	\hat{K}	\hat{x}	\hat{X}
\hat{l}	\hat{L}	\hat{y}	\hat{Y}
\hat{m}	\hat{M}	\hat{z}	\hat{Z}

Table 8-1: Comparison of shifted accents against original placement

8.1 Dependencies

MRTsfacc loads the MRTif package and uses its test \MRTifLetterGTF.