

UWMadThesis Class Manual

Troy Christopher Haskin

Table of Contents

I	User Guide	1
1	Thesis and PDF Information	2
1.1	Required	2
1.2	Optional	3
1.3	Accessors	4
2	Special Pages	6
2.1	Title Page	6
2.2	License Page	6
2.2.1	Copyright	7
2.2.2	Creative Commons	8
3	Layout And Style	12
3.1	Captions	12
3.2	Links	12
4	Sectioning	13
4.1	Front Matter	13
4.2	Appendix	13
4.3	Table of Contents Tweaks	14
5	List Environments	15
5.1	Nomenclature	15
5.1.1	Command Descriptions	15
5.1.2	Examples	16
5.1.3	Customization	18
5.2	Acronym	19

	ii
5.2.1 Description	19
5.2.2 Example	19
6 Math	22
6.1 Derivative Commands	22
6.2 Operators	25
6.3 Miscellaneous Commands	28
7 Programming	31
II Implementation	32
1 Front Matter	33
1.1 expl3 Package and Identification	33
1.2 Identification and Defaults	33
1.3 Options	34
1.4 Package Loads	36
2 Programming	38
2.1 Utility Commands	38
2.2 Collections	40
2.2.1 Stacks	40
2.2.2 Queues	42
2.2.3 Deques	44
2.2.4 Hashes	46
2.3 User-Level Abstractions	48
2.3.1 Utility Commands	48
2.3.2 Command Creators	50
2.3.3 Types	51

3	Layout And Styles	54
4	Sectioning	56
4.1	Appendix	56
4.2	Front Matter	57
4.3	TOC Tweaks	59
4.4	Section-Level Commands	61
5	Math	63
6	ListOf	68
6.1	Nomenclature	75
7	Thesis and PDF Information	85
7.1	Metadata clist and Aux Write	85
7.2	Thesis Information	86
8	Special Pages	93
8.1	MakeTitlePage	93
8.2	LicensePage	94
8.2.1	Copyright	95
8.2.2	Creative Commons	95
8.2.3	LicensePage Proper	98
	Change History	101
	Index	102

Part I

User Guide

The `UWMadThesis` class is aimed at providing a $\text{\LaTeX} 2_{\epsilon}$ class that conforms to the style and format guidelines of the Graduate School of the University of Wisconsin--Madison. A copy of the current style guidelines and other associated

In addition to that primary goal, the class also loads a number of useful packages and defines or expands on a number of commands and utilities for creating a high-quality document.

Feature Set 1

Thesis and PDF Information

In order for the [Title Page](#) to function properly, a certain amount of information about the thesis must be given. The `UWMadThesis` class has a set of commands to provide both the thesis information and PDF metadata to \LaTeX .

It is highly encouraged to use all of these commands in the preamble such that any PDF metadata can be directly set before the document begins. If the commands are used within the `document` environment, it will require another \LaTeX compilation to include the metadata since `UWMadThesis` class will automatically write the information to an external file.

1.1 Required

These commands are required for the document to be typeset properly. It is encouraged to use these commands in the preamble of the document.

<code>\Title</code>	<code>\Title</code> $\{\langle title \rangle\}$
<code>\Author</code>	<code>\Author</code> $\{\langle author name \rangle\}$
<code>\Program</code>	<code>\Program</code> $\{\langle program \rangle\}$
<code>\Degree</code>	<code>\Degree</code> $\{\langle degree \rangle\}$

Each of these commands must be used once; if not, their respective variables be empty while being typeset. They can, of course, be used more than once, but the additional usages would only redefine the value of the associated variable.

<code>\DefenseDate</code>	<code>\DefenseDate</code> $\{\langle defense date \rangle\}$
<code>\DefenceDate</code>	<code>\DefenceDate</code> $\{\langle defense date \rangle\}$

Only one of these commands is needed since they all point to the same variable $\{\langle defense date \rangle\}$. The aliases were created for personal preference only.

Since $\{\langle defense date \rangle\}$ has no parsing performed on it, it may be entered any which way and will be typeset as-entered.

<code>\Institution</code>	<code>\Institution {\langle institution name \rangle}</code>
<code>\University</code>	<code>\University {\langle institution name \rangle}</code>

Only one of these commands is needed since they both point to the same variable `{\langle institution name \rangle}`. The aliases were created for personal preference only.

<code>\CommitteeMember</code>	<code>\CommitteeMember {\langle member name \rangle} {\langle member position \rangle}</code>
-------------------------------	---

`\CommitteeMember` can be used as many times as required. However, if the list of members becomes too large, formatting of the [title page](#) will suffer. This may be fixed in the future but would require a much more sophisticated template for the title page (possibly using `expl3` `coffins`).

1.2 Optional

These commands are not required for the document to be typeset properly. However, they do provide metadata for the PDF (e.g., keywords and document subject) that is convenient for searching and categorization. It is encouraged to use these commands in the preamble of the document.

<code>\DocumentType</code>	<code>\DocumentType {\langle document type \rangle}</code>
----------------------------	--

By default, the `\MakeTitlePage` command prints the phrase “A `{\langle document type \rangle}` submitted in partial fulfillment of the requirements for the degree of” on the title page”. The default `{\langle document type \rangle}` is “report”. This command sets the value to any valid text. However, facilitate good semantic mark-up, some prepared commands to set the document type are given below.

<code>\Dissertation</code>	<code>\Dissertation</code>
<code>\DoctoralThesis</code>	<code>\DoctoralThesis</code>
<code>\MastersThesis</code>	<code>\MastersThesis</code>
<code>\Thesis</code>	<code>\Thesis</code>
<code>\Prelim</code>	<code>\Prelim</code>

These commands set the value of `{\langle document type \rangle}` to a value similar to their command name:

- `\Dissertation` sets `{\langle document type \rangle}` to “dissertation”
- `\DoctoralThesis` sets `{\langle document type \rangle}` to “doctoral thesis”
- `\MastersThesis` sets `{\langle document type \rangle}` to “master's thesis”
- `\Thesis` sets `{\langle document type \rangle}` to “thesis”
- `\Prelim` sets `{\langle document type \rangle}` to “preliminary report”

<code>\Advisor</code>	<code>\Advisor {<advisor name>}{<advisor position>}</code>
<code>\Adviser</code>	<code>\Adviser {<advisor name>}{<advisor position>}</code>

Using either of these commands automatically adds the advisor/adviser to the top of the committee list created by `\CommitteeMember`. Also, on the title page's committee list, the advisor/adviser is marked as such by ```(Advisor)''` or ```(Adviser)''`. This is a rare exception where the choice of alias has a side-effect.

<code>\Subject</code>	<code>\Subject {<document subject>}</code>
<code>\Keywords</code>	<code>\Keywords {<list of keywords>}</code>

These commands set the subject and keyword portions of the PDF metadata. The `{<document subject>}` is typically a one-ish line description of the document. The `{<list of keywords>}` can be a long, punctuation-delimited list (e.g., comma or semicolon) of keywords.

<code>\Producer</code>	<code>\Producer {<pdf producer>}</code>
<code>\Creator</code>	<code>\Creator {<pdf creator>}</code>

These commands set the PDF Producer and PDF Creator fields of the metadata. These fields are a little confusing in their intended usage. The best explanation found is

Creator The application used to create the original document which became the PDF.

Producer The application used to convert the original document into the PDF.

These are very thin distinctions and complicated by the typical workflow of a \LaTeX document: installing a \TeX distribution, editing a text file in $\text{\TeX}/\text{\LaTeX}$ editor, and running the document through a \TeX engine with the \LaTeX format. In order to give credit at all levels (while maintaining proper separation of the processes involved), it is recommended to state the editor and \TeX format used as the creator and state the engine and distribution used as the producer. For example, this document would declare the following:

```
\Creator{TeXnicCenter 2.02, LaTeX2e+}
\Producer{pdfTeX 1.40.14, MiKTeX 2.9}
```

But as stated before, this is all optional.

1.3 Accessors

\TheTitle
\TheAuthor
\TheProgram
\TheDegree
\TheDefenseDate
\TheDefenceDate
\TheInstitution
\TheDocumentType
\TheAdvisor
\TheSubject
\TheKeywords
\TheProducer
\TheCreator

If, for any reason, the thesis information or metadata registered with the document is required, these accessor commands exist to retrieve the stored value.

Feature Set 2

Special Pages

2.1 Title Page

This is a replace for the default `\maketitle`. Per the example provided by the University of Wisconsin--Madison Graduate School's sample, the sample page flows (in order): thesis title, author by-line, partial fulfillment clause, degree, program, university identification, oral defense date, and oral committee list. The styles can be re-worked by redefining the commands as presented in the [MakeTitlePage](#) implementation. The formatting of the commands is standard L^AT_EX 2_ε to facilitate customization.

NOTE: The `\MakeTitlePage` command needs the required thesis information from the commands described in the [Required subsection](#).

2.2 License Page

There are two main licenses `UWMadThesis` class supports: Copyright and Creative Commons. If an author wishes to use these supported licenses to create a license page, all of the commands listed must be placed within a `LicensePage` environment, or the commands will not work (by design).

To declare a simple Copyright input

```
\begin{LicensePage}
  \Copyright
\end{LicensePage}
```

To declare a simple Creative Commons input

```
\begin{LicensePage}
  \CreativeCommons
\end{LicensePage}
```

There are more features for the Creative Commons license and are discussed below.

The above examples will automatically create a page using default values for license owner (the [thesis author](#)), year (the current year), and license specifics (outlined below). If either is incorrect for the current usage, use the following commands:

<code>\LicenseOwner</code>	<code>\LicenseOwner {<i><owner name></i>}</code>
<code>\LicenseYear</code>	<code>\LicenseYear {<i><year></i>}</code>

These commands override the default values with the supplied, mandatory argument.

2.2.1 Copyright

The Copyright Act of 1976 ([Title 17 of the United States Code](#), section 106) lists the following six exclusive rights the owner of copyright and any other sanctioned parties have:

1. to reproduce the copyrighted work in copies or phonorecords
2. to prepare derivative works based upon the copyrighted work
3. to distribute copies or phonorecords of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending
4. in the case of literary, musical, dramatic, and choreographic works, pantomimes, and motion pictures and other audiovisual works, to perform the copyrighted work publicly
5. in the case of literary, musical, dramatic, and choreographic works, pantomimes, and pictorial, graphic, or sculptural works, including the individual images of a motion picture or other audiovisual work, to display the copyrighted work publicly
6. in the case of sound recordings, to perform the copyrighted work publicly by means of a digital audio transmission

There are a number of exceptions and limitations to these rights as outlined by subsequent sections (Title 17 of the United States Code, sections 107 -- 122), but these will not be discussed. Under section 302 of the Copyright Act, the exclusive rights granted to a singular author of a work persist for 70 years following her death.

Section 401 of the Copyright Act requires a Form of Notice of copyright. It consists of the elements: the copyright symbol © (or the word ``Copyright''), the year of first publication (with more requirements for derivative works), and the name of the owner of the copyright (or some other designation). All works containing this notice of copyright fall under the protection of the Copyright Law of the United States.

Section 408 of the Copyright Act states: for any work produced after 1978, ``the owner of copyright or of any exclusive right in the work may obtain registration of the copyright claim by delivering to the

Copyright Office the deposit specified by this section, together with the application and fee". In others words, a copy of the work can be submitted to the Copyright Office and subsequently placed in the Library of Congress for official recognition of copyright. However, registration is not compulsory since "[s]uch registration is not a condition of copyright protection".

`\Copyright` `\Copyright`

Using this command within a `LicensePage` environment will print a Copyright Notice at the bottom of a page and place a link in the table of contents.

An example of usage (along with a redefined owner and year) would be

```
\begin{LicensePage}
  \LicenseOwner{Theodore Huxton}
  \LicenseYear{3001}
  \Copyright
\end{LicensePage}
```

This input would generate the following text at the bottom of a new page (with a link in the table of contents:

Copyright © 3001 by Theodore Huxton

2.2.2 Creative Commons

Creative Commons (CC) is a collective set of licenses that is most aptly described as "some rights reserved". That is, while Copyright requires explicit permission of the author for a lot of uses, Creative Commons immediately waives those rights. Why is this a good thing? To quote from [CreativeCommons.org](https://creativecommons.org):

Creative Commons is a nonprofit organization that enables the sharing and use of creativity and knowledge through free legal tools. ...

If you want to give people the right to share, use, and even build upon a work you've created, you should consider publishing it under a Creative Commons license. CC gives you flexibility (for example, you can choose to allow only non-commercial uses) and protects the people who use your work, so they don't have to worry about copyright infringement, as long as they abide by the conditions you have specified.

Therefore, the goal of CC is to begin from the "most free" license of public domain (termed CC0) and then add on conditions for legal use of the material. CC license are copyright licenses in that (aside from

CC0) that author retains certain ownership rights, but a subset of the rights are relaxed or waived to encourage free sharing and extension of the work. To this end, Creative Commons defines the following four conditions:

Attribution Appropriate credit must be given to the original author, a link to the license provided, and indication of any changes that were made. This may be done in any reasonable manner, but not in any way that suggests the licensor endorses the new author or her use.

ShareAlike If the work is remixed, transformed, or built upon the licensed material, the author of the new work MUST DISTRIBUTE the contributions under the same license as the original.

NoDerivs If the work is remixed, transformed, or built upon the licensed material, the author of the new work MAY NOT distribute the modified material.

NonCommercial The licensed work MAY NOT be used the material for commercial purposes.

These conditions are then combined into six, non-contradictory licenses. The licenses are ``layered'' into Legal Code (the official text determining the delineating usage), the License deed (non-legal text aimed to be non-lawyer readable), and machine readable code (the license put into an HTML-like style for search engines). The CC licenses and associated links) for the latest version are

CC BY

Attribution only ([License Deed](#) | [Legal Code](#)).

CC BY-SA

Attribution and ShareAlike ([License Deed](#) | [Legal Code](#)).

CC BY-ND

Attribution and NoDerivs ([License Deed](#) | [Legal Code](#)).

CC BY-NC

Attribution and NonCommerical ([License Deed](#) | [Legal Code](#)).

CC BY-NC-SA

Attribution, NonCommercial, and ShareAlike ([License Deed](#) | [Legal Code](#)).

CC BY-NC-ND

Attribution, NonCommercial, and NoDerivs ([License Deed](#) | [Legal Code](#)).

Prior to version 4.0 (the current one), there were a number of ``ports'' of the licenses to particular locales to deal with the specifics of individual countries. However, with the release of version 4.0 of the CC licenses, usage of the international version is highly encouraged as ports will be made ``[only where a compelling need is demonstrated](#)'' . As such, version 4.0 International is the default license base for the UWMadThesis class class. Of course, this choice can be changed.

<code>\CreativeCommons</code>	<code>\CreativeCommons</code>
-------------------------------	-------------------------------

Using this command within a `LicensePage` environment will declare you have chosen a Creative Commons license. By default, the license will be ```Creative Commons Attribution 4.0 International''`.

<code>\Attribution</code>	<code>\Attribution</code>
<code>\ShareAlike</code>	<code>\ShareAlike</code>
<code>\NonCommercial</code>	<code>\NonCommercial</code>
<code>\NoDerivs</code>	<code>\NoDerivs</code>

Using any of these commands (in any order) within a `LicensePage` environment will declare you have chosen to add the associated condition to the license of the work. However, since all six licensees require Attribution, it is always on by default but should be included for clarity.

An example of usage would be

```
\begin{LicensePage}
  \CreativeCommons
  \Attribution
  \NonCommercial
  \ShareAlike
\end{LicensePage}
```

This input would generate the following text at the bottom of a new page (with a link in the table of contents):

This work is released under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International](#) license.

Troy Christopher Haskin, 2014

Notice that since neither the `\LicenseOwner` nor `\LicenseYear` commands were used, the author of this document and then-current year were used as defaults.

<code>\CCVersion</code>	<code>\CCVersion{<CC version>}</code>
<code>\CCPorting</code>	<code>\CCPorting{<CC porting>}</code>
<code>\CCURL</code>	<code>\CCURL {<CC link>}</code>
<code>\CCURLText</code>	<code>\CCURLText{<CC link text>}</code>

These commands exist to override the default 4.0 International Creative Commons license. The link provided SHOULD NOT contain `http://` nor end with a `/`. Use these commands only if there is a compelling reason not to use the latest version of the license.

An example of usage would be

```
\begin{LicensePage}  
  \CreativeCommons  
  \CCVersion{3.0}  
  \CCPorting{United States}  
  \CCURL{creativecommons.org/licenses/by/3.0/us}  
  \CCURLText{Creative Commons Attribution 3.0 United States}  
\end{LicensePage}
```

This input would generate the following text at the bottom of a new page (with a link in the table of contents):

This work is released under a [Creative Commons Attribution 3.0 United States](https://creativecommons.org/licenses/by/3.0/us/) license.
Troy Christopher Haskin, 2014

Feature Set 3

Layout And Style

The `UWMadThesis` class has several default styling differences from the standard $\text{\LaTeX} 2_{\epsilon}$ class it is based on. Some of these changes exist to abide by the UW--Madison dissertation guidelines and others are based on the author's preferences. They are, however, readily changeable using the facilities of the packages used to make the changes. The defaults and methods for changing the style are list in this section.

3.1 Captions

The `UWMadThesis` class uses the `caption` and `subcaption` packages to style float captions and subcaptions. It is possible to adjust the defaults showcase below by using the packages' utilities outlines in their respective manuals.

FIGURE 1: *Here is an example of a figure caption. The default style for the `UWMadThesis` class is a slanted font (abbrev. “sl”) and small capitals (abbrev. “sc”) for the float label. Notice that long captions, like this, are indented such that the caption text is visibly separated from the float label.*

TABLE 1: *Here is a shorter example of a table caption. The default styling is identical to the figure caption.*

3.2 Links

The `UWMadThesis` class loads the `hyperref` and `bookmark` packages to create hyperlinks and a clickable documents. The default color for document links is [blue](#), for urls is [violet](#), and for citations is [UWMadGreen](#) (a darker version of [green](#)). These defaults can be change using the facilities of the `hyperref` package as described in its manual. New colors can be created using the facilities of the `xcolor` package as described in its manual.

References may be handled by the `hyperref` package using `\autocite` or by the `cleveref` package using `\cref/\Cref` (the latter producing a capital letter for the reference type).

Feature Set 4

Sectioning

Sectioning concerns the overall structure of your document into chunks called sections. The default sections in $\text{\LaTeX 2}_{\epsilon}$ are `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, and `subparagraph`. The `UWMadThesis` class defines some new section commands and makes some other adjustments to the default commands.

4.1 Front Matter

Front Matter (or preliminary pages) is the whole of content that precedes the main document (i.e., the first unstarred chapter). UW--Madison requires that these pages are numbered in lower roman numerals and have that page number in the upper right-hand corner. This requirement is automatically handled by the class. The Front Matter commands are all semantically named and set as starred (unnumbered) chapters.

<code>\dedications</code>	<code>\dedications</code>	<code>{\title}</code>
<code>\acknowledgments</code>	<code>\acknowledgments</code>	<code>{\title}</code>
<code>\abstract</code>	<code>\abstract</code>	<code>{\title}</code>
<code>\umiabstract</code>	<code>\umiabstract</code>	<code>{\title}</code>
<code>\preface</code>	<code>\preface</code>	<code>{\title}</code>

The title IS OPTIONAL. If the title is omitted, the default is a capitalized version of the command's name. For example, `\dedications` will have the title “Dedications”.

4.2 Appendix

The standard method of including appendices in \LaTeX is calling for some initialization to be done by using the `\appendix` command and then using the `\chapter` commands. The `UWMadThesis` class takes a different approach to encourage good semantic mark-up in \LaTeX documents and, therefore, redefines `\appendix`.

<code>\appendix</code>	<code>\appendix [⟨short title⟩]{⟨title⟩}</code>
	<code>\appendix* [⟨short title⟩]{⟨title⟩}</code>

The appendix commands now acts like a `\chapter` commands and is typeset in the Table of Contents as one.

NOTE: The usage `\appendix` should be after all the chapter material is set since some of the `\chapter` internals are changed. Once the `\appendix` command is used, there is no mechanism to switch the internals back.

4.3 Table of Contents Tweaks

Invoking the Table of Contents, List of Tables, and List of Figures commands now puts the start of those sections into the Table of Contents as chapters.

<code>\TableOfContentsName</code>	<code>\TableOfContentsName{⟨toc title⟩}</code>
<code>\ListOfTablesName</code>	<code>\ListOfTablesName {⟨lot title⟩}</code>
<code>\ListOfFiguresName</code>	<code>\ListOfFiguresName {⟨lof title⟩}</code>

These commands redefine the title used in the associated sections. The defaults for the TOC, LOT, and LOF are, respectively, “Table of Contents”, “List of Tables”, and “List of Figures”.

<code>\TableOfContents</code>	<code>\TableOfContents</code>
<code>\ListOfTables</code>	<code>\ListOfTables</code>
<code>\ListOfFigures</code>	<code>\ListOfFigures</code>

Camel-cased versions of the standard \LaTeX commands. These exist due to the preferences of the `UWMadThesis` class author.

Feature Set 5

List Environments

The `UWMadThesis` class has a special set of functions from creating list environments (called `ListOf` in the implementation). The functions use queues and associative arrays to store and use data before it is typeset. These data structures allow for operations to be carried out without writing external files or repeating compilation; of course, there is added memory usage which could lead to problems on older systems.

The primary motivation for such a system was the creation of a nomenclature environment and, subsequently, an acronym environment/system. These two similar features are discussed here.

5.1 Nomenclature

The `Nomenclature` environment is, by default, a list of `(symbol, description)` entries. There is a user option for changing the system to a list of `(symbol, units, description)` entries if a separate unit column is desired. For every set of entries, the nomenclature system measures the width of the `symbol` and (if present) `units` to determine the maximum width of the `description` such that no text overflows into the margins of the page.

When first adding entries to a nomenclature, the symbols are part of the so-called Main group. The Main group has a title and a section level associated with it. By default, the Main group title is ```Nomenclature''` and the section is ```chapter''`. The entries can be put into two lower sectioned groups using the `\Group` and `\Subgroup` commands described below. The grouping commands allows a set of symbols to be classified as ```Greek Symbols''` while another is ```Subscripts''`. The default titles for these lower groups are empty by default and the default section is ```section''` and ```subsection''`.

All of these defaults can be changed by the `\NomenclatureSetup` command described below.

5.1.1 Command Descriptions

A sketch of the `Nomenclature` implementation would be:

```
\begin{Nomenclature}[\langle section \rangle]{\langle title \rangle}
```

```

\Entry{<symbol>}{<description>}

\Group{<group title>}

\Entry{<symbol>}{<description>}

\Subgroup{<subgroup title>}

\Entry{<symbol>}{<description>}

\end{Nomenclature}

```

The square brace-delimited [*<section>*] is OPTIONAL and overrides the default Main group section. The curly brace-delimited {<title>} is OPTIONAL and overrides the default Main group title.

<code>\Entry</code>	<code>\Entry{<symbol>}{<description>}</code>
	<code>\Entry{<symbol>}{<units>}{<description>}</code>

Within the environment, entries are added to the nomenclature using the `\Entry` command above. All arguments are required. The second version above is if a units column is requested (see [Customization](#)).

<code>\Group</code>	<code>\Group{<group title>}</code>
<code>\Subgroup</code>	<code>\Subgroup{<subgroup title>}</code>

Creates a group or subgroup with the indicated title and using the default section. The default section can be changed by the user (see [Customization](#)).

5.1.2 Examples

As an example, the following input

```

\begin{Nomenclature}[subsubsection]{Symbol Table}
\Entry{<\rho>}{Density}
\Entry{LongNotRealSymbol}{
  In publishing and graphic design, lorem ipsum is a placeholder
  text commonly used to demonstrate the graphic elements of a
  document or visual presentation. By replacing the distraction
  of meaningful content with filler text of scrambled Latin it
  allows viewers to focus on graphical elements such as font,
  typography, and layout.}
\Entry{<\mu>}{Viscosity}
\end{Nomenclature}

```

would be typeset as:

Symbol Table

ρ	Density
LongNotRealSymbol	In publishing and graphic design, lorem ipsum is a placeholder text commonly used to demonstrate the graphic elements of a document or visual presentation. By replacing the distraction of meaningful content with filler text of scrambled Latin it allows viewers to focus on graphical elements such as font, typography, and layout.
μ	Viscosity

As can be seen, the symbol column is as wide as the widest symbol (plus some padding) and lengthy text can be put into the description without penalty. Of course, this example is purposefully extreme. We can tweak the example a bit more by putting the second two items under a group:

Symbol Table

ρ	Density
Group 1 Title	
LongNotRealSymbol	In publishing and graphic design, lorem ipsum is a placeholder text commonly used to demonstrate the graphic elements of a document or visual presentation. By replacing the distraction of meaningful content with filler text of scrambled Latin it allows viewers to focus on graphical elements such as font, typography, and layout.
μ	Viscosity

By default, the section level used by `\Group` is one below that of the main nomenclature section; therefore,

since the nomenclature's section level is defined as `subsection`, the `\Group` is a `subsubsection`. Not shown: using `\Subgroup` would typeset the title as a `paragraph` in this example.

5.1.3 Customization

As mentioned, there are several options available to the user for customizing the nomenclature. These options are set by giving a comma-separated list of key-value pairs to the function `\NomenclatureSetup`

<code>\NomenclatureSetup</code>	<code>\NomenclatureSetup{<key-value CSV>}</code>
---------------------------------	--

The format is more appropriately shown as

```
\NomenclatureSetup {
    key-one = option,
    key-two = {option two},
    ...
    key-n = {option n},
}
```

A table of the keys, meaning, defaults, and allow value is given in [table 2](#).

5.2 Acronym

5.2.1 Description

The `Acronym` environment is a specialized extension of the `Nomenclature` environment. It has the same basic syntax, but a `units` column is not supported. Also, instead of `\Entry` taking `(symbol, description)` pairs, it takes `(acronym, meaning)` pairs. Lastly, it comes equipped with a new command: `\Acro`.

<code>\Acro</code>	<code>\Acro{<i>acronym</i>}</code>
--------------------	------------------------------------

`\Acro` is meant to be used throughout the document to reference back to the `Acronym` environment where it was defined. If an `Acronym` environment contains the line `\Entry{TBD}{To be determined}`, the first usage of `\Acro{TBD}` will be typeset as 'To be determined (TBD)' while subsequent uses will simply be 'TBD'. Also, if links are not turned off (they are on by default), the acronym will be a link back to the original environment entry.

<code>\AcronymSetup</code>	<code>\AcronymSetup{<i>key-value CSV</i>}</code>
----------------------------	--

An exact copy of `\NomenclatureSetup`.

5.2.2 Example

The following input

```
\AcronymSetup {
    main-section = section,
    main-title = {Acronym Table},
    entry-padding = 1in
}
\begin{Acronym}
    \Entry{RCCS}{Reactor Cavity Cooling System}
    \Entry{NRC}{Nuclear Regulatory Commission}
\end{Acronym}
```

is typeset as

Acronym

RCCS	Reactor Cavity Cooling System
NRC	Nuclear Regulatory Commission

The first usage of `\Acro{NRC}` is `Nuclear Regulatory Commission (NRC)' while the second usage is `\NRC`.

TABLE 2: *List of key-value pairs for Nomenclature customization.*

Key	Meaning	Default	Allow value
title-skip	Vertical space following the printed title	0pt	dimension
print-skip	Vertical space following a printing of entries	1em	dimension
entry-margin-left	Horizontal margin left of an entry	1em	dimension
entry-margin-bottom	Vertical margin below a printed entry	0.25em	dimension
entry-padding	Horizontal space between columns	0.75em	dimension
main-section	Section level for Main group	chapter	section
group-section	Section level for <code>\Group</code> command	section	section
subgroup-section	Section level for <code>\Subgroup</code> command	subsection	section
main-title	Title for the nomenclature	Nomenclature	text
group-title	Title for the <code>\Group</code> command	---	text
subgroup-title	Title for the <code>\Subgroup</code> command	---	text
include-in-toc	Include the nomenclature in the TOC	true	boolean
with-units	Include a units column	false	boolean

TABLE 3: *Additional key-value pairs for Acronym environment.*

Key	Meaning	Default	Allow value
use-links	Create hyperlink to Acronym entry	true	boolean
link-color	Color of hyperlink text	blue	color

Feature Set 6

Math

As the feature name may suggest, all of the commands in this section deal with mathematical typesetting.

6.1 Derivative Commands

These command set deal with quick and easy typesetting of derivatives.

<code>\deriv</code>	<code>\deriv {⟨function⟩} {⟨variable⟩} {⟨order⟩}</code>
<code>\pderiv</code>	<code>\pderiv {⟨function⟩} {⟨variable⟩} {⟨order⟩}</code>
<code>\tderiv</code>	<code>\tderiv {⟨function⟩} {⟨variable⟩} {⟨order⟩}</code>

This function set is meant to typeset three different kinds of derivatives: ordinary, partial, and total (i.e., material or Lagrangian). The only difference between them is the differential symbol: `\deriv` uses ``d'`, `\pderiv` uses ``∂'`, and `\tderiv` used ``D'`.

These commands typeset the derivative of a given `{⟨function⟩}` with respect to `{⟨variable⟩}` of n -th `{⟨order⟩}` using Leibniz's notation. The `{⟨order⟩}` is optional and defaults to empty (first derivative). For example, the input

```
\begin{align}
&\deriv{y}{x}{2} + \deriv{y}{x} + y(x) && \&= 0 && \\\[0.50em]
&\pderiv{T}{t} - \alpha \pderiv{T}{z}{2} && \&= 0 && \\\[0.50em]
&\tderiv{\rho{u}}{t} + \pderiv{P}{z} - \rho g && \&= 0
\end{align}
```

and is typeset as

$$\frac{d^2y}{dx^2} + \frac{dy}{dx} + y(x) = 0 \quad (1)$$

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial z^2} = 0 \quad (2)$$

$$\frac{D(\rho u)}{Dt} + \frac{\partial P}{\partial z} - \rho g = 0 \quad (3)$$

<code>\derivbig</code>	<code>\derivbig</code>	<code>[\langle left delim \rangle]</code>	<code>\{\langle function \rangle\}</code>	<code>[\langle right delim \rangle]</code>	<code>\{\langle variable \rangle\}</code>	<code>\{\langle order \rangle\}</code>
<code>\pderivbig</code>	<code>\pderivbig</code>	<code>[\langle left delim \rangle]</code>	<code>\{\langle function \rangle\}</code>	<code>[\langle right delim \rangle]</code>	<code>\{\langle variable \rangle\}</code>	<code>\{\langle order \rangle\}</code>
<code>\tderivbig</code>	<code>\tderivbig</code>	<code>[\langle left delim \rangle]</code>	<code>\{\langle function \rangle\}</code>	<code>[\langle right delim \rangle]</code>	<code>\{\langle variable \rangle\}</code>	<code>\{\langle order \rangle\}</code>

This function set is identical to the non-**big** versions above, except that `\{\langle function \rangle\}` is placed to the right of the derivative operator and wrapped by `\left` and `\right`. The default delimiters for the stretch commands are ``` and `'`, and either can be individually overridden via the two optional arguments. For example, the input

```
\begin{align}
&-\derivbig{ p(x) \deriv{y}{x} }{x} + \\
&\quad q(x) (1 - \lambda) y(x) \quad \&= 0 \quad \ll[0.50em] \\
&\tderivbig{ \rho{i} + \frac{1}{2} \rho u^2 }{[t]} - \\
&\quad \pderivbig[\lvert]{ \kappa \pderiv{T}{z} }{z} \quad \&= 0 \\
\end{align}
```

and is typeset as

$$-\frac{d}{dx}\left[p(x)\frac{dy}{dx}\right] + q(x)(1 - \lambda)y(x) = 0 \quad (4)$$

$$\frac{D}{Dt}\left[\rho i + \frac{1}{2}\rho u^2\left(-\frac{\partial}{\partial z}\left|\kappa\frac{\partial T}{\partial z}\right|\right)\right] = 0 \quad (5)$$

<hr/>	<code>\DerivativeGeneral</code>	<code>\DerivativeGeneral</code>	<code>{\langle function \rangle}</code>	<code>{\langle variable \rangle}</code>	<code>{\langle order \rangle}</code>	<code>{\langle symbol \rangle}</code>
	<code>\DerivativeGeneralBig</code>	<code>\DerivativeGeneralBig</code>	<code>{\langle function \rangle}</code>	<code>{\langle variable \rangle}</code>	<code>{\langle order \rangle}</code>	<code>{\langle symbol \rangle}</code>
<hr/>				<code>{\langle left delim \rangle}</code>	<code>{\langle right</code>	<code>delim \rangle}</code>

These commands are lower-level commands used by the `deriv` family above. All of the arguments are mandatory. If a change to the general style of the derivatives or another version of the `deriv` family is desired, these commands are available for usage.

<code>\derivSymbol</code>	<code>\derivSymbol</code>
<code>\pderivSymbol</code>	
<code>\tderivSymbol</code>	These commands take no arguments and expand to the current symbol used for the associated <code>deriv</code> command. The defaults require math mode to be typeset. Therefore, <code>\pderivSymbol</code> will be appear as ∂ .

 $\backslash\mathrm{derivSymbolChange}$ $\backslash\mathrm{derivSymbolChange} \{\langle symbol \rangle\}$
 $\backslash\mathrm{pderivSymbolChange}$

 $\backslash\mathrm{tderivSymbolChange}$

These commands will TEMPORARILY change the symbol used by the associated `deriv` commands. The symbol will revert back to the original, default value after leaving the T_EX group where the switch was made (more often than not for L^AT_EX users, this means ``upon exiting an environment"). For example:

```
\begin{equation}
  \mathrm{deriv}\{U\}{t} =
  \mathrm{derivSymbolChange}\{\delta\}
  \mathrm{deriv}\{Q\}{t} - \mathrm{deriv}\{W\}{t}
\end{equation}
```

typesets as

$$\frac{dU}{dt} = \frac{\delta Q}{\delta t} - \frac{\delta W}{\delta t} \quad (6)$$

and now, after the environment, the `\derivSymbol` is once again ``d'`.

 $\backslash\mathrm{derivSymbolChangeDefault}$ $\backslash\mathrm{derivSymbolChangeDefault} \{\langle symbol \rangle\}$
 $\backslash\mathrm{pderivSymbolChangeDefault}$ $\backslash\mathrm{pderivSymbolChangeDefault} \{\langle symbol \rangle\}$

 $\backslash\mathrm{tderivSymbolChangeDefault}$ $\backslash\mathrm{tderivSymbolChangeDefault} \{\langle symbol \rangle\}$

These commands will PERMANENTLY change the symbol used by the associated `deriv` commands. For example:

```
\begin{equation}
  \mathrm{deriv}\{U\}{t} =
  \mathrm{derivSymbolChangeDefault}\{\delta\}
  \mathrm{deriv}\{Q\}{t} - \mathrm{deriv}\{W\}{t}
\end{equation}
```

typesets as

$$\frac{dU}{dt} = \frac{\delta Q}{\delta t} - \frac{\delta W}{\delta t} \quad (7)$$

and now, after the environment, the `\derivSymbol` is ``δ'`.

`\DelimiterChangeDefault` `\DelimiterChangeDefault {⟨left delim⟩} {⟨right delim⟩}`

This command changes the default delimiters used by the `big` commands above. Any valid delimiters can be used. For example:

```
\DelimiterChangeDefault{({})}
\begin{equation}
-\derivbig{ p(x) \deriv{y}{x} }{x} +
q(x) (1 - \lambda) y(x) = 0 \\\[0.50em]
\end{equation}
```

and is typeset as

$$-\frac{\delta}{\delta x}\left(p(x)\frac{\delta y}{\delta x}\right) + q(x)(1 - \lambda)y(x) = 0 \quad (8)$$

and notice that the `\derivSymbol` is still δ .

6.2 Operators

These operators are added to the standard set using the \mathcal{AMS} operator system. Some are new while others are simply in a camel-cased versions of the standard ones.

`\Sup`
`\Inf`

Supremum and Infimum operators using the math operator system. For example, the input

```
\begin{align}
\Inf_{x \in \mathbb{R}} \{0 < x < 1\} &= 0 \\\[0.50em]
\Sup_{x \in \mathbb{R}} \{0 < x < 1\} &= 1
\end{align}
```

is typeset as

$$\inf_{x \in \mathbb{R}} \{0 < x < 1\} = 0 \quad (9)$$

$$\sup_{x \in \mathbb{R}} \{0 < x < 1\} = 1 \quad (10)$$

`\Lim`

The limit operator:

```
\begin{equation}
\Lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = \mathrm{e}
\end{equation}
```

is typeset as

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e \quad (11)$$

`\Min`
`\Max`

The maximum and minimum value operators

```
\begin{equation}
\begin{align}
\Min_{x \in \mathbb{R}} \Sin(x) &= -1 \quad \ll[0.50em]
\Max_{x \in \mathbb{R}} \Sin(x) &= +1
\end{align}
\end{equation}
```

is typeset as

$$\Min_{x \in \mathbb{R}} \Sin(x) = -1 \quad (12)$$

$$\Max_{x \in \mathbb{R}} \Sin(x) = +1 \quad (13)$$

`\ArgMin`
`\ArgMax`

The maximum and minimum argument operators

```

\begin{equation}
\begin{align}
\ArgMin_{x \in \mathbb{R}} \Sin(x) &= \frac{3\pi}{2} + 2 \pi n \quad \ll[0.50em] \\
\ArgMax_{x \in \mathbb{R}} \Sin(x) &= \frac{\pi}{2} + 2 \pi n
\end{align}
\end{equation}

```

is typeset as

$$\ArgMin_{x \in \mathbb{R}} \Sin(x) = \frac{3\pi}{2} + 2\pi n \quad (14)$$

$$\ArgMax_{x \in \mathbb{R}} \Sin(x) = \frac{\pi}{2} + 2\pi n \quad (15)$$

`\Abs`
`\Ln`
`\Log`
`\Exp`

Common set of operators in uppercase form.

`\Cos`
`\Sin`
`\Tan`
`\Sec`
`\Csc`
`\Cot`

Standard trigonometric functions and their reciprocals.

`\Cosh`
`\Sinh`
`\Tanh`
`\Sech`
`\Csch`
`\Coth`

Hyperbolic trigonometric functions and their reciprocals.

<code>\ArcCos</code>	Standard inverse trigonometric functions and their reciprocals.
<code>\ArcSin</code>	
<code>\ArcTan</code>	
<code>\ArcSec</code>	
<code>\ArcCsc</code>	
<code>\ArcCot</code>	

<code>\ArcCosh</code>	Hyperbolic inverse trigonometric functions and their reciprocals.
<code>\ArcSinh</code>	
<code>\ArcTanh</code>	
<code>\ArcSech</code>	
<code>\ArcCsch</code>	
<code>\ArcCoth</code>	

6.3 Miscellaneous Commands

<code>\Sqrt</code>	<code>\Sqrt</code> [<i>n</i>] { <i>argument</i> }
--------------------	---

This command typesets the [*n*]-th root of a given {*argument*} with a closing tail. This command differs from the default `\sqrt` in appearance only:

$$\sqrt[n]{\frac{f(x)}{g(x)}} = \sqrt[n]{\frac{f(x)}{g(x)}} \quad (16)$$

<code>\IfMathModeTF</code>	<code>\IfMathModeTF</code> { <i>math mode code</i> } { <i>text mode code</i> }
----------------------------	--

This is an abstraction of `expl3`'s `\mode_if_math:TF` function. It was added to give more control on the following `\subs` and `\sup`s commands since `expl3`'s syntax is disabled to make `_` a subscript shift and not a letter.

<code>\subs</code>	<code>\subs</code>	<code>[\space]</code>	<code>{\text subscript}</code>		
<code>\sups</code>	<code>\sups</code>	<code>[\space]</code>	<code>{\text superscript}</code>		
<code>\subsup</code>	<code>\subsup</code>	<code>[\subscript space]</code>	<code>{\text subscript}</code>	<code>[\superscript space]</code>	<code>{\text superscript}</code>

These command typeset a subscript or superscript IN TEXT MODE. They are useful if the subscript or superscript are not variable, and therefore should be in non-math text, or for making subscripts or superscripts in text mode. The optional argument `[\space]` is meant for adjusting the spacing of the scripts and exists in IN MATH MODE, so technically, any valid math statement can be used. However, it is encouraged to only use this argument for spacing. For example, the input ``T\subs{P}`, `$T\subs{P}$`, `T_P` is typeset as ``TP`, `TP`, `TP`', and the input ``T\subs[!]{P}`, `T\subs[:]{P}` is typeset as ``TP`, `TP`'. `TP`

<code>\OneOver</code>	<code>\OneOver</code>	<code>{\denominator}</code>
<code>\oneo</code>		

A simple command the typesets a fraction whose numerator is always one. For example, the input

```
\begin{equation}
\OneOver{\Sqrt{x^2 + 1}}
\end{equation}
```

is typeset as

$$\frac{1}{\sqrt{x^2 + 1}} \quad (17)$$

<code>\dd</code>	<code>\dd</code>	<code>{\variable}</code>
------------------	------------------	--------------------------

A simple command the typesets a non-math ``d` in math mode and is meant to be used for differentials. For example, the input

```
\derivSymbolChangeDefault{\mathrm{d}}
\begin{equation}
f(b) - f(a) = \int_a^b \deriv{f}{t} \dd{t}
\end{equation}
```

is typeset as

$$f(b) - f(a) = \int_a^b \frac{df}{dt} dt \quad (18)$$

`\dprime` `\dprime`
`\tprime`

These commands take no arguments and simply mean 'double prime' and 'triple prime'. For example, the input

```
\begin{equation}
q^{\prime} = q^{\dprime} 2\pi\{R\} = q^{\tprime} \pi\{R^2\}
\end{equation}
```

is typeset as

$$q' = q''2\pi R = q''' \pi R^2 \quad (19)$$

Feature Set 7

Programming

The [Implementation section](#) for this module outlines the programming layer used for the class. There is a user-facing API but is not documented here as it is experimental.

Part II

Implementation

Module 1

Front Matter

Much of this class is written using the L^AT_EX3 Programming Layer; this will be denoted as **expl3**. The **expl3** is the first piece of a new system designed to succeed L^AT_EX 2_ε in the future. However, while the programming layer is solid and remarkable, a lot of presentation work still needs to be done. Therefore, this class uses L^AT_EX 2_ε code where necessary and will hopefully be slowly pulled out as needed. The good news is that since everything is more-or-less an abstraction of T_EX, it should work together well.

1.1 expl3 Package and Identification

The **expl3** package loads the **expl3** and is therefore required. If the package is not recent enough, the class aborts and requests the user update.

```

1 \RequirePackage{expl3}[2013/07/28]
2 \@ifpackagelater{expl3}{2013/07/28} {} {%
3   \PackageError{UWMadThesis}{Version of l3kernel is too old}
4   {%
5     Please install an up to date version of l3kernel\MessageBreak
6     using your TeX package manager or from CTAN.
7   }%
8   \endinput
9 }%

10 \ExplSyntaxOn
```

1.2 Identification and Defaults

If the **expl3** package is recent enoughw, define some identification variables (token lists).

```

11 \tl_const:Nn \c__UWMad_Class_Name_tl      {UWMadThesis}
12 \tl_const:Nn \c__UWMad_Class_Version_tl   {1.0}
13 \tl_const:Nn \c__UWMad_Class_Date_tl      {2014/04/01}
14 \tl_const:Nn \c__UWMad_Class_Description_tl {
```

```

15   LaTeX2e+~Thesis~Class~for~UW~Madison
16 }
17 \tl_const:Nn \c__UWMad_UniversityLong_tl   {University-of-Wisconsin--Madison}
18 \tl_const:Nn \c__UWMad_UniversityShort_tl  {UW--Madison}

```

Assuming the the `expl3` package is recent enough, we provide the class using the `expl3`'s `provide` function.

```

19 \ProvidesExplClass
20   {\c__UWMad_Class_Name_tl}   {\c__UWMad_Class_Date_tl}
21   {\c__UWMad_Class_Version_tl}{\c__UWMad_Class_Description_tl}

```

In an effort to allow the thesis class to adapt to new underlying classes, the class that the `UWMadThesis` class loads is decalred as a mutable token list. The default is the L^AT_EX base class `report`.

```

22 \tl_new:N      \g_UWMad_ParentClass_tl
23 \tl_gset:Nn    \g_UWMad_ParentClass_tl {report}

```

1.3 Options

This command is used to suppress warning issued from `UWMadThesis` class. The first argument is a coonditional that would normally determine if a warning were to be thrown, but the decision is now superceeded by a switch to determine if warnings are disabled or not.

```

24 \cs_new:Nn \__UWMad_ThrowWarnings:NTF {
25   \bool_if:NTF \g__UWMad_ThrowWarnings_bool {
26     \bool_if:NTF #1 {
27       #2
28     } {
29       #3
30     }
31   } {}
32 }
33 \cs_new:Nn \__UWMad_ThrowWarnings:TF {
34   \bool_if:NTF \g__UWMad_ThrowWarnings_bool {
35     #1
36   } {
37     #2
38   }
39 }

```

First, a command is created to throw a warning if an option that violates University of Wisconsin--Madison's dissertation guidelines.

```

40 \msg_new:nnn{UWMadThesis}{Options/StyleViolation}{

```

```

41     Option~'#1'~violates~\c_UWMadUniversityShort_tl{}~
42     Dissertation~Guidelines;~consider~omission
43 }
44 \cs_new:Nn \__UWMad_FrontMatter_StyleWarning:n {
45     \__UWMad_ThrowWarnings:TF {
46         \msg_warning:nnn {UWMadThesis}{Options/StyleViolation}
47         {#1}
48     } { }
49     \PassOptionsToClass{#1}{\g_UWMad_ParentClass_tl}
50 }

```

Now, declare booleans for the option processing. All new booleans are false by default.

```

51 \bool_new:N \g__UWMad_MathTweaks_bool
52 \bool_gset_true:N \g__UWMad_MathTweaks_bool
53 \bool_new:N \g__UWMad_ThrowWarnings_bool
54 \bool_gset_true:N \g__UWMad_ThrowWarnings_bool

```

Declare the options.

```

55 \DeclareOption{NoMath} {
56     \bool_gset_false:N \g__UWMad_MathTweaks_bool
57 }
58 \DeclareOption{Quiet} {
59     \bool_gset_false:N \g__UWMad_ThrowWarnings_bool
60 }

```

Catch the couple of default options that violate the requirements: 8.5 by 11 paper for single-sided printing.

```

61 \DeclareOption{a4paper} {
62     \__UWMad_ThrowWarnings:TF {
63         \__UWMad_FrontMatter_StyleWarning:n {\CurrentOption}
64     } { }
65 }
66 \DeclareOption{twoside} {
67     \__UWMad_ThrowWarnings:TF {
68         \__UWMad_FrontMatter_StyleWarning:n {\CurrentOption}
69     } { }
70 }

```

These options change the default report class to the ones indicated.

```

71 \DeclareOption{article} {
72     \tl_gset:Nn \g_UWMad_ParentClass_tl {article}
73 }

```

This is a special class option for generating the documentation. Users should not use this unless they know what they're doing. The line below the `ParentClass` class prevents the `thumbpdf` package from being loaded.

```

74 \DeclareOption{l3doc} {
75     \tl_gset:Nn \g_UWMad_ParentClass_tl {l3doc}
76     \tl_const:cn {ver@thumbpdf.sty} {}
77 }

```

Pass all remaining options to the base class.

```

78 \DeclareOption*{
79     \PassOptionsToClass
80     {\CurrentOption}{\g_UWMad_ParentClass_tl}
81 }

```

Process the options with some defaults and load the base class.

```

82 \ExecuteOptions{oneside,12pt}
83 \ProcessOptions\relax
84 \LoadClass{\g_UWMad_ParentClass_tl}[1995/12/01]

```

1.4 Package Loads

Load some packages that give nice features and are not hyperlink sensitive.

```

85 \RequirePackage{xparse}
86 \RequirePackage{fixltx2e}
87 \RequirePackage{microtype}
88 \RequirePackage{array}
89 \RequirePackage{graphicx}
90 \RequirePackage{setspace}
91 \RequirePackage{geometry}

```

Load the \mathcal{AMS} suite.

```

92 \RequirePackage{amsmath}
93 \RequirePackage{amsfonts}
94 \RequirePackage{amssymb}
95 \RequirePackage{mathtools}

```

And now we load some packages that give nice features and are hyperlink sensitive.

```

96 \RequirePackage[noabbrev,nameinlink]{cleveref}
97 \RequirePackage[usenames,dvipsnames,svgnames,table,hyperref]{xcolor}
98 \RequirePackage{caption}
99 \RequirePackage{subcaption}

```


If links were not negated by the options, `bookmark` and `hyperref` are loaded.

```
100 \RequirePackage{hyperref}
101 \RequirePackage{bookmark}
```

Conditionally load either the `polyglossia` or `babel` language packages depending on the engine in use.

```
102 \bool_if:nTF {\xetex_if_engine_p: || \luatex_if_engine_p:} {
103   \RequirePackage{fontspec}
104   \setmainfont
105     [SmallCapsFont = {Latin~Modern~Roman~Caps}]
106     {Latin~Modern~Roman}
107   %
108   \RequirePackage{polyglossia}
109   \setmainlanguage[variant = usmax]{english}
110 } {
111   \RequirePackage[T1]{fontenc}
112   \RequirePackage{lmodern}
113   %
114   \RequirePackage[english]{babel}
115 }
```

And since these identifications may be desired in typesetting more, where `expl3`'s syntax will be turned off, we define some aliases.

```
116 \DeclareDocumentCommand \UWMadClass { } {
117   \texttt{\c__UWMad_Class_Name_tl}~class
118 }
119 \DeclareDocumentCommand \UWMadClassVersion { } {
120   \c__UWMad_Class_Version_tl
121 }
122 \DeclareDocumentCommand \UWMadClassDate { } {
123   \c__UWMad_Class_Date_tl
124 }
125 \DeclareDocumentCommand \UWMadLong { } {
126   \c__UWMad_UniversityLong_tl
127 }
128 \DeclareDocumentCommand \UWMadShort { } {
129   \c__UWMad_UniversityShort_tl
130 }
```

Module 2

Programming

This section outlines the Programming module for the `UWMadThesis` class. It encompasses thin abstractions from the standard `expl3`'s type and collection systems and provides $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X } 2_{\varepsilon}$ abstractions for several other features.

2.1 Utility Commands

Define some messages for the rest of the module.

```

131 \msg_new:nnn {UWMadThesis} {Programming/UnregisteredVariable} {
132   `#1'~is~not~a~registered~#2.~~The~#2~must~be~defined~
133   before~usage~by~the~function~\string\UWMad_#2_DefineLocal:n~or~
134   \string\UWMad_#2_DefineGlobal:n.
135 }
136 \msg_new:nnn {UWMadThesis} {Programming/Undefined} {
137   The~#2~`#1'~is~undefined.~~The~#2~must~be~defined~
138   before~usage~by~the~function~\string\UWMad_#2_Define:n.
139 }
140 \msg_new:nnn {UWMadThesis} {Programming/Defined} {
141   The~#2~`#1'~is~already~defined~and~will~not~altered.
142 }
```

`\UWMad_Hook_Prepnd:nn` These commands allow additional code to be prepended or appended to a specified command.

`\UWMad_Hook_Append:nn`

```

143 \cs_new:Nn \UWMad_Hook_Prepnd:nn{
144   \cs_new_eq:cc {#1-Default:} {#1}
145   \cs_gset:cn {#1:} {#2 \cs:w #1-Default:\cs_end:}
146   \cs_undefine:c {#1}
147   \cs_new_eq:cc {#1} {#1:}
148 }
149 \cs_new:Nn \UWMad_Hook_Append:nn{
150   \cs_new_eq:cc {#1-Default:} {#1}
151   \cs_gset:cn {#1:} {\cs:w #1-Default:\cs_end: #2}
152   \cs_undefine:c {#1}
153   \cs_new_eq:cc {#1} {#1:}
154 }
155 \cs_new:Nn \UWMad_Hook_Prepnd:Nn{
156   \cs_new_eq:cN {\string#1-Default:} #1
```

```

157     \cs_gset:cn    {\string#1:}          {#2 \cs:w\string#1-Default:\cs_end:}
158     \cs_undefine:N #1
159     \cs_new_eq:Nc  #1          {\string#1:}
160 }
161 \cs_new:Nn \UWMad_Hook_Append:Nn{
162     \cs_new_eq:cN  {\string#1-Default:} #1
163     \cs_gset:cn    {\string#1:}          {\cs:w\string#1-Default:\cs_end: #2}
164     \cs_undefine:N #1
165     \cs_new_eq:Nc  #1          {\string#1:}
166 }

```

`\UWMad_Definition_Swap:nn` These commands “swap” in a new definition of a command and, when called, reset it to it's default
`\UWMad_Definition_Reset:nn` definition.

```

167 \cs_new:Nn \UWMad_Definition_Swap:Nn {
168     \cs_if_exist:NTF #1 {
169         \cs_new_eq:cN  {\string#1-Default:} #1
170         \cs_gset_eq:Nc #1 {#2}
171     } {
172         \cs_new:Nn #1 {#2}
173     }
174 }
175 \cs_new:Nn \UWMad_Definition_Reset:N {
176     \cs_if_exist:cTF {\string#1-Default:} {
177         \cs_gset_eq:Nc #1          {\string#1-Default:}
178         \cs_undefine:c  {\string#1-Default:}
179     } { }
180 }
181 \cs_generate_variant:Nn \UWMad_Definition_Swap:Nn {cn}
182 \cs_generate_variant:Nn \UWMad_Definition_Reset:N  {c}

```

`__UWMad_IfDefined:nnnnT` These commands accept a $\{\langle Prefix \rangle\}$, an $\{\langle ID \rangle\}$, a $\{\langle Suffix \rangle\}$, a $\{\langle Type \rangle\}$, and $\{\langle Code \rangle\}$. It determines if a
`__UWMad_IfUndefined:nnnnT` command named by the concatenation of $\{\langle Prefix \rangle\}$, $\{\langle ID \rangle\}$, and $\{\langle Suffix \rangle\}$ is defined or not and executes
 $\{\langle Code \rangle\}$ depending on the existence.

Usage:

`__UWMad_IfUndefined:nnnnT $\{\langle Prefix \rangle\}\{\langle ID \rangle\}\{\langle Suffix \rangle\}\{\langle Type \rangle\}\{\langle Code \rangle\}$`

```

183 \cs_new:Nn \__UWMad_IfDefined:nnnnT{
184     \cs_if_exist:cTF {#1#2#3} {
185         #5
186     }{
187         \msg_error:nnnn
188             {UWMadThesis}
189             {Programming/Undefined}
190             {#2}
191             {#4}
192     }
193 }

```

```

194 \cs_new:Nn \__UWMad_IfUndefined:nnnnT{
195     \cs_if_free:cTF {#1#2#3} {
196         #5
197     }{
198         \msg_warning:nnnn
199             {UWMadThesis}
200             {Programming/Defined}
201             {#2}
202             {#4}
203     }
204 }

```

`__UWMad_IfDefined:nT` These commands are simplifications of the above commands and that only take a $\{\langle CommandName \rangle\}$ and `__UWMad_IfUndefined:nT` $\{\langle TrueCode \rangle\}$.

Usage:

`__UWMad_IfUndefined:nT $\{\langle CommandName \rangle\}$ $\{\langle TrueCode \rangle\}$`

```

205 \cs_new:Nn \__UWMad_IfDefined:nT{
206     \__UWMad_IfDefined:nnnnT{#1}{-}{-}{command}{#2}
207 }
208 \cs_new:Nn \__UWMad_IfUndefined:nT{
209     \__UWMad_IfUndefined:nnnnT{#1}{-}{-}{command}{#2}
210 }

```

2.2 Collections

In the following subsections, commands that create and manipulate various collection data types will be discussed. The collections currently implemented are stacks (LIFO), queues (FIFO), dequeues (LIFO+FIFO), and hashes (key-value pairs).

All of the collection systems are thin abstractions of `expl3`'s `l3tl`, `l3seq`, and `l3prop` modules to avoid developing one-shot systems while allowing more endeavoring authors access to the features without learning $\text{\LaTeX}3$ programming if they load the abstractions.

2.2.1 Stacks

This set of commands is a simple system for creating and working with stacks. Stacks are a last-in first-out collection data type; this means that the data element (in this any unexpanded token/token list) last

pushed on to the stack is the first popped. Data elements can also be walked (iterated over) with an inline callback in a LIFO sense.

`_UWMad_Stack_IfDefined:nT` Shortcuts for the more general commands outlined above.

`_UWMad_Stack_IfUndefined:nT`

```

211 \cs_new:Nn \_UWMad_Stack_IfDefined:nT {
212   \_UWMad_IfDefined:nnnnT{g__UWMad_Stack_}{#1}{}{Stack}{#2}
213 }
214 \cs_new:Nn \_UWMad_Stack_IfUndefined:nT{
215   \_UWMad_IfUndefined:nnnnT{g__UWMad_Stack_}{#1}{}{Stack}{#2}
216 }
```

`\UWMad_Stack_Define:n` Define a new Stack.

```

217 \cs_new:Nn \UWMad_Stack_Define:n {
218   \_UWMad_Stack_IfUndefined:nT {#1} {
219     \tl_new:c {g__UWMad_Stack_#1}
220   }
221 }
```

`\UWMad_Stack_Clear:n` Clear but do not undefine a defined Stack.

```

222 \cs_new:Nn \UWMad_Stack_Clear:n {
223   \_UWMad_Stack_IfDefined:nT {#1} {
224     \tl_gclear:c {g__UWMad_Stack_#1}
225   }
226 }
```

`\UWMad_Stack_Delete:n` Clear and undefine a defined Stack.

```

227 \cs_new:Nn \UWMad_Stack_Delete:n {
228   \_UWMad_Stack_IfDefined:nT {#1} {
229     \tl_gclear:c {g__UWMad_Stack_#1}
230     \cs_undefine:c {g__UWMad_Stack_#1}
231   }
232 }
```

`\UWMad_Stack_Push:nn` Push a value on to a defined Stack.

```

233 \cs_new:Nn \UWMad_Stack_Push:nn {
234   \_UWMad_Stack_IfDefined:nT {#1} {
235     \tl_gput_left:cn {g__UWMad_Stack_#1} {#2}
236   }
237 }
238 %
```

```

239 %
240 \cs_generate_variant:Nn \tl_head:N { c }
241 \cs_generate_variant:Nn \tl_tail:N { c }

```

\UWMad_Stack_Pop:n Pop a value off a defined Stack and place it in the input stream.

```

242 \cs_new:Nn \UWMad_Stack_Pop:n {
243   \__UWMad_Stack_IfDefined:nT {#1} {
244     \tl_set:Nf \l_tmpa_tl {\tl_head:c {g__UWMad_Stack_#1}}
245     \tl_set:cf {g__UWMad_Stack_#1} {\tl_tail:c {g__UWMad_Stack_#1}}
246     \tl_use:N \l_tmpa_tl
247   }
248 }

```

\UWMad_Stack_Walk:nn Iterate of the elements of a defined Stack in a FILO sense with supplied code.

```

249 \cs_new:Nn \UWMad_Stack_Walk:nn {
250   \tl_map_inline:cn {g__UWMad_Stack_#1} {#2}
251 }

```

2.2.2 Queues

This set of commands is a simple system for creating and working with queue. Queues are a first-in first-out collection data type; this means that the data element (in this any unexpanded token/token list) first pushed on to the queue is the first popped. Data elements can also be walked (iterated over) with an inline callback in a FIFO sense.

__UWMad_Queue_IfDefined:nT Shortcuts for the more general commands outlined above.

```

\__UWMad_Queue_IfUndefined:nT
252 \cs_new:Nn \__UWMad_Queue_IfDefined:nT {
253   \__UWMad_IfDefined:nnnT{g__UWMad_Queue_}{#1}{Queue}{#2}
254 }
255 \cs_new:Nn \__UWMad_Queue_IfUndefined:nT{
256   \__UWMad_IfUndefined:nnnT{g__UWMad_Queue_}{#1}{Queue}{#2}
257 }

```

\UWMad_Queue_Define:n Define a new Queue.

```

258 \cs_new:Nn \UWMad_Queue_Define:n {
259   \__UWMad_Queue_IfUndefined:nT {#1} {
260     \tl_new:c {g__UWMad_Queue_#1}
261   }
262 }

```

`\UWMad_Queue_Clear:n` Clear but do not undefine a defined Queue.

```

263 \cs_new:Nn \UWMad_Queue_Clear:n {
264     \__UWMad_Queue_IfDefined:nT {#1} {
265         \tl_gclear:c {g__UWMad_Queue_#1}
266     }
267 }
```

`\UWMad_Queue_Delete:n` Clear and undefine a defined Queue.

```

268 \cs_new:Nn \UWMad_Queue_Delete:n {
269     \__UWMad_Queue_IfDefined:nT {#1} {
270         \tl_gclear:c {g__UWMad_Queue_#1}
271         \cs_undefine:c {g__UWMad_Queue_#1}
272     }
273 }
```

`\UWMad_Queue_Pop:nn` Push an item on to the start of a defined Queue.

```

274 \cs_new:Nn \UWMad_Queue_Push:nn {
275     \__UWMad_Queue_IfDefined:nT {#1} {
276         \tl_gput_left:cn {g__UWMad_Queue_#1} {{#2}}
277     }
278 }
279 %
280 %
281 \cs_generate_variant:Nn \tl_head:N { c }
282 \cs_generate_variant:Nn \tl_tail:N { c }
```

`\UWMad_Queue_Pop:n` Pop an item from the end of a defined Queue and place it in the input stream.

```

283 \cs_new:Nn \UWMad_Queue_Pop:n {
284     \__UWMad_Queue_IfDefined:nT {#1} {
285         \tl_reverse:c {g__UWMad_Queue_#1}
286         \tl_set:Nf \l_tmpa_tl
287             {\tl_head:c {g__UWMad_Queue_#1}}
288         \tl_set:cf {g__UWMad_Queue_#1}
289             {\tl_tail:c {g__UWMad_Queue_#1}}
290         \tl_reverse:c {g__UWMad_Queue_#1}
291         \tl_use:N \l_tmpa_tl
292     }
293 }
```

`\UWMad_Queue_Walk:nn` Iterate of the elements of a defined Queue in a FIFO sense with supplied code.

```

294 \cs_new:Nn \UWMad_Queue_Walk:nn {
295   \__UWMad_Queue_IfDefined:nT {#1} {
296     \group_begin:
297       \tl_reverse:c {g__UWMad_Queue_#1}
298       \tl_map_inline:cn {g__UWMad_Queue_#1} {#2}
299     \group_end:
300   }
301 }

```

`\UWMad_Queue_IfEmpty:nTF` Execute true/false code depending on the emptiness of a defined Queue.

```

302 \cs_new:Nn \UWMad_Queue_IfEmpty:nTF {
303   \__UWMad_Queue_IfDefined:nT {#1} {
304     \tl_if_empty:cTF {g__UWMad_Queue_#1}{
305       #2
306     }{
307       #3
308     }
309   }
310 }

```

2.2.3 Deques

This set of commands is a simple system for creating and working with double-ended queues (deques, pronounced *deck*). Deques are a generalization of stacks and queues in that data can be pushed, popped, and walked from either end of the list (i.e., LIFO+FIFO).

`__UWMad_Deque_IfDefined:nT` Shortcuts for the more general commands outlined above.

```

\__UWMad_Deque_IfUndefined:nT
311 \cs_new:Nn \__UWMad_Deque_IfDefined:nT {
312   \__UWMad_IfDefined:nnnnT{g__UWMad_Deque_}{#1}{#2}{Deque}{#2}
313 }
314 \cs_new:Nn \__UWMad_Deque_IfUndefined:nT{
315   \__UWMad_IfUndefined:nnnnT{g__UWMad_Deque_}{#1}{#2}{Deque}{#2}
316 }

```

`\UWMad_Deque_Define:n` Define a new Deque.

```

317 \cs_new:Nn \UWMad_Deque_Define:n {
318   \__UWMad_Deque_IfUndefined:nT {#1} {
319     \seq_new:c {g__UWMad_Deque_#1}
320   }
321 }

```


`\UWMad_Deque_Clear:n` Clear but do not undefine a defined Deque.

```

322 \cs_new:Nn \UWMad_Deque_Clear:n {
323     \__UWMad_Deque_IfDefined:nT {#1} {
324         \seq_gclear:c {g__UWMad_Deque_#1}
325     }
326 }
```

`\UWMad_Deque_Clear:n` Clear and undefine a defined Deque.

```

327 \cs_new:Nn \UWMad_Deque_Delete:n {
328     \__UWMad_Deque_IfDefined:nT {#1} {
329         \seq_gclear:c {g__UWMad_Deque_#1}
330         \cs_undefine:c {g__UWMad_Deque_#1}
331     }
332 }
```

`\UWMad_Deque_PushLeft:nn` Push an element on to the left or right of a defined Deque.

`\UWMad_Deque_PushRight:nn`

```

333 \cs_new:Nn \UWMad_Deque_PushLeft:nn {
334     \__UWMad_Deque_IfDefined:nT {#1} {
335         \seq_gput_left:cn {g__UWMad_Deque_#1} {#2}
336     }
337 }
338 \cs_new:Nn \UWMad_Deque_PushRight:nn {
339     \__UWMad_Deque_IfDefined:nT {#1} {
340         \seq_gput_right:cn {g__UWMad_Deque_#1} {#2}
341     }
342 }
```

`\UWMad_Deque_PushLeft:nn` Pop an element from the left or right of a defined Deque and place it into the input stream.

`\UWMad_Deque_PushRight:nn`

```

343 \cs_new:Nn \UWMad_Deque_PopLeft:n {
344     \__UWMad_Deque_IfDefined:nT {#1} {
345         \seq_gpop_left:cN {g__UWMad_Deque_#1} \l_tmpa_tl
346         \tl_use:N \l_tmpa_tl
347     }
348 }
349 \cs_new:Nn \UWMad_Deque_PopRight:n {
350     \__UWMad_Deque_IfDefined:nT {#1} {
351         \seq_gpop_right:cN {g__UWMad_Deque_#1} \l_tmpa_tl
352         \tl_use:N \l_tmpa_tl
353     }
354 }
```

`\UWMad_Deque_WalkLeftToRight:nn` Iterate over the elements left-to-right or right-to-left of a defined Deque with supplied code.

`\UWMad_Deque_WalkRightToLeft:nn`

```

355 \cs_new:Nn \UWMad_Deque_WalkLeftToRight:nn {
356   \__UWMad_Deque_IfDefined:nT {#1} {
357     \seq_map_inline:cn {g__UWMad_Deque_#1} {#2}
358   }
359 }
360 \cs_generate_variant:Nn \seq_reverse:N {c}
361 \cs_new:Nn \UWMad_Deque_WalkRightToLeft:nn {
362   \__UWMad_Deque_IfDefined:nT {#1} {
363     \group_begin:
364       \seq_reverse:c {g__UWMad_Deque_#1}
365       \seq_map_inline:cn {g__UWMad_Deque_#1} {#2}
366     \group_end:
367   }
368 }

```

2.2.4 Hashes

This set of commands is a simple system for creating and working with hashes (more often called associative arrays or dictionaries, but erring on the side of usability, Ruby's jargon will be used). Hashes are a type of array that indexes values by (at least in \LaTeX) alphanumeric keys instead of just integers. Data can be set by key, retrieved by key, unset by key, deleted, and walked.

A hash walk, like the collection walks above, iterates through all of the keys and values in the hash while applying a user supplied function. However, unlike the collection walks, **a hash's walk order is not gauranteed to be the set order**. If walk order is needed to be gauranteed, see the previous collection data types.

The system is a thin abstraction of `expl3`'s `l3prop` module to avoid developing a one-shot system while allowing more endeavoring authors access to the feature without learning \LaTeX 3 programming.

```

369 \cs_generate_variant:Nn \prop_gput:Nnn { c x n }
370 \cs_generate_variant:Nn \prop_if_in:NnTF { c x TF }
371 \cs_generate_variant:Nn \prop_if_in:NnTF { c f TF }
372 \cs_generate_variant:Nn \prop_get:Nn { c x }
373 \cs_generate_variant:Nn \prop_get:Nn { c f }
374 \cs_generate_variant:Nn \prop_get:NnNTF { c x N TF }
375 \cs_generate_variant:Nn \prop_gremove:Nn { c x }

```

```

\__UWMad_Hash_IfDefined:nT Shortcuts for the more general commands outlined above.
\__UWMad_Hash_IfUndefined:nT

```

```

376 \cs_new:Nn \__UWMad_Hash_IfDefined:nT {
377   \__UWMad_IfDefined:nnnnT{g__UWMad_Hash_}{#1}{Hash}{#2}
378 }
379 \cs_new:Nn \__UWMad_Hash_IfUndefined:nT{
380   \__UWMad_IfUndefined:nnnnT{g__UWMad_Hash_}{#1}{Hash}{#2}

```

```
381 }
```

`\UWMad_Hash_Define:n` Define a new Hash.

```
382 \cs_new:Nn \UWMad_Hash_Define:n {
383   \__UWMad_Hash_IfUndefined:nT {#1} {
384     \prop_new:c {g__UWMad_Hash_#1}
385   }
386 }
```

`\UWMad_Hash_Set:nnn` Set the value of a key of a defined Hash.

Usage:

`\UWMad_Hash_Set:nnn{<HashID>}{<Key>}{<Value>}`

```
387 \cs_new:Nn \UWMad_Hash_Set:nnn {
388   \__UWMad_Hash_IfDefined:nT {#1} {
389     \prop_gput:cxn {g__UWMad_Hash_#1}{#2}{#3}
390   }
391 }
```

`\UWMad_Hash_Get:nn` Get the value of a key of a defined Hash and place it into the input stream.

```
392 \cs_generate_variant:Nn \prop_get:cn {cf}
393 \cs_new:Nn \UWMad_Hash_Get:nn {
394   \__UWMad_Hash_IfDefined:nT {#1} {
395     \prop_get:cf {g__UWMad_Hash_#1}{#2}
396   }
397 }
```

`\UWMad_Hash_Unset:nn` Undefine a key-value pair in a defined Hash.

```
398 \cs_new:Nn \UWMad_Hash_Unset:nn {
399   \__UWMad_Hash_IfDefined:nT {#1} {
400     \prop_gremove:cx {g__UWMad_Hash_#1} {#2}
401   }
402 }
```

`\UWMad_Hash_IfKeySet:nnTF` Execute true/false code depending on if a key is set in a defined Hash.

```
403 \cs_generate_variant:Nn \tl_to_lowercase:n {f}
404 \cs_new:Nn \UWMad_Hash_IfKeySet:nnTF {
405   \__UWMad_Hash_IfDefined:nT {#1} {
```

```

406     \prop_if_in:cfTF {g__UWMad_Hash_#1} {\tl_trim_spaces:n{#2}} {
407         #3
408     }{
409         #4
410     }
411 }
412 }

```

`\UWMad_Hash_Walk:nn` Iterate over the key-value pairs of a defined Hash with supplied code. **No order is gauranteed.**

```

413 \cs_new:Nn \UWMad_Hash_Walk:nn {
414     \__UWMad_Hash_IfDefined:nT {#1} {
415         \prop_map_inline:cn {g__UWMad_Hash_#1} {#2}
416     }
417 }

```

`\UWMad_Hash_Delete:n` Clear and undefine a defined Hash.

```

418 \cs_new:Nn \UWMad_Hash_Delete:n {
419     \__UWMad_Hash_IfDefined:nT {#1} {
420         \prop_gclear:c {g__UWMad_Hash_#1}
421         \cs_undefine:c {g__UWMad_Hash_#1}
422     }
423 }

```

2.3 User-Level Abstractions

The commands that follow are L^AT_EX 2_ε-like commands that use the `expl3` as the underlying system. **The commands are not loaded by default; they must be invoked by calling the following command.**

2.3.1 Utility Commands

`\IfCommandExists` This command pair is used instead of L^AT_EX's `\@ifundefined`. Since it is ε -T_EX, this command will allow `\IfCommandDoesNotExist` for a switch to `\@ifundefined` if problems arise from non- ε -T_EX users in the future.

Usage:

```
\IfCommandExists{<Command Name>}{<True>}{<False>}
```

```
\IfCommandDoesNotExist{<Command Name>}{<True>}{<False>}
```

```
424 \DeclareDocumentCommand \IfCommandExistsTF { m +m +m } {
425     \cs_if_exist:cTF {#1}{
426         #2
427     }{
428         #3
429     }
430 }
431 \DeclareDocumentCommand \IfCommandDoesNotExistTF { m +m +m } {
432     \cs_if_free:cTF {#1}{
433         #2
434     }{
435         #3
436     }
437 }
```

\IfStringEmpty Checks if a given string is empty. It uses the `etoolbox`'s `\ifblank`. This command will not expand input.

Usage:

```
\IfStringEmpty{<String>}{<True>}{<False>}
```

```
438 \cs_generate_variant:Nn \tl_if_blank:nTF {fTF}
439 \DeclareDocumentCommand \IfEmptyTF { m +m +m } {
440     \tl_if_blank:fTF {#1}{
441         #2
442     }{
443         #3
444     }
445 }
```

\IfCommandEmpty Uses the `etoolbox`'s `\ifdefempty` command to test if a command expands to an empty string and is followed by the given conditional code.

Usage:

```
IfCommandEmpty{<Command>}{<True>}{<False>}
```

```
446 \DeclareDocumentCommand \IfCommandEmptyTF { m +m +m }{
447     \tl_if_blank:oTF{#1}{
448         #2
449     }{
450         #3
451     }
452 }
```

2.3.2 Command Creators

`\MakeCommand` This command pair uses the `etoolbox`'s `\csdef` to define a commands via a supplied string $\{\langle Command Name \rangle\}$ and a set of $\{\langle Code \rangle\}$. If the requested command is not defined, `\MakeCommand` will create it; however, if the requested command is already defined, `\MakeCommand` will throw a warning and not make the command. If the requested command is defined, `\ReMakeCommand` will redefine it; however, if the requested command is not defined, `\ReMakeCommand` will throw a warning and not make the command.

Usage:

`\MakeCommand{\langle Command Name \rangle}{\langle Code \rangle}`
`\ReMakeCommand{\langle Command Name \rangle}{\langle Code \rangle}`

```

453 \DeclareDocumentCommand \MakeCommand { 0{ } m +m } {
454     \cs_if_free:cTF {#2} {
455         \cs_set:cpn {#2} #1 {#3}
456     }{
457         \msg_warning:nnnn
458             {UWMadThesis}{Programming/Defined}{#2}{command}
459     }
460 }
461 \DeclareDocumentCommand \ReMakeCommand { 0{ } m +m }{
462     \cs_if_exist:cTF {#2} {
463         \cs_set:cpn {#2} #1 {#3}
464     }{
465         \msg_error:nnnn
466             {UWMadThesis}{Programming/Undefined}{#2}{command}
467     }
468 }
```

`\MakeGlobalCommand` Similar to `\MakeCommand` except the creation is made regardless of the requested command's definition and the creation is global.

Usage:

`\MakeGlobalCommand{\langle Command Name \rangle}{\langle Code \rangle}`

```

469 \DeclareDocumentCommand \MakeGlobalCommand { 0{ } +m m } {
470     \cs_gset:cpn {#2} #1 {#3}
471 }
```

`\MakeCommandUndefined` Globally undefines the command specified by $\{\langle Command Name \rangle\}$.

Usage:

`\MakeCommandUndefined{\langle Command Name \rangle}`

```

472 \DeclareDocumentCommand \MakeCommandUndefined { m } {
473   \cs_undefine:c {#1}
474 }

```

\CopyCommand Copies the definition of the command named $\{\langle Command Name 1 \rangle\}$ to a new command named $\{\langle Command Name 2 \rangle\}$. If $\{\langle Command Name 2 \rangle\}$ already has a definition, **\CopyCommand** will throw a warning *but* still make the copy.

Usage:

\CopyCommand $\{\langle Command Name 1 \rangle\}\{\langle Command Name 2 \rangle\}$

```

475 \DeclareDocumentCommand \CopyCommand { m m } {
476   \cs_if_free:cTF {#1} {
477     \cs_if_free:cTF {#2} {
478       \cs_gset_eq:cc {#2}{#1}
479     }{
480       \msg_warning:nnnn
481         {UWMadThesis}{Programming/Defined}{#2}{command}
482     }
483   }{
484     \msg_warning:nnnn
485       {UWMadThesis}{Programming/Defined}{#1}{command}
486   }
487 }

```

2.3.3 Types

\CreateBoolean L^AT_EX 2_ε version of the Boolean Type system above.

\CreateBooleanTrue

\CreateBooleanFalse

\SetBooleanTrue

\SetBooleanFalse

\IfBooleanTrueTF

\IfBooleanFalseTF

```

488 \DeclareDocumentCommand \CreateBoolean { m } {
489   \bool_new:c {g__UWMad_Programming_API_#1_bool}
490 }
491 \DeclareDocumentCommand \CreateBooleanTrue { m } {
492   \bool_new:c {g__UWMad_Programming_API_#1_bool}
493   \bool_gset_true:c {g__UWMad_Programming_API_#1_bool}
494 }
495 \DeclareDocumentCommand \CreateBooleanFalse { m } {
496   \bool_new:c {g__UWMad_Programming_API_#1_bool}
497 }
498 \DeclareDocumentCommand \SetBooleanTrue { m } {
499   \bool_gset_true:c {g__UWMad_Programming_API_#1_bool}
500 }
501 \DeclareDocumentCommand \SetBooleanFalse { m } {
502   \bool_gset_false:c {g__UWMad_Programming_API_#1_bool}
503 }

```

```

504 \DeclareDocumentCommand \IfBooleanTrueTF { m +m +m } {
505     \bool_if:cTF {g__UWMad_Programming_API_#1_bool} {
506         #2
507     } {
508         #3
509     }
510 }
511 \DeclareDocumentCommand \IfBooleanFalseTF { m +m +m } {
512     \bool_if:cTF {g__UWMad_Programming_API_#1_bool} {
513         #3
514     } {
515         #2
516     }
517 }

```

`\CreateLength` L^AT_EX 2_ε version of the Boolean Type system above.

```

\AddToLength
\SetLength
\ValueOfLength
\IfLengthTF
518 \DeclareDocumentCommand \CreateLength { m m } {
519     \dim_new:c {g__UWMad_Programming_API_#1_dim}
520     \dim_gset:cn {g__UWMad_Programming_API_#1_dim} {#2}
521 }
522 \DeclareDocumentCommand \AddToLength { m m } {
523     \dim_gadd:cn {g__UWMad_Programming_API_#1_dim} {#2}
524 }
525 \DeclareDocumentCommand \SetLength { m m } {
526     \dim_gset:cn {g__UWMad_Programming_API_#1_dim} {#2}
527 }
528 \DeclareDocumentCommand \ValueOfLength { m } {
529     \dim_use:c {g__UWMad_Programming_API_#1_dim}
530 }
531 \DeclareDocumentCommand \IfLengthTF { m m m +m +m } {
532     \dim_compare:nNnTF {#1} #2 {#3} {
533         #4
534     } {
535         #5
536     }
537 }

```

`\CreateCounter` L^AT_EX 2_ε version of the Counter Type system above.

```

\AddToCounter
\StepCounter
\SetCounter
\ValueOfCounter
\IfCounterTF
\CounterToArabic
\CounterToALPHA
\CounterToAlpha
\CounterToROMAN
\CounterToRoman
538 \DeclareDocumentCommand \CreateCounter { m m } {
539     \int_new:c {g__UWMad_Programming_API_#1_int}
540     \int_gset:cn {g__UWMad_Programming_API_#1_int} {#2}
541 }
542 \DeclareDocumentCommand \AddToCounter { m m } {
543     \int_gadd:cn {g__UWMad_Programming_API_#1_int} {#2}
544 }
545 \DeclareDocumentCommand \StepCounter { m m } {
546     \int_gincr:cn {g__UWMad_Programming_API_#1_int} {#2}
547 }
548 \DeclareDocumentCommand \SetCounter { m m } {
549     \int_gset:cn {g__UWMad_Programming_API_#1_int} {#2}

```



```

550 }
551 \DeclareDocumentCommand \ValueOfCounter { m m } {
552   \int_use:c {g__UWMad_Programming_API_#1_int}
553 }
554 \DeclareDocumentCommand \IfCounterTF { m m m +m +m } {
555   \int_compare:nNnTF {#1} {#2} {#3} {
556     #4
557   } {
558     #5
559   }
560 }
561 \DeclareDocumentCommand \CounterToArabic { m } {
562   \int_to_arabic:c {g__UWMad_Programming_API_#1_int}
563 }
564 \DeclareDocumentCommand \CounterToALPHA { m } {
565   \int_to_Alph:c {g__UWMad_Programming_API_#1_int}
566 }
567 \DeclareDocumentCommand \CounterToAlpha { m } {
568   \int_to_alph:c {g__UWMad_Programming_API_#1_int}
569 }
570 \DeclareDocumentCommand \CounterToROMAN { m } {
571   \int_to_Roman:c {g__UWMad_Programming_API_#1_int}
572 }
573 \DeclareDocumentCommand \CounterToRoman { m } {
574   \int_to_roman:c {g__UWMad_Programming_API_#1_int}
575 }

```

Module 3

Layout And Styles

```

576 \geometry{
577     includehead = true,
578     margin      = 1.0in,
579     paper       = letterpaper,
580 }
581 %
582 \creflabelformat{equation}{#2#1#3}
583 %
584 \captionsetup [table] {
585     format      = hang                ,
586     labelsep    = colon                ,
587     justification = justified          ,
588     labelfont   = sc                   ,
589     textfont    = sl                   ,
590     font        = {normal,stretch=1.1},
591     width       = 0.9\textwidth        ,
592     position    = above                ,
593     skip        = 0.50em
594 }
595 %
596 \captionsetup [figure] {
597     format      = hang                ,
598     labelsep    = colon                ,
599     justification = justified          ,
600     labelfont   = sc                   ,
601     textfont    = sl                   ,
602     font        = {normal,stretch=1.1},
603     width       = 0.9\textwidth        ,
604     position    = above                ,
605     skip        = 0.5em
606 }
607 %
608 \definecolor{UWMadGreen}{rgb}{0,0.7,0}
609 \hypersetup {
610     colorlinks      = true            ,
611     linkcolor       = blue             ,
612     citecolor       = UWMadGreen      ,
613     urlcolor        = violet          ,
614     pdfdisplaydoctitle = true          ,
615     pdfview          = {FitH}          ,
616     pdfstartview     = {FitH}          ,
617     pdfpagelayout    = OneColumn      ,
618     plainpages       = false           ,
619     hypertexnames    = true           ,
620     bookmarksopenlevel = 1            ,

```

```

621     bookmarksopen      = true      ,
622     unicode             = true
623 }
624 %
625 %
626 \doublespacing
627 \UWMad_Hook_Prepended:Nn \singlespacing {
628     \_UWMad_FrontMatter_StyleWarning:n {
629         University~guidelines~require~double~spacing.~
630         If~this~is~for~temporary~use,~please~use~the~spacing~environment.
631     }
632 }
633 \UWMad_Hook_Prepended:Nn \onehalfspacing {
634     \_UWMad_FrontMatter_StyleWarning:n {
635         University~guidelines~require~double~spacing.~
636         If~this~is~for~temporary~use,~please~use~the~spacing~environment.
637     }
638 }
639 %
640 %
641 %
642 %   Standard \LaTeX{} styles and lengths
643 \pagestyle{myheadings}
644 \setlength{\parindent}{0pt}
645 \setlength{\parskip}{10pt}
646 \setlength{\headsep}{15pt}

```

Module 4

Sectioning

Prefix some code such that `\chapter` has the page number in the upper right-hand corner and ensures that the page numbering is arabic before the first unnumbered chapter is used.

```

647 \UWMad_Hook_Prepnd:Nn \@chapter {
648   \thispagestyle{myheadings}
649   \int_compare:nNnTF {\value{chapter}} = {0} {
650     \pagenumbering{arabic}
651   } { }
652 }
653 \UWMad_Hook_Prepnd:Nn \@schapter {
654   \thispagestyle{myheadings}
655 }

```

4.1 Appendix

Here the `\appendix` command is redefined to act like the `\chapter` command. Before, `\appendix` simply changed the `chaptername` to “Appendix”.

Define the appendix counter.

```

656 \int_new:N \g__UWMad_Appendix_Counter_int
657 \int_set:Nn \g__UWMad_Appendix_Counter_int {0}

```

This command is used when the first `\appendix` command is used. It sets the `chaptername` to “Appendix” and sets the `\thechapter` to use the appendix counter above.

```

658 \cs_new:Nn \__UWMad_Appendix_Initialize:{
659   \par
660   \setcounter{section}{0}
661   \cs_gset_eq:NN \@chapapp \appendixname
662   \cs_gset:Npn \thechapter {
663     \int_to_Alph:n {
664       \g__UWMad_Appendix_Counter_int
665     }
666   }
667 }

```

Now, `\appendix` is undefined (to avoid a warning from `xparse`) and redefined with standard $\LaTeX 2_{\epsilon}$ sectioning arguments.

```

668 \cs_undefine:N \appendix
669 \DeclareDocumentCommand \appendix { s o m } {
670
671     \int_compare:nNnTF {\g__UWMad_Appendix_Counter_int} = {0} {
672         \__UWMad_Appendix_Initialize:
673     } { }
674     \int_gincr:N \g__UWMad_Appendix_Counter_int
675
676     \IfBooleanTF { #1 } {
677         \chapter*{#3}
678     } {
679         \IfNoValueTF { #2 } {
680             \chapter[#3]{#3}
681         } {
682             \chapter[#2]{#3}
683         }
684     }
685 }
```

4.2 Front Matter

Front Matter commands (sometimes called preliminary pages) are defined here. These are the sections of the document the precede the main body of the work.

Initialize a counter for the FrontMatter.

```

686 %
687 \int_new:N \g__UWMad_FrontMatter_Counter_int
```

This command enters the Front Matter with a given name and section level into the Table of Contents.

```

688 \cs_new:Nn \__UWMad_FrontMatter_Register:nm {
689
690     \int_compare:nNnTF {\g__UWMad_FrontMatter_Counter_int} = {0} {
691         \pagenumbering{roman}
692     } { }
693
694     \int_gincr:N \g__UWMad_FrontMatter_Counter_int
695     \addcontentsline
696         {toc}
697         {#1}
698         {#2}
```

```
699 }
```

These variables hold the default names of the Front Matter sections.

```
700 \tl_new:N \g__UWMad_FrontMatter_Title_Dedications_tl
701 \tl_new:N \g__UWMad_FrontMatter_Title_Acknowledgments_tl
702 \tl_new:N \g__UWMad_FrontMatter_Title_Abstract_tl
703 \tl_new:N \g__UWMad_FrontMatter_Title_UMIAbstract_tl
704 \tl_new:N \g__UWMad_FrontMatter_Title_Preface_tl
705 %
706 \tl_gset:Nn \g__UWMad_FrontMatter_Title_Dedications_tl
707   {Dedications}
708 \tl_gset:Nn \g__UWMad_FrontMatter_Title_Acknowledgments_tl
709   {Acknowledgments}
710 \tl_gset:Nn \g__UWMad_FrontMatter_Title_Abstract_tl
711   {Abstract}
712 \tl_gset:Nn \g__UWMad_FrontMatter_Title_UMIAbstract_tl
713   {Abstract}
714 \tl_gset:Nn \g__UWMad_FrontMatter_Title_Preface_tl
715   {Preface}
```

First the `abstract` environment from the L^AT_EX base class is undefined, and the Front Matter commands as described in the User Guide are defined.

```
716 \cs_undefine:N \abstract
717 \cs_undefine:N \endabstract
718
719 \DeclareDocumentCommand \FrontMatterSetSection { m m } {
720
721   \tl_set_eq:Nc
722     \l_tmpa_tl
723     {g__UWMad_FrontMatter_Title_#2_tl}
724
725   \IfNoValueTF { #1 } { } {
726     \IfEmptyTF { #1 } { } {
727       \tl_set:Nn \l_tmpa_tl {#1}
728     }
729   }
730
731   \chapter*{\l_tmpa_tl}
732   \__UWMad_FrontMatter_Register:nn {chapter} {
733     \l_tmpa_tl
734   }
735
736 }
737 \DeclareDocumentCommand \dedications { g } {
738   \FrontMatterSetSection{#1}{Dedications}
739 }
740 \DeclareDocumentCommand \acknowledgments { g } {
741   \FrontMatterSetSection{#1}{Acknowledgments}
742 }
743 \DeclareDocumentCommand \abstract { g } {
744   \FrontMatterSetSection{#1}{Abstract}
```

```

745 }
746 \DeclareDocumentCommand \umiabstract { g } {
747   \FrontMatterSetSection{#1}{Abstract}
748 }
749 \DeclareDocumentCommand \preface { g } {
750   \FrontMatterSetSection{#1}{Preface}
751 }

```

4.3 TOC Tweaks

This section tweaks the Table of Contents, the List of Tables, and the List of Figures commands to insert them into the bookmark tree of the PDF. Also, the commands for changing the titles used for each of the commands' associated sections are given.

First, store the original commands and then undefine them.

```

752 \cs_gset_eq:NN \TableOfContentsDefault \tableofcontents
753 \cs_gset_eq:NN \ListOfTablesDefault \listoftables
754 \cs_gset_eq:NN \ListOfFiguresDefault \listoffigures
755 \cs_undefine:N \tableofcontents
756 \cs_undefine:N \listoftables
757 \cs_undefine:N \listoffigures

```

Now create token list variables to store the titles of the sections and assign defaults.

```

758 \tl_new:N \g__UWMad_TOC_Name_TOC_tl
759 \tl_new:N \g__UWMad_TOC_Name_LOT_tl
760 \tl_new:N \g__UWMad_TOC_Name_LOF_tl
761 \tl_gset:Nn \g__UWMad_TOC_Name_TOC_tl {Table~of~Contents}
762 \tl_gset:Nn \g__UWMad_TOC_Name_LOT_tl {List~of~Tables}
763 \tl_gset:Nn \g__UWMad_TOC_Name_LOF_tl {List~of~Figures}

```

Define the new user-level commands. Since these commands are technically Front Matter, they are registered as such.

```

764 \DeclareDocumentCommand \tableofcontents { } {
765
766   \tl_gset_eq:NN \contentsname \g__UWMad_TOC_Name_TOC_tl
767
768   \group_begin:
769     \setstretch{1.05}
770     \phantomsection
771     \ExplSyntaxOff
772     \TableOfContentsDefault
773     \ExplSyntaxOn
774     \__UWMad_FrontMatter_Register:nn

```

```

775         {chapter}
776         {\contentsname}
777     \clearpage
778 \group_end:
779 }
780 \DeclareDocumentCommand \listoftables { } {
781
782     \cs_set_eq:NN \listtablename \g__UWMad_TOC_Name_LOT_tl
783
784     \group_begin:
785         \setstretch{1.05}
786         \ExplSyntaxOff
787         \ListOfTablesDefault
788         \ExplSyntaxOn
789         \__UWMad_FrontMatter_Register:nn
790             {chapter}
791             {\listtablename}
792     \clearpage
793 \group_end:
794 }
795 \DeclareDocumentCommand \listoffigures { } {
796
797     \cs_set_eq:NN \listfigurename \g__UWMad_TOC_Name_LOF_tl
798
799     \group_begin:
800         \setstretch{1.05}
801         \ExplSyntaxOff
802         \ListOfFiguresDefault
803         \ExplSyntaxOn
804         \__UWMad_FrontMatter_Register:nn
805             {chapter}
806             {\listfigurename}
807     \clearpage
808 \group_end:
809 }

```

Camel-cased aliases.

```

810 \cs_set_eq:NN \TableOfContents \tableofcontents
811 \cs_set_eq:NN \ListOfTables \listoftables
812 \cs_set_eq:NN \ListOfFigures \listoffigures

```

User-level commands to change the default names.

```

813 \DeclareDocumentCommand \TableOfContentsName { m } {
814     \tl_gset:Nn \g__UWMad_TOC_Name_TOC_tl {#1}
815 }
816 \DeclareDocumentCommand \ListOfTablesName { m } {
817     \tl_gset:Nn \g__UWMad_TOC_Name_LOT_tl {#1}
818 }
819 \DeclareDocumentCommand \ListOfFiguresName { m } {
820     \tl_gset:Nn \g__UWMad_TOC_Name_LOF_tl {#1}
821 }

```


4.4 Section-Level Commands

These commands are used internally when needing to check if a user-supplied `section` is a $\text{\LaTeX} 2_{\varepsilon}$ -defined section and also easily acquired/use the relationships among section levels when needed.

These variables map a `section` to a level number and also serve to define the existence of the level.

```

822 \tl_const:Nn \c__UWMad_SectionsLevel_part_tl      {-1}
823 \tl_const:Nn \c__UWMad_SectionsLevel_chapter_tl   {0}
824 \tl_const:Nn \c__UWMad_SectionsLevel_section_tl   {1}
825 \tl_const:Nn \c__UWMad_SectionsLevel_subsection_tl {2}
826 \tl_const:Nn \c__UWMad_SectionsLevel_subsubsection_tl {3}
827 \tl_const:Nn \c__UWMad_SectionsLevel_paragraph_tl {4}
828 \tl_const:Nn \c__UWMad_SectionsLevel_subparagraph_tl {5}

```

Define a message to warn about an undefined section and associated command to check if a section exists.

```

829 \msg_new:nnn { UWMadThesis } { Sectioning / UndefinedSection } {
830   Undefined~section~'#1'~used.
831 }
832 \cs_new:Nn \UWMad_IfSectionExists:nT {
833   \tl_if_exist:cTF {c__UWMad_SectionsLevel_ #1 _tl} {
834     #2
835   } {
836     \msg_error:nnn
837       { UWMadThesis }
838       { Sectioning / UndefinedSection }
839       {#1}
840   }
841 }

```

Variables that map a level number to a section.

```

842 \tl_const:cn {c__UWMad_LevelsSection_-1_tl} {part}
843 \tl_const:cn {c__UWMad_LevelsSection_ 0_tl} {chapter}
844 \tl_const:cn {c__UWMad_LevelsSection_ 1_tl} {section}
845 \tl_const:cn {c__UWMad_LevelsSection_ 2_tl} {subsection}
846 \tl_const:cn {c__UWMad_LevelsSection_ 3_tl} {subsubsection}
847 \tl_const:cn {c__UWMad_LevelsSection_ 4_tl} {paragraph}
848 \tl_const:cn {c__UWMad_LevelsSection_ 5_tl} {subparagraph}

```

Variables that map a section to it's next lower one.

```

849 \tl_const:Nn \c__UWMad_NextSection_part_tl      {chapter}

```

```

850 \tl_const:Nn \c__UWMad_NextSection_chapter_tl      {section}
851 \tl_const:Nn \c__UWMad_NextSection_section_tl      {subsection}
852 \tl_const:Nn \c__UWMad_NextSection_subsection_tl   {subsubsection}
853 \tl_const:Nn \c__UWMad_NextSection_subsubsection_tl {paragraph}
854 \tl_const:Nn \c__UWMad_NextSection_paragraph_tl    {subparagraph}

```

Variables that map a section to it's next higher one.

```

855 \tl_const:Nn \c__UWMad_PreviousSection_chapter_tl  {part}
856 \tl_const:Nn \c__UWMad_PreviousSection_section_tl  {chapter}
857 \tl_const:Nn \c__UWMad_PreviousSection_subsection_tl {section}
858 \tl_const:Nn \c__UWMad_PreviousSection_subsubsection_tl {subsection}
859 \tl_const:Nn \c__UWMad_PreviousSection_paragraph_tl {subsubsection}
860 \tl_const:Nn \c__UWMad_PreviousSection_subparagraph_tl {paragraph}

```

Given a section, acquire its level number.

```

861 \cs_new:Nn \UWMad_SectionToLevel:n {
862   \UWMad_IfSectionExists:nT {#1} {
863     \tl_use:c {c__UWMad_SectionsLevel_ #1 _tl}
864   }
865 }

```

Given a level number, acquire its section.

```

866 \cs_new:Nn \UWMad_LevelToSection:n {
867   \UWMad_IfSectionExists:nT {#1} {
868     \tl_use:c {c__UWMad_LevelsSection_ #1 _tl}
869   }
870 }

```

Given a section, acquire its next lower one.

```

871 \cs_new:Nn \UWMad_NextSection:n {
872   \UWMad_IfSectionExists:nT {#1} {
873     \tl_use:c {c__UWMad_NextSection_ #1 _tl}
874   }
875 }

```

Given a section, acquire its next higher one.

```

876 \cs_new:Nn \UWMad_PreviousSection:n {
877   \UWMad_IfSectionExists:nT {#1} {
878     \tl_use:c {c__UWMad_PreviousSection_ #1 _tl}
879   }
880 }

```

Module 5

Math

```

881 %
882 %
883 %
884 \tex_everydisplay:D \exp_after:wN {
885     \tex_the:D \tex_everydisplay:D
886     \cs_set_eq:NN \frac \dfrac
887 }
888 %
889 %
890 %
891 %
892 %
893 % ===== %
894 %                               Derivative Commands %
895 % ===== %
896 %
897 \tl_new:N    \g_UWMad_Math_derivSymbol_tl
898 \tl_gset:Nn \g_UWMad_Math_derivSymbol_tl {\mathrm{d}}
899 \tl_new:N    \g_UWMad_Math_pderivSymbol_tl
900 \tl_gset:Nn \g_UWMad_Math_pderivSymbol_tl {\partial}
901 \tl_new:N    \g_UWMad_Math_tderivSymbol_tl
902 \tl_gset:Nn \g_UWMad_Math_tderivSymbol_tl {\mathrm{D}}
903 \tl_new:N    \g_UWMad_Math_DelimiterDefaultLeft_tl
904 \tl_gset:Nn \g_UWMad_Math_DelimiterDefaultLeft_tl {}
905 \tl_new:N    \g_UWMad_Math_DelimiterDefaultRight_tl
906 \tl_gset:Nn \g_UWMad_Math_DelimiterDefaultRight_tl {}
907 \tl_new:N    \l_UWMad_Math_DelimiterLeft_tl
908 \tl_new:N    \l_UWMad_Math_DelimiterRight_tl
909 %
910 %
911 \DeclareDocumentCommand \derivSymbol { } {
912     \g_UWMad_Math_derivSymbol_tl
913 }
914 \DeclareDocumentCommand \pderivSymbol { } {
915     \g_UWMad_Math_pderivSymbol_tl
916 }
917 \DeclareDocumentCommand \tderivSymbol { } {
918     \g_UWMad_Math_tderivSymbol_tl
919 }
920 %
921 %
922 \DeclareDocumentCommand \derivSymbolChange { m } {
923     \tl_set:Nn \g_UWMad_Math_derivSymbol_tl {#1}
924 }
925 \DeclareDocumentCommand \pderivSymbolChange { m } {

```

```

926 \tl_set:Nn \g_UWMad_Math_pderivSymbol_tl {#1}
927 }
928 \DeclareDocumentCommand \tderivSymbolChange { m } {
929 \tl_set:Nn \g_UWMad_Math_tderivSymbol_tl {#1}
930 }
931 %
932 %
933 \DeclareDocumentCommand \derivSymbolChangeDefault { m } {
934 \tl_gset:Nn \g_UWMad_Math_derivSymbol_tl {#1}
935 }
936 \DeclareDocumentCommand \pderivSymbolChangeDefault { m } {
937 \tl_gset:Nn \g_UWMad_Math_pderivSymbol_tl {#1}
938 }
939 \DeclareDocumentCommand \tderivSymbolChangeDefault { m } {
940 \tl_gset:Nn \g_UWMad_Math_tderivSymbol_tl {#1}
941 }
942 %
943 %
944 \DeclareDocumentCommand \DelimiterChangeDefault { m m } {
945 \tl_gset:Nn \g_UWMad_Math_DelimiterDefaultLeft_tl {#1}
946 \tl_gset:Nn \g_UWMad_Math_DelimiterDefaultRight_tl {#2}
947 }
948 %
949 %
950 \DeclareDocumentCommand \DerivativeGeneral { +m +m m m } {
951 \frac{ #4^{#3} #1 }
952 { #4 #2^{#3} }
953 }
954 \DeclareDocumentCommand \DerivativeGeneralBig { +m +m m m m m } {
955
956 \IfNoValueTF {#5} {
957 \tl_set_eq:NN
958 \l_UWMad_Math_DelimiterLeft_tl
959 \g_UWMad_Math_DelimiterDefaultLeft_tl
960 } {
961 \tl_set:Nn \l_UWMad_Math_DelimiterLeft_tl {#5}
962 }
963
964 \IfNoValueTF {#6} {
965 \tl_set_eq:NN
966 \l_UWMad_Math_DelimiterRight_tl
967 \g_UWMad_Math_DelimiterDefaultRight_tl
968 } {
969 \tl_set:Nn \l_UWMad_Math_DelimiterRight_tl {#6}
970 }
971
972 \frac{ #4^{#3} }
973 { #4 #2^{#3} }
974 \!\!
975 \left\l_UWMad_Math_DelimiterLeft_tl
976 #1
977 \right\l_UWMad_Math_DelimiterRight_tl
978 }
979 %
980 %
981 \DeclareDocumentCommand \deriv { +m +m G{} } {

```

```

982 \DerivativeGeneral
983   {#1}{#2}{#3}{\derivSymbol}
984 }
985 \DeclareDocumentCommand \pderiv { +m +m G{} } {
986   \DerivativeGeneral
987   {#1}{#2}{#3}{\pderivSymbol}
988 }
989 \DeclareDocumentCommand \tderiv { +m +m G{} } {
990   \DerivativeGeneral
991   {#1}{#2}{#3}{\tderivSymbol}
992 }
993 %
994 %
995 \DeclareDocumentCommand \derivbig { o +m o +m G{} } {
996   \DerivativeGeneralBig
997   {#2}{#4}{#5}{\derivSymbol}{#1}{#3}
998 }
999 \DeclareDocumentCommand \pderivbig { o +m o +m G{} } {
1000   \DerivativeGeneralBig
1001   {#2}{#4}{#5}{\pderivSymbol}{#1}{#3}
1002 }
1003 \DeclareDocumentCommand \tderivbig { o +m o +m G{} } {
1004   \DerivativeGeneralBig
1005   {#2}{#4}{#5}{\tderivSymbol}{#1}{#3}
1006 }
1007 %
1008 %
1009 %
1010 %
1011 \DeclareMathOperator*\{Sup} {Sup}
1012 \DeclareMathOperator*\{Inf} {Inf}
1013 \DeclareMathOperator*\{Lim} {Lim}
1014 \DeclareMathOperator*\{Min} {Min}
1015 \DeclareMathOperator*\{Max} {Max}
1016 \DeclareMathOperator*\{ArgMin} {ArgMin}
1017 \DeclareMathOperator*\{ArgMax} {ArgMax}
1018 \DeclareMathOperator*\{Abs} {Abs}
1019 \DeclareMathOperator*\{Ln} {Ln}
1020 \DeclareMathOperator*\{Log} {Log}
1021 \DeclareMathOperator*\{Exp} {Exp}
1022 \DeclareMathOperator*\{Cos} {Cos}
1023 \DeclareMathOperator*\{Sin} {Sin}
1024 \DeclareMathOperator*\{Tan} {Tan}
1025 \DeclareMathOperator*\{Sec} {Sec}
1026 \DeclareMathOperator*\{Csc} {Csc}
1027 \DeclareMathOperator*\{Cot} {Cot}
1028 \DeclareMathOperator*\{Cosh} {Cosh}
1029 \DeclareMathOperator*\{Sinh} {Sinh}
1030 \DeclareMathOperator*\{Tanh} {Tanh}
1031 \DeclareMathOperator*\{Sech} {Sech}
1032 \DeclareMathOperator*\{Csch} {Csch}
1033 \DeclareMathOperator*\{Coth} {Coth}
1034 \DeclareMathOperator*\{ArcCos} {ArcCos}
1035 \DeclareMathOperator*\{ArcSin} {ArcSin}
1036 \DeclareMathOperator*\{ArcTan} {ArcTan}
1037 \DeclareMathOperator*\{ArcSec} {ArcSec}

```

```

1038 \DeclareMathOperator{\ArcCsc} {\ArcCsc}
1039 \DeclareMathOperator{\ArcCot} {\ArcCot}
1040 \DeclareMathOperator{\ArcCosh} {\ArcCosh}
1041 \DeclareMathOperator{\ArcSinh} {\ArcSinh}
1042 \DeclareMathOperator{\ArcTanh} {\ArcTanh}
1043 \DeclareMathOperator{\ArcSech} {\ArcSech}
1044 \DeclareMathOperator{\ArcCsch} {\ArcCsch}
1045 \DeclareMathOperator{\ArcCoth} {\ArcCoth}
1046 %
1047 %
1048 %
1049 %
1050 %
1051 % ===== %
1052 % Miscellaneous Commands %
1053 % ===== %
1054 \cs_new:Nn \UWMad_Math_RootWithTail:nn {
1055
1056   \hbox_set:Nn \l_tmpa_box {
1057     $
1058       \mathchoice
1059         {\root #1 \of {#2\:\!}}
1060         {\root #1 \of {#2\:\!}}
1061         {\root #1 \of {#2\:\!}}
1062         {\root #1 \of {#2\:\!}}
1063     $
1064   }
1065   %
1066   \dim_set:Nn \l_tmpa_dim {\box_ht:N \l_tmpa_box}
1067   \dim_set:Nn \l_tmpb_dim {0.8\l_tmpa_dim}
1068   %
1069   \hbox_set:Nn \l_tmpb_box {
1070     \tex_vrule:D height \l_tmpa_dim depth -\l_tmpb_dim
1071   }
1072   %
1073   \box_use:N \l_tmpa_box
1074   \box_move_down:nn {0.40pt}{\box_use:N \l_tmpb_box}
1075 }
1076 \DeclareDocumentCommand \Sqrt { 0{} m } {
1077   \UWMad_Math_RootWithTail:nn{#1}{#2}
1078 }
1079 %
1080 %
1081 \DeclareExpandableDocumentCommand \IfMathModeTF { +m +m } {
1082   \mode_if_math:TF {
1083     #1
1084   }{
1085     $#2$
1086   }
1087 }
1088 \cs_gset_eq:NN \supsipa \sups
1089 \cs_undefine:N \sups
1090 \ExplSyntaxOff
1091 \DeclareDocumentCommand \subs { 0{} +m } {%
1092   \IfMathModeTF{%
1093     _{\!\!\:\!#1\text{\scriptsize #2}}}%

```

```

1094     }{%
1095     _{\!#1\text{\scriptsize #2}}}%
1096     }%
1097 }%
1098 \DeclareDocumentCommand \supsub { 0{} +m } {%
1099     \IfMathModeTF{%
1100         ^{#1\text{\scriptsize #2}}}%
1101     }{%
1102         ^{#1\text{\scriptsize #2}}}%
1103     }%
1104 }%
1105 \DeclareDocumentCommand \subsupsub { 0{} +m 0{} +m } {%
1106     \IfMathModeTF{%
1107         _{#1\text{\scriptsize #2}}^{!\!#3\text{\scriptsize #4}}}%
1108     }{%
1109         _{#1\text{\scriptsize #2}}^{!\!#3\text{\scriptsize #4}}}%
1110     }%
1111 }%
1112 \ExplSyntaxOn
1113 %
1114 \DeclareDocumentCommand \OneOver { +m } {
1115     \frac{1}{#1}
1116 }
1117 \DeclareDocumentCommand \oneo { +m } {
1118     \OneOver{#1}
1119 }
1120 \DeclareDocumentCommand \dd { m } {
1121     \mathrm{d}{#1}
1122 }
1123 \DeclareDocumentCommand \dprime { } {
1124     {\prime\prime}
1125 }
1126 \DeclareDocumentCommand \tprime { } {
1127     {\prime\prime\prime}
1128 }
1129 \DeclareDocumentCommand \LessThan { } {<}
1130 \DeclareDocumentCommand \GreaterThan { } {>}
1131 %

```

Module 6

ListOf

The ListOf Module is a collection of commands that enables the easy creation and typesetting of Lists.

Lists are taken to be any collection of entries that is to be typeset with a particular style. For example, a simple Nomenclature could be considered a list of (symbol, description) entries to be typeset with a fixed style for all entries. The ListOf commands create a system specifically for this scenario.

Of course, as the commands description will show, lists can be much more complicated than two items. For the ListOf system to function, an author really only needs to define the ListOf, create a command to push (enqueue) entries on to the ListOf queue, and at some point tell the ListOf to typeset the entries it has stored (if display of the content is desired).

`\UWMad_ListOf_Define:n` Define a new ListOf with $\{\langle ID \rangle\}$. This command creates the commands to store the section commands and title for each group, the booleans to indicate if the sections should be numbered and if the sections should be included in the table of contents (regardless of numbering), a hash to hold of the user-defined hooks for the ListOf, and a queue to store the entries for typesetting.

```

1132 \cs_new:Nn \UWMad_ListOf_Define:n {
1133   \tl_const:cn {c__UWMad_ListOf#1_IsDefined_tl}{ }
1134   %
1135   \tl_new:c {g__UWMad_ListOf#1_Section_Main_tl}
1136   \tl_new:c {g__UWMad_ListOf#1_Section_Group_tl}
1137   \tl_new:c {g__UWMad_ListOf#1_Section_Subgroup_tl}
1138   %
1139   \tl_new:c {g__UWMad_ListOf#1_Title_Main_tl}
1140   \tl_new:c {g__UWMad_ListOf#1_Title_Group_tl}
1141   \tl_new:c {g__UWMad_ListOf#1_Title_Subgroup_tl}
1142   %
1143   \bool_new:c {g__UWMad_ListOf#1_ClearAfterPrint_bool}
1144   \bool_gset_true:c {g__UWMad_ListOf#1_ClearAfterPrint_bool}
1145   \bool_new:c {g__UWMad_ListOf#1_IsNumbered_bool}
1146   \bool_gset_true:c {g__UWMad_ListOf#1_IsNumbered_bool}
1147   \bool_new:c {g__UWMad_ListOf#1_IncludeInTOC_bool}
1148   \bool_gset_true:c {g__UWMad_ListOf#1_IncludeInTOC_bool}
1149   \UWMad_Queue_Define:n {g__ListOf#1_EntryQueue}
1150   \UWMad_Hash_Define:n {g__ListOf#1_Hook}
1151 }
```

`\UWMad_ListOf_Delete:n` Simply undefines all of the commands created in the Define command above for the given $\{\langle ID \rangle\}$.


```

1152 \cs_new:Nn \UWMad_ListOf_Delete:n {
1153   \cs_undefine:c {c__UWMad_ListOf#1_IsDefined_tl}
1154 %
1155   \cs_undefine:c {g__UWMad_ListOf#1_Section_Main_tl}
1156   \cs_undefine:c {g__UWMad_ListOf#1_Section_Group_tl}
1157   \cs_undefine:c {g__UWMad_ListOf#1_Section_Subgroup_tl}
1158 %
1159   \cs_undefine:c {g__UWMad_ListOf#1_Title_Main_tl}
1160   \cs_undefine:c {g__UWMad_ListOf#1_Title_Group_tl}
1161   \cs_undefine:c {g__UWMad_ListOf#1_Title_Subgroup_tl}
1162 %
1163   \cs_undefine:c {g__UWMad_ListOf#1_ClearAfterPrint_bool}
1164   \cs_undefine:c {g__UWMad_ListOf#1_IsNumbered_bool}
1165   \cs_undefine:c {g__UWMad_ListOf#1_IncludeInTOC_bool}
1166   \UWMad_Queue_Delete:n {g__ListOf#1_EntryQueue}
1167   \UWMad_Hash_Delete:n {g__ListOf#1_Hook}
1168 }

```

`\UWMad_ListOf_IfDefined:nT` Checks to see if a ListOf with $\{\langle ID \rangle\}$ has been created and errors if not.

```

1169 \cs_new:Nn \UWMad_ListOf_IfDefined:nT {
1170   \__UWMad_IfDefined:nnnnT
1171   {c__UWMad_ListOf}
1172   {#1}
1173   {_IsDefined_tl}
1174   {ListOf}
1175   {#2}
1176 }

```

`\UWMad_ListOf_MakeNumbered:n` Makes the current section of the ListOf with $\{\langle ID \rangle\}$ numbered or unnumbered (i.e., a star version).

`\UWMad_ListOf_MakeNotNumbered:n`

```

1177 \cs_new:Nn \UWMad_ListOf_MakeNumbered:n {
1178   \UWMad_ListOf_IfDefined:nT {#1} {
1179     \bool_set_true:c {g__UWMad_ListOf#1_IsNumbered_bool}
1180   }
1181 }
1182 \cs_new:Nn \UWMad_ListOf_MakeNotNumbered:n {
1183   \UWMad_ListOf_IfDefined:nT {#1} {
1184     \bool_set_false:c {g__UWMad_ListOf#1_IsNumbered_bool}
1185   }
1186 }

```

`\UWMad_ListOf_IfNumbered:nTF` Branches to $\{\langle True Code \rangle\}$ or $\{\langle False Code \rangle\}$ depending on whether the ListOf with $\{\langle ID \rangle\}$ is numbered or not.

```

1187 \cs_new:Nn \UWMad_ListOf_IfNumbered:nTF {
1188   \UWMad_ListOf_IfDefined:nT {#1} {
1189     \bool_if:cTF {g__UWMad_ListOf#1_IsNumbered_bool} {
1190       #2

```

```

1191         }{
1192         #3
1193     }
1194 }
1195 }

```

`\UWMad_ListOf_IncludeInTOC:n` Makes the current section of the `ListOf` with `{\langle ID \rangle}` appear in the Table of Contents (TOC) or not, regardless of if it is numbered/unnumbered.

`\UWMad_ListOf_DoNotIncludeInTOC:n`

```

1196 \cs_new:Nn \UWMad_ListOf_IncludeInTOC:n {
1197     \UWMad_ListOf_IfDefined:nT {#1} {
1198         \bool_set_true:c {c__UWMad_ListOf#1_IncludeInTOC_bool}
1199     }
1200 }
1201 \cs_new:Nn \UWMad_ListOf_DoNotIncludeInTOC:n {
1202     \UWMad_ListOf_IfDefined:nT {#1} {
1203         \bool_set_false:c {c__UWMad_ListOf#1_IncludeInTOC_bool}
1204     }
1205 }

```

`\UWMad_ListOf_IfIncludeInTOC:n` Branches to `{\langle True Code \rangle}` or `{\langle False Code \rangle}` depending on whether the `ListOf` with `{\langle ID \rangle}` is to be included or not.

```

1206 \cs_new:Nn \UWMad_ListOf_IfIncludeInTOC:nTF {
1207     \UWMad_ListOf_IfDefined:nT {#1} {
1208         \bool_if:cTF {c__UWMad_ListOf#1_IncludeInTOC_bool} {
1209             #2
1210         }{
1211             #3
1212         }
1213     }
1214 }

```

\UWMad_ListOf_SetTitle_Main:nn	\UWMad_ListOf_SetTitle_Main:nn{<ID>}{<Title>}
\UWMad_ListOf_SetTitle_Group:nn	\UWMad_ListOf_SetTitle_Group:nn{<ID>}{<Title>}
\UWMad_ListOf_SetTitle_Subgroup:nn	\UWMad_ListOf_SetTitle_Subgroup:nn{<ID>}{<Title>}

Sets the value of the title of the sections to {<Title>} for the ListOf with {<ID>}

```

1215 \cs_new:Nn \UWMad_ListOf_SetTitle_Main:nn {
1216     \UWMad_ListOf_IfDefined:nT {#1} {
1217         \tl_set:cn {g__UWMad_ListOf#1_Title_Main_tl}{#2}
1218     }
1219 }
1220 \cs_new:Nn \UWMad_ListOf_SetTitle_Group:nn {
1221     \UWMad_ListOf_IfDefined:nT {#1} {
1222         \tl_set:cn {g__UWMad_ListOf#1_Title_Group_tl}{#2}
1223     }
1224 }
1225 \cs_new:Nn \UWMad_ListOf_SetTitle_Subgroup:nn {
1226     \UWMad_ListOf_IfDefined:nT {#1} {
1227         \tl_set:cn {g__UWMad_ListOf#1_Title_Subgroup_tl}{#2}
1228     }
1229 }

```

\UWMad_ListOf_GetTitle_Main:nn	\UWMad_ListOf_GetTitle_Main:n {<ID>}
\UWMad_ListOf_GetTitle_Group:nn	\UWMad_ListOf_GetTitle_Group:n {<ID>}
\UWMad_ListOf_GetTitle_Subgroup:nn	\UWMad_ListOf_GetTitle_Subgroup:n{<ID>}

Retrieves the value of the title of the section for the ListOf with {<ID>}.

```

1230 \cs_new:Nn \UWMad_ListOf_GetTitle_Main:n {
1231     \UWMad_ListOf_IfDefined:nT {#1} {
1232         \tl_use:c {g__UWMad_ListOf#1_Title_Main_tl}
1233     }
1234 }
1235 \cs_new:Nn \UWMad_ListOf_GetTitle_Group:n {
1236     \UWMad_ListOf_IfDefined:nT {#1} {
1237         \tl_use:c {g__UWMad_ListOf#1_Title_Group_tl}
1238     }
1239 }
1240 \cs_new:Nn \UWMad_ListOf_GetTitle_Subgroup:n {
1241     \UWMad_ListOf_IfDefined:nT {#1} {
1242         \tl_use:c {g__UWMad_ListOf#1_Title_Subgroup_tl}
1243     }
1244 }

```

<code>\UWMad_ListOf_SetSection_Main:nn</code>	<code>\UWMad_ListOf_SetSection_Main:nn{<ID>}{<Section>}</code>
<code>\UWMad_ListOf_SetSection_Group:nn</code>	<code>\UWMad_ListOf_SetSection_Group:nn{<ID>}{<Section>}</code>
<code>\UWMad_ListOf_SetSection_Subgroup:nn</code>	<code>\UWMad_ListOf_SetSection_Subgroup:nn{<ID>}{<Section>}</code>

Sets the value of the section level and (currently) the sectioning command for a particular group to `{<Section>}` of the `ListOf` with `{<ID>}`.

```

1245 \cs_new:Nn \UWMad_ListOf_SetSection_Main:nn {
1246     \UWMad_ListOf_IfDefined:nT {#1} {
1247         \tl_set:cn {g__UWMad_ListOf#1_Section_Main_tl}{#2}
1248     }
1249 }
1250 \cs_new:Nn \UWMad_ListOf_SetSection_Group:nn {
1251     \UWMad_ListOf_IfDefined:nT {#1} {
1252         \tl_set:cn {g__UWMad_ListOf#1_Section_Group_tl}{#2}
1253     }
1254 }
1255 \cs_new:Nn \UWMad_ListOf_SetSection_Subgroup:nn {
1256     \UWMad_ListOf_IfDefined:nT {#1} {
1257         \tl_set:cn {g__UWMad_ListOf#1_Section_Subgroup_tl}{#2}
1258     }
1259 }
```

<code>\UWMad_ListOf_GetSection_Main:n</code>	<code>\UWMad_ListOf_GetSection_Main:n{<ID>}</code>
<code>\UWMad_ListOf_GetSection_Group:n</code>	<code>\UWMad_ListOf_GetSection_Group:n{<ID>}</code>
<code>\UWMad_ListOf_GetSection_Subgroup:n</code>	<code>\UWMad_ListOf_GetSection_Subgroup:n{<ID>}</code>

Gets the value of the section level for a particular group of the `ListOf` with `{<ID>}`.

```

1260 \cs_new:Nn \UWMad_ListOf_GetSection_Main:n {
1261     \UWMad_ListOf_IfDefined:nT {#1} {
1262         \tl_use:c {g__UWMad_ListOf#1_Section_Main_tl}
1263     }
1264 }
1265 \cs_new:Nn \UWMad_ListOf_GetSection_Group:n {
1266     \UWMad_ListOf_IfDefined:nT {#1} {
1267         \tl_use:c {g__UWMad_ListOf#1_Section_Group_tl}
1268     }
1269 }
1270 \cs_new:Nn \UWMad_ListOf_GetSection_Subgroup:n {
1271     \UWMad_ListOf_IfDefined:nT {#1} {
1272         \tl_use:c {g__UWMad_ListOf#1_Section_Subgroup_tl}
1273     }
1274 }
```

```
\UWMad_ListOf_SetHook:nnn \UWMad_ListOf_SetHook:nnn{<ID>}{<Hook name>}{<Hook code>}
```

Sets {<Hook name>} to {<Hook code>} for the ListOf with {<ID>}. The current hooks used are: PrePush, PostPush, PrePrint, and PostPrint.

```
1275 \cs_new:Nn \UWMad_ListOf_SetHook:nnn {
1276     \UWMad_Hash_Set:nnn{g__ListOf#1_Hook}{#2}{#3}
1277 }
```

```
\UWMad_ListOf_PushEntry:nn \UWMad_ListOf_PushEntry:nn {<ID>}{<Entry>}
```

Pushes {<Entry>} on to the entry queue of the ListOf with {<ID>}.

```
1278 \cs_new:Nn \UWMad_ListOf_PushEntry:nn {
1279     \UWMad_Hash_Get:nn {g__ListOf#1_Hook}{PrePush}
1280     \UWMad_Queue_Push:nn {g__ListOf#1_EntryQueue}{#2}
1281     \UWMad_Hash_Get:nn {g__ListOf#1_Hook}{PostPush}
1282 }
```

```
UWMad_ListOf_PrintEntries:n \UWMad_ListOf_PrintEntries:n{<ID>}
```

Prints all entries currently in the ListOf queue with {<ID>} and clears the queue. The PrePrint and PostPrint hooks are also called here.

```
1283 \cs_new:Nn \UWMad_ListOf_PrintEntries:n {
1284     \UWMad_Hash_Get:nn {g__ListOf#1_Hook}{PrePrint}
1285     \UWMad_Queue_Walk:nn {g__ListOf#1_EntryQueue}{##1}
1286     \UWMad_Queue_Clear:n {g__ListOf#1_EntryQueue}
1287     \UWMad_Hash_Get:nn {g__ListOf#1_Hook}{PostPrint}
1288 }
```

`\UWMad_ListOf_PrintTitle:nn` `\UWMad_ListOf_PrintTitle:nn{<ID>}{<Group>}`

Prints the title for the `{<Group>}` of the `ListOf` with `{<ID>}` at the section indicated by its associated token list. Numbering and table of contents adding is done according to the current values of their respective booleans.

```

1289 \cs_new:Nn \__UWMad_ListOf_CurrentSectioningCommand:n {}
1290 \cs_new:Nn \UWMad_ListOf_PrintTitle:nn {
1291
1292     \cs_set_eq:Nc
1293     \__UWMad_ListOf_CurrentSectioningCommand:n
1294     {\tl_use:c{g__UWMad_ListOf#1_Section_#2_tl}}
1295
1296     \UWMad_ListOf_IfNumbered:nTF {#1} {
1297
1298         \tl_if_eq:nnTF {#2} {Main} {
1299             \UWMad_ListOf_IfIncludeInTOC:nTF {#1} { } {
1300                 \int_set_eq:NN \l_tmpa_int \c@tocdepth
1301                 \setcounter{tocdepth}{-1}
1302             }
1303         } {
1304             \int_set_eq:NN \l_tmpa_int \c@tocdepth
1305             \setcounter{tocdepth}{-1}
1306         }
1307
1308
1309         \__UWMad_ListOf_CurrentSectioningCommand:n
1310         {\tl_use:c {g__UWMad_ListOf#1_Title_#2_tl}}
1311
1312
1313         \tl_if_eq:nnTF #2 {Main} {
1314             \UWMad_ListOf_IfIncludeInTOC:nTF {#1} { } {
1315                 \setcounter{tocdepth}{\l_tmpa_int}
1316             }
1317         } {
1318             \setcounter{tocdepth}{\l_tmpa_int}
1319         }
1320
1321     } {
1322
1323         \cs_generate_variant:Nn \tl_if_eq:nnTF {onTF}
1324         \tl_set:Nn \l_tmpa_tl {Main}
1325
1326         \phantomsection
1327         \__UWMad_ListOf_CurrentSectioningCommand:n*
1328         {\tl_use:c {g__UWMad_ListOf#1_Title_#2_tl}}
1329         \tl_if_eq:onTF {#2} {Main} {
1330             \UWMad_ListOf_IfIncludeInTOC:nTF {#1} {
1331                 \addcontentsline
1332                 {toc}
1333                 {\tl_use:c {g__UWMad_ListOf#1_Section_#2_tl}}
1334                 {\tl_use:c {g__UWMad_ListOf#1_Title_#2_tl}}
1335             } { }
1336         } { }
1337     }
1338
1339 }
```

```
\UWMad_ListOf_StartGroup:nn \UWMad_ListOf_StartGroup:n{<ID>}{<Group>}
```

A shortcut command that prints the entires in the current queue and then starts the next section by printing the title.

```
1340 \cs_new:Nn \UWMad_ListOf_StartGroup:nn {
1341     \UWMad_ListOf_PrintEntries:n{#1}
1342     \UWMad_ListOf_PrintTitle:nn {#1}{#2}
1343 }
```

6.1 Nomenclature

Dimensions that are calculated are declared first.

```
1344 \dim_new:N \l__UWMad_Nomenclature_WidestSymbol_dim
1345 \dim_new:N \l__UWMad_Nomenclature_WidestUnit_dim
1346 \dim_new:N \l__UWMad_Nomenclature_Entry_WidthSymbol_dim
1347 \dim_new:N \l__UWMad_Nomenclature_Entry_WidthUnits_dim
1348 \dim_new:N \l__UWMad_Nomenclature_Entry_WidthDescription_dim
```

Then user-adjustable dimensions are declared.

```
1349 \dim_new:N \l__UWMad_Nomenclature_TitleSkip_dim
1350 \dim_new:N \l__UWMad_Nomenclature_PrintSkip_dim
1351 \dim_new:N \l__UWMad_Nomenclature_Entry_MarginLeft_dim
1352 \dim_new:N \l__UWMad_Nomenclature_Entry_MarginBottom_dim
1353 \dim_new:N \l__UWMad_Nomenclature_Entry_Padding_dim
```

The token lists that hold the section and title of the groups are declared

```
1354 \tl_new:N \l__UWMad_Nomenclature_Section_Main_tl
1355 \tl_new:N \l__UWMad_Nomenclature_Section_Group_tl
1356 \tl_new:N \l__UWMad_Nomenclature_Section_Subgroup_tl
1357 \tl_new:N \l__UWMad_Nomenclature_Title_Main_tl
1358 \tl_new:N \l__UWMad_Nomenclature_Title_Group_tl
1359 \tl_new:N \l__UWMad_Nomenclature_Title_Subgroup_tl
```

Now the keys for user-customization are defined:

```
1360 \keys_define:nn { UWMad/Nomenclature } {
```

Adjustable dimensions:

```
1361     title-skip .dim_set:N = \l__UWMad_Nomenclature_TitleSkip_dim,
```

```

1362 print-skip .dim_set:N = \l__UWMad_Nomenclature_PrintSkip_dim,
1363 entry-margin-left .dim_set:N =
1364     \l__UWMad_Nomenclature_Entry_MarginLeft_dim,
1365 entry-margin-bottom .dim_set:N =
1366     \l__UWMad_Nomenclature_Entry_MarginBottom_dim,
1367 entry-padding .dim_set:N =
1368     \l__UWMad_Nomenclature_Entry_Padding_dim,

```

Adjustable dimension defaults:

```

1369 title-skip .default:n = {0.00pt},
1370 print-skip .default:n = {1.00em},
1371 entry-margin-left .default:n = {1.00em},
1372 entry-margin-bottom .default:n = {0.25em},
1373 entry-padding .default:n = {0.75em},

```

Group section adjustments:

```

1374 main-section .code:n = {
1375     \tl_set:Nn
1376         \l__UWMad_Nomenclature_Section_Main_tl {#1}
1377 },
1378 group-section .code:n = {
1379     \tl_set:Nn
1380         \l__UWMad_Nomenclature_Section_Group_tl {#1}
1381 },
1382 subgroup-section .code:n = {
1383     \tl_set:Nn
1384         \l__UWMad_Nomenclature_Section_Subgroup_tl {#1}
1385 },

```

The default nomenclature section is chapter. Since the other two groups of empty by default, the Nomenclature environment will handle them.

```

1386 main-section .default:n = chapter,

```

Group title adjustments:

```

1387 main-title .code:n = {
1388     \tl_set:Nn
1389         \l__UWMad_Nomenclature_Title_Main_tl {#1}
1390 },
1391 group-title .code:n = {
1392     \tl_set:Nn
1393         \l__UWMad_Nomenclature_Title_Group_tl {#1}
1394 },
1395 subgroup-title .code:n = {
1396     \tl_set:Nn
1397         \l__UWMad_Nomenclature_Title_Subgroup_tl {#1}
1398 },

```


Group title default for main group only:

```
1399     main-title .default:n = Nomenclature,
```

Miscellaneous options:

```
1400     numbered      .bool_gset:N =
1401         \g__UWMad_Nomenclature_IsNumbered_bool,
1402     include-in-toc .bool_gset:N =
1403         \g__UWMad_Nomenclature_IncludeInTOC_bool,
1404     with-units     .bool_gset:N =
1405         \g__UWMad_Nomenclature_IncludeUnitsColumn_bool,
```

Miscellaneous option defaults:

```
1406     numbered      .default:n = false,
1407     include-in-toc .default:n = true,
1408     with-units     .default:n = false
1409 }
```

And the defaults for all keys are now set.

```
1410 \keys_set:nn { UWMad/Nomenclature } {
1411     title-skip      ,
1412     print-skip      ,
1413     entry-margin-left ,
1414     entry-margin-bottom ,
1415     entry-padding    ,
1416     numbered        ,
1417     include-in-toc   ,
1418     with-units       ,
1419     main-section     ,
1420     main-title
1421 }
```

And the defaults for all keys are now set.

<code>\UWMad_Nomenclature_UpdateWidest:Nn</code>	<code>\UWMad_Nomenclature_UpdateWidest:Nn<dim>{\object}</code>
<code>\UWMad_Nomenclature_UpdateWidest_Symbol:n</code>	<code>\UWMad_Nomenclature_UpdateWidest_Symbol:n{\symbol}</code>
<code>\UWMad_Nomenclature_UpdateWidest_Units:n</code>	<code>\UWMad_Nomenclature_UpdateWidest_Units:n{\units}</code>

These commands update the widest symbol and widest unit lengths.

```

1422 \cs_new:Nn \UWMad_Nomenclature_UpdateWidest:Nn {
1423   \hbox_set:Nn \l_tmpa_box {#2}
1424   \dim_set:Nn \l_tmpa_dim {\box_wd:N \l_tmpa_box}
1425   \dim_compare:nNnTF {#1} < {\l_tmpa_dim} {
1426     \dim_set:Nn #1 {\l_tmpa_dim}
1427   } { }
1428 }
1429 \cs_new:Nn \UWMad_Nomenclature_UpdateWidest_Symbol:n {
1430   \UWMad_Nomenclature_UpdateWidest:Nn
1431   \l__UWMad_Nomenclature_WidestSymbol_dim {#1}
1432 }
1433 %
1434 \cs_new:Nn \UWMad_Nomenclature_UpdateWidest_Units:n {
1435   \UWMad_Nomenclature_UpdateWidest:Nn
1436   \l__UWMad_Nomenclature_WidestUnit_dim {#1}
1437 }
1438 %
1439 %
1440 \cs_new:Nn \UWMad_Nomenclature_ZeroWidest_Symbol: {
1441   \dim_set:Nn \l__UWMad_Nomenclature_WidestSymbol_dim {0pt}
1442 }
1443 \cs_new:Nn \UWMad_Nomenclature_ZeroWidest_Unit: {
1444   \dim_set:Nn \l__UWMad_Nomenclature_WidestUnit_dim {0pt}
1445 }

1446 %
1447 %
1448 % ===== %
1449 %                               Set Entry Widths                               %
1450 % ===== %
1451 \cs_new:Nn \UWMad_Nomenclature_SetEntryWidths_NoUnits: {
1452   %
1453   % Define symbol width
1454   \dim_set:Nn \l__UWMad_Nomenclature_Entry_WidthSymbol_dim {
1455     1.01\l__UWMad_Nomenclature_WidestSymbol_dim
1456   }
1457   %
1458   % Define description width
1459   \dim_set:Nn \l__UWMad_Nomenclature_Entry_WidthDescription_dim {
1460     0.999\textwidth -
1461     \l__UWMad_Nomenclature_Entry_MarginLeft_dim -
1462     \l__UWMad_Nomenclature_Entry_WidthSymbol_dim -
1463     \l__UWMad_Nomenclature_Entry_Padding_dim
1464   }
1465 }
1466 %
1467 \cs_new:Nn \UWMad_Nomenclature_SetEntryWidths_Units: {

```

```

1468 %
1469 % Define symbol width
1470 \dim_set:Nn \l__UWMad_Nomenclature_Entry_WidthSymbol_dim {
1471     1.01\l__UWMad_Nomenclature_WidestSymbol_dim
1472 }
1473 %
1474 % Define unit width
1475 \dim_set:Nn \l__UWMad_Nomenclature_Entry_WidthUnit_dim {
1476     1.01\l__UWMad_Nomenclature_WidestUnit_dim
1477 }
1478 %
1479 % Define description width
1480 \dim_set:Nn \l__UWMad_Nomenclature_Entry_WidthDescription_dim {
1481     0.999\textwidth -
1482     \l__UWMad_Nomenclature_Entry_MarginLeft_dim -
1483     \l__UWMad_Nomenclature_Entry_WidthSymbol_dim -
1484     \l__UWMad_Nomenclature_Entry_WidthUnit_dim -
1485     2\l__UWMad_Nomenclature_Entry_Padding_dim
1486 }
1487 }
1488 %
1489 \cs_new:Nn \UWMad_Nomenclature_SetEntryWidths: {
1490     \bool_if:NTF \g__UWMad_Nomenclature_IncludeUnitsColumn_bool {
1491         \UWMad_Nomenclature_SetEntryWidths_Units:
1492     } {
1493         \UWMad_Nomenclature_SetEntryWidths_NoUnits:
1494     }
1495 }
1496 %
1497 %
1498 %
1499 %
1500 % ===== %
1501 %                               Typeset Entry %
1502 % ===== %
1503 %
1504 \coffin_new:N \l_tmpc_coffin
1505 %
1506 \cs_new:Nn \UWMad_Nomenclature_SetEntry_NoUnits:nn {
1507 % Set the entry material in the temporary coffins
1508     \vcoffin_set:Nnn
1509         \l_tmpa_coffin
1510         {\l__UWMad_Nomenclature_Entry_WidthSymbol_dim}
1511         {#1}
1512     \vcoffin_set:Nnn
1513         \l_tmpb_coffin
1514         {\l__UWMad_Nomenclature_Entry_WidthDescription_dim}
1515         {#2}
1516 %
1517 % Typeset the material and skips
1518     \group_begin:
1519         \setstretch{1.1}
1520         \skip_horizontal:n {\l__UWMad_Nomenclature_Entry_MarginLeft_dim}
1521         \coffin_typeset:Nnnnn \l_tmpa_coffin {l}{t}{Opt}{Opt}
1522         \skip_horizontal:n {\l__UWMad_Nomenclature_Entry_Padding_dim}
1523         \coffin_typeset:Nnnnn \l_tmpb_coffin {l}{t}{Opt}{Opt}

```

```

1524         \skip_vertical:n {\l__UWMad_Nomenclature_Entry_MarginBottom_dim}
1525     \group_end:
1526 }
1527 %
1528 \cs_new:Nn \UWMad_Nomenclature_SetEntry_Units:nnn {
1529 % Set the entry material in the temporary coffins
1530     \vcoffin_set:Nnn
1531         \l_tmpa_coffin
1532         {\l__UWMad_Nomenclature_Entry_WidthSymbol_dim}
1533         {#1}
1534     \vcoffin_set:Nnn
1535         \l_tmpb_coffin
1536         {\l__UWMad_Nomenclature_Entry_WidthUnit_dim}
1537         {#2}
1538     \vcoffin_set:Nnn
1539         \l_tmpc_coffin
1540         {\l__UWMad_Nomenclature_Entry_WidthDescription_dim}
1541         {#3}
1542 %
1543 % Typeset the material and skips
1544     \group_begin:
1545         \setstretch{1.1}
1546         \skip_horizontal:n {\l__UWMad_Nomenclature_Entry_MarginLeft_dim}
1547         \coffin_typeset:Nnnnn \l_tmpa_coffin {l}{t}{Opt}{Opt}
1548         \skip_horizontal:n {\l__UWMad_Nomenclature_Entry_Padding_dim}
1549         \coffin_typeset:Nnnnn \l_tmpb_coffin {l}{t}{Opt}{Opt}
1550         \skip_horizontal:n {\l__UWMad_Nomenclature_Entry_Padding_dim}
1551         \coffin_typeset:Nnnnn \l_tmpc_coffin {l}{t}{Opt}{Opt}
1552         \skip_vertical:n {\l__UWMad_Nomenclature_Entry_MarginBottom_dim}
1553     \group_end:
1554 }
1555 %
1556 \cs_new:Nn \UWMad_Nomenclature_SetEntry: {
1557     \bool_if:NTF \g__UWMad_Nomenclature_IncludeUnitsColumn_bool {
1558         \UWMad_Nomenclature_SetEntry_Units:
1559     } {
1560         \UWMad_Nomenclature_SetEntry_NoUnits:
1561     }
1562 }
1563 %
1564 %
1565 %
1566 %
1567 %
1568 % ===== %
1569 %                               User Front-Ends %
1570 % ===== %
1571 \DeclareDocumentCommand \NomenclatureSetup { m } {
1572     \keys_set:nn { UWMad/Nomenclature } { #1 }
1573 }
1574 %
1575 %
1576 \DeclareDocumentEnvironment {Nomenclature} { o g } {
1577 %
1578 % Create the ListOf
1579     \UWMad_ListOf_Define:n {Nomenclature}

```

```

1580 %
1581 %
1582 % Check for an optional section declaration and
1583 % set Main section token list.
1584 \IfNoValueTF {#1} { } {
1585     \UWMad_IfSectionExists:nT {#1} { }
1586     \tl_set:Nn \l__UWMad_Nomenclature_Section_Main_tl {#1}
1587 }
1588 %
1589 %
1590 % Check for an optional section declaration and
1591 % set Main section token list.
1592 \IfNoValueTF {#2} { } {
1593     \tl_set:Nf
1594         \l__UWMad_Nomenclature_Title_Main_tl {#2}
1595 }
1596 \UWMad_ListOf_SetTitle_Main:nn {Nomenclature}
1597 {\l__UWMad_Nomenclature_Title_Main_tl}
1598 %
1599 %
1600 \bool_if:NTF \g__UWMad_Nomenclature_IsNumbered_bool {
1601     \UWMad_ListOf_MakeNumbered:n {Nomenclature}
1602 }{
1603     \UWMad_ListOf_MakeNotNumbered:n {Nomenclature}
1604 }
1605 %
1606 %
1607 %
1608 % If Group section token list is empty, set it to the following
1609 % section after Main in the LaTeX sectioning hierarchy.
1610 % Otherwise, take the value at its word.
1611 \tl_if_empty:NTF \l__UWMad_Nomenclature_Section_Group_tl {
1612     \tl_set:Nf \l__UWMad_Nomenclature_Section_Group_tl {
1613         \UWMad_NextSection:n {
1614             \l__UWMad_Nomenclature_Section_Main_tl
1615         }
1616     }
1617 } { }
1618 %
1619 % If Subgroup section token list is empty, set it to the following
1620 % section after Group in the LaTeX sectioning hierarchy.
1621 % Otherwise, take the value at its word.
1622 \tl_if_empty:NTF \l__UWMad_Nomenclature_Section_Subgroup_tl {
1623     \tl_set:Nf \l__UWMad_Nomenclature_Section_Subgroup_tl {
1624         \UWMad_NextSection:n {
1625             \l__UWMad_Nomenclature_Section_Group_tl
1626         }
1627     }
1628 } { }
1629 %
1630 % Set the sections with the Nomenclature ListOf instance
1631 \UWMad_ListOf_SetSection_Main:nn
1632 {Nomenclature} {\l__UWMad_Nomenclature_Section_Main_tl}
1633 \UWMad_ListOf_SetSection_Group:nn
1634 {Nomenclature} {\l__UWMad_Nomenclature_Section_Group_tl}
1635 \UWMad_ListOf_SetSection_Subgroup:nn

```

```

1636         {Nomenclature} {\l__UWMad_Nomenclature_Section_Subgroup_tl}
1637 %
1638 %
1639 % Determine if this nomenclature should be in the Table of Contents
1640 \bool_if:NTF \g__UWMad_Nomenclature_IncludeInTOC_bool {
1641     \UWMad_ListOf_IncludeInTOC:n {Nomenclature}
1642 } {
1643     \UWMad_ListOf_DoNotIncludeInTOC:n {Nomenclature}
1644 }
1645 %
1646 % Set some hooks in the Nomenclature ListOf instance
1647 \UWMad_ListOf_SetHook:nnn {Nomenclature} {PrePrint} {
1648     \UWMad_Nomenclature_SetEntryWidths:
1649 }
1650 \UWMad_ListOf_SetHook:nnn {Nomenclature} {PostPrint} {
1651     \UWMad_Nomenclature_ZeroWidest_Symbol:
1652 }
1653 %
1654 %
1655 % User front-end for creating a Group
1656 \DeclareDocumentCommand \Group { G{} } {
1657     \IfNoValueTF {##1} { } {
1658         \tl_set:Nn
1659             \l__UWMad_Nomenclature_Title_Group_tl {##1}
1660     }
1661     \UWMad_ListOf_SetTitle_Group:nn {Nomenclature}
1662         {\l__UWMad_Nomenclature_Title_Group_tl}
1663     \UWMad_ListOf_StartGroup:nn{Nomenclature}{Group}
1664 }
1665 %
1666 % User front-end for creating a Subgroup
1667 \DeclareDocumentCommand \Subgroup { G{} } {
1668     \IfNoValueTF {##1} { } {
1669         \tl_set:Nn
1670             \l__UWMad_Nomenclature_Title_Subgroup_tl {##1}
1671     }
1672     \UWMad_ListOf_SetTitle_Subgroup:nn {Nomenclature}
1673         {\l__UWMad_Nomenclature_Title_Subgroup_tl}
1674     \UWMad_ListOf_StartGroup:nn{Nomenclature}{Subgroup}
1675 }
1676 %
1677 % User front-end for creating an entry
1678 \bool_if:NTF \g__UWMad_Nomenclature_IncludeUnitsColumn_bool {
1679     \DeclareDocumentCommand \Entry { m m m } {
1680         \UWMad_ListOf_PushEntry:nn {Nomenclature} {
1681             \UWMad_Nomenclature_SetEntry_Units:nnn
1682                 {##1} {##2} {##3}
1683         }
1684         \UWMad_Nomenclature_UpdateWidest_Symbol:n{##1}
1685     }
1686 } {
1687     \DeclareDocumentCommand \Entry { m m } {
1688         \UWMad_ListOf_PushEntry:nn {Nomenclature} {
1689             \UWMad_Nomenclature_SetEntry_NoUnits:nn
1690                 {##1} {##2}
1691         }

```

```

1692         \UWMad_Nomenclature_UpdateWidest_Symbol:n{##1}
1693     }
1694 }
1695 %
1696 % User front-end for resetting the column width
1697 \DeclareDocumentCommand \ResetColumnWidth { } {
1698     \UWMad_Nomenclature_ZeroWidest_Symbol:
1699     \UWMad_Nomenclature_ZeroWidest_Unit:
1700 }
1701 %
1702 %
1703 % Print the main section title
1704 \UWMad_ListOf_PrintTitle:nn {Nomenclature}{Main}
1705 %
1706 } {
1707 % Flush the remaining entries from the ListOf queue and
1708 % delete the Nomenclature ListOf instance.
1709 \UWMad_ListOf_PrintEntries:n {Nomenclature}
1710 \UWMad_ListOf_Delete:n{Nomenclature}
1711 }
1712 %
1713 %
1714 %
1715 %
1716 \DeclareDocumentEnvironment {Acronym} { o G{Acronym} } {
1717
1718     \begin{Nomenclature}[#1]{#2}
1719 %
1720 %
1721     \UWMad_Hash_Define:n{Acronyms}
1722     \UWMad_Hash_Define:n{AcronymMeanings}
1723 %
1724 %
1725     \cs_undefine:N \Entry
1726     \DeclareDocumentCommand \Entry { o m m } {
1727         \IfNoValueTF {##1} {
1728
1729             \UWMad_Hash_Set:nnn{Acronyms}          {##2}{##2}
1730             \UWMad_Hash_Set:nnn{AcronymMeanings}{##2}{##3}
1731             \bool_new:c {g__UWMad_Acronym_WasSet_##2_bool}
1732             %
1733             \UWMad_ListOf_PushEntry:nn {Nomenclature} {
1734                 \hypertarget{Acronym:##2}{}
1735                 \UWMad_Nomenclature_SetEntry_NoUnits:nn
1736                 {##2} {##3}
1737             }
1738
1739         } {
1740
1741             \UWMad_Hash_Set:nnn{Acronyms}          {##1}{##2}
1742             \UWMad_Hash_Set:nnn{AcronymMeanings}{##1}{##3}
1743             \bool_new:c {g__UWMad_Acronym_WasSet_##1_bool}
1744             %
1745             \UWMad_ListOf_PushEntry:nn {Nomenclature} {
1746                 \hypertarget{Acronym:##1}{}
1747                 \UWMad_Nomenclature_SetEntry_NoUnits:nn

```

```

1748         {##2} {##3}
1749     }
1750
1751 }
1752 \UWMad_Nomenclature_UpdateWidest_Symbol:n{##2}
1753 }
1754 } {
1755
1756 \end{Nomenclature}
1757
1758 }
1759
1760 \DeclareDocumentCommand \Acro { m } {
1761     \UWMad_Hash_IfKeySet:nnTF {Acronyms} {#1} {
1762         \bool_if:cTF {g__UWMad_Acronym_WasSet_#1_bool} {
1763             \hyperlink{Acronym:#1}{
1764                 \UWMad_Hash_Get:nn{Acronyms}{#1}
1765             }
1766         } {
1767             \UWMad_Hash_Get:nn{AcronymMeanings}{#1}~
1768             (
1769                 \UWMad_Hash_Get:nn{Acronyms}{#1}
1770             )
1771             \bool_gset_true:c {g__UWMad_Acronym_WasSet_#1_bool}
1772         }
1773     } { }
1774 }
1775 %
1776 %
1777 \cs_new_eq:NN \AcronymSetup \NomenclatureSetup
1778 %
1779 %
1780 %

```


Module 7

Thesis and PDF Information

7.1 Metadata clist and Aux Write

Since the metadata (i.e., properties) of a PDF must be set in the preamble but typically a user defines them in the document, these routines write the supported metadata that a user may define to an auxiliary file that is then imported upon recompilation. It uses the `expl3` `clist` commands to define and build the CSV list, and then writes to the file.

Define the `clist`.

```
1781 \clist_new:N \g__UWMad_MetaDataList_clist
```

Define a command for pushing entries (with a brace guard) on to the `clist`.

```
1782 \cs_new:Nn \UWMad_MetaData_PushToList:nn {
1783   \clist_gput_right:Nn \g__UWMad_MetaDataList_clist {
1784     #1={#2}
1785   }
1786 }
```

Define to booleans: one to tell if a auxiliary file is needed and to tell if the `document` has begun.

```
1787 \bool_new:N \g__UWMad_MetaData_GenerateAux_bool
1788 \bool_new:N \g__UWMad_MetaData_IsDocument_bool
```

Look for a auxiliary file and load it if it exists.

```
1789 \file_if_exist:nTF{\c_job_name_tl.UWMad.PDFMetaData.aux} {
1790   \file_input:n {\c_job_name_tl.UWMad.PDFMetaData.aux}
1791 }{} }
```

At the beginning of the document, if data has been pushed to the list, pass it to `\hypersetup` so the PDF gets it. Also, set the `IsDocument` boolean true.

```

1792 \AtBeginDocument{
1793     \clist_if_empty:NTF \g__UWMad_MetaDataList_clist { } {
1794         \exp_args:Nx \hypersetup {
1795             \clist_use:Nn\g__UWMad_MetaDataList_clist{,}
1796         }
1797     } { }
1798     \bool_gset_true:N \g__UWMad_MetaData_IsDocument_bool
1799 }

```

If thesis information of PDF metadata was used within document, write that information to an auxiliary file.

```

1800 \AtEndDocument{
1801     \bool_if:NTF \g__UWMad_MetaData_GenerateAux_bool {
1802         \clist_if_empty:NTF \g__UWMad_MetaDataList_clist { } {
1803             \iow_new:N \g__UWMad_PDFMetaData_HyperSetup_io
1804             \iow_open:Nn \g__UWMad_PDFMetaData_HyperSetup_io {
1805                 \c_job_name_tl.UWMad.PDFMetaData.aux
1806             }
1807             \iow_now:Nx \g__UWMad_PDFMetaData_HyperSetup_io {
1808                 \noexpand\ExplSyntaxOff
1809                 \noexpand\hypersetup
1810                 {\clist_use:Nn\g__UWMad_MetaDataList_clist{,}}
1811                 \noexpand\ExplSyntaxOn
1812             }
1813             \iow_close:N \g__UWMad_PDFMetaData_HyperSetup_io
1814         } { }
1815     } { }
1816 }

```

7.2 Thesis Information

Declare the ThesisInfo token list variables.

```

1817 \tl_new:N \g__UWMad_ThesisInfo_Title_tl
1818 \tl_new:N \g__UWMad_ThesisInfo_Author_tl
1819 \tl_new:N \g__UWMad_ThesisInfo_DefenseDate_tl
1820 \tl_new:N \g__UWMad_ThesisInfo_Department_tl
1821 \tl_new:N \g__UWMad_ThesisInfo_Program_tl
1822 \tl_new:N \g__UWMad_ThesisInfo_Degree_tl
1823 \tl_new:N \g__UWMad_ThesisInfo_DocumentType_tl
1824 \tl_new:N \g__UWMad_ThesisInfo_AdvisorName_tl
1825 \tl_new:N \g__UWMad_ThesisInfo_AdvisorPosition_tl
1826 \tl_new:N \g__UWMad_ThesisInfo_AdvisorAssociation_tl
1827 \tl_new:N \g__UWMad_ThesisInfo_AdvisorMarker_tl
1828 \tl_new:N \g__UWMad_ThesisInfo_Institution_tl

```

Set the document type default.

```
1829 \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {report}
```

Define some booleans for required information/

```
1830 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Title_bool
1831 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Author_bool
1832 \bool_new:N \g__UWMad_ThesisInfo_IsSet_DefenseDate_bool
1833 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Program_bool
1834 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
1835 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Institution_bool
1836 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool
```

Declare the user front-end for the title.

```
1837 \DeclareDocumentCommand \Title { m } {
```

Set the associated token list variable

```
1838 \tl_gset:Nn \g__UWMad_ThesisInfo_Title_tl {#1}
```

Pass it to the default L^AT_EX \title command.

```
1839 \title{#1}
```

Push the value to the MetaData clist.

```
1840 \UWMad_MetaData_PushToList:nn{pdftitle} {#1}
```

If this command was used within the document, tell the class to write an auxiliary file.

```
1841 \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
1842 \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
1843 } { }
```

Tell the class this variable is now set.

```
1844 \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Title_bool
1845 }
```

Similar flow to the \Title definition.

```
1846 \DeclareDocumentCommand \Author { m } {
1847 \tl_gset:Nn \g__UWMad_ThesisInfo_Author_tl {#1}
1848 \author{#1}
```

```

1849     \UWMad_MetaData_PushToList:nn{pdfauthor}    {#1}
1850     \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
1851         \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
1852     } { }
1853     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Author_bool
1854 }

```

A simple setter command.

```

1855 \DeclareDocumentCommand \Program { m } {
1856     \tl_gset:Nn \g__UWMad_ThesisInfo_Program_tl {#1}
1857     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Program_bool
1858 }

```

A simple setter command.

```

1859 \DeclareDocumentCommand \Degree { m } {
1860     \tl_gset:Nn \g__UWMad_ThesisInfo_Degree_tl {#1}
1861     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
1862 }

```

A simple setter command.

```

1863 \DeclareDocumentCommand \DocumentType { m } {
1864     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {#1}
1865 }

```

A semantic setter command.

```

1866 \DeclareDocumentCommand \Dissertation { } {
1867     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
1868         dissertation
1869     }
1870 }

```

A semantic setter command.

```

1871 \DeclareDocumentCommand \DoctoralThesis { } {
1872     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
1873         doctoral~thesis
1874     }
1875 }

```

A semantic setter command.

```

1876 \DeclareDocumentCommand \MastersThesis { } {
1877     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {

```

```

1878         master's~thesis
1879     }
1880 }

```

A semantic setter command.

```

1881 \DeclareDocumentCommand \Thesis { } {
1882     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
1883         thesis
1884     }
1885 }

```

A semantic setter command.

```

1886 \DeclareDocumentCommand \Prelim { } {
1887     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
1888         preliminary~report
1889     }
1890 }

```

A simple setter command and aliases.

```

1891 \DeclareDocumentCommand \DefenseDate { m } {
1892     \tl_gset:Nn \g__UWMad_ThesisInfo_DefenseDate_tl {#1}
1893     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_DefenseDate_bool
1894 }
1895 \cs_gset_eq:NN \DefenceDate \DefenseDate

```

A simple setter command and aliases.

```

1896 \DeclareDocumentCommand \Institution { m } {
1897     \tl_gset:Nn \g__UWMad_ThesisInfo_Institution_tl {#1}
1898     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Institution_bool
1899 }
1900 \cs_set_eq:NN \University \Institution
1901 %
1902 %
1903 %
1904 %
1905 %
1906 % User front-ends (Optional)
1907 \DeclareDocumentCommand \Department { m } {
1908     \tl_gset:Nn \g__UWMad_ThesisInfo_Department_tl {#1}
1909 }
1910 \DeclareDocumentCommand \Advisor { m m m } {
1911     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool
1912     \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorName_tl {#1}
1913     \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorPosition_tl {#2}
1914     \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorAssociation_tl {#3}
1915     \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorMarker_tl {Advisor}

```

```

1916 }
1917 \DeclareDocumentCommand \Adviser { m m m } {
1918     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool
1919     \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorName_tl      {#1}
1920     \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorPosition_tl  {#2}
1921     \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorAssociation_tl {#3}
1922     \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorMarker_tl    {Adviser}
1923 }
1924 %
1925 %
1926 %
1927 \msg_new:nnn { UWMadThesis } { ThesisInfo / UnsetInformation } {
1928     The~required~information~for~the~#1~is~not~set.
1929 }
1930 \DeclareDocumentCommand \IfInfoIsSetT { m +m } {
1931     \bool_if:cTF {g__UWMad_ThesisInfo_IsSet_ #1 _bool} {
1932         #2
1933     } {
1934         \msg_error:nnn
1935             { UWMadThesis }
1936             { ThesisInfo / UnsetInformation }
1937             {#1}
1938     }
1939 }
1940 %
1941 %
1942 %
1943 %
1944 % User front-end accessors
1945 \DeclareDocumentCommand \TheTitle { } {
1946     \g__UWMad_ThesisInfo_Title_tl
1947 }
1948 \DeclareDocumentCommand \TheAuthor { } {
1949     \g__UWMad_ThesisInfo_Author_tl
1950 }
1951 \DeclareDocumentCommand \TheProgram { } {
1952     \g__UWMad_ThesisInfo_Program_tl
1953 }
1954 \DeclareDocumentCommand \TheDegree { } {
1955     \g__UWMad_ThesisInfo_Degree_tl
1956 }
1957 \DeclareDocumentCommand \TheDocumentType { } {
1958     \g__UWMad_ThesisInfo_DocumentType_tl
1959 }
1960 \DeclareDocumentCommand \TheDefenseDate { } {
1961     \g__UWMad_ThesisInfo_DefenseDate_tl
1962 }
1963 \cs_gset_eq:NN \TheDefenceDate \TheDefenseDate
1964 \DeclareDocumentCommand \TheInstitution { } {
1965     \g__UWMad_ThesisInfo_Institution_tl
1966 }
1967 \cs_set_eq:NN \TheUniversity \TheInstitution
1968 %
1969 \DeclareDocumentCommand \TheDepartment { } {
1970     \g__UWMad_ThesisInfo_Department_tl
1971 }

```

```

1972 \DeclareDocumentCommand \TheAdvisor { } {
1973     \g__UWMad_ThesisInfo_AdvisorName_tl
1974 }
1975 %
1976 %
1977 %
1978 \int_new:N \l__UWMad_ThesisInfo_CommitteeCount_int
1979 \UWMad_ListOf_Define:n {CommitteeList}
1980 \DeclareDocumentCommand \CommitteeMember { m m m } {
1981     \int_incr:N \l__UWMad_ThesisInfo_CommitteeCount_int
1982     \UWMad_ListOf_PushEntry:nn {CommitteeList} {
1983         #1,~#2,~#3\skip_vertical:n{-1em}
1984     }
1985 }
1986 \DeclareDocumentCommand \PrintCommitteeMemberList { } {
1987     \bool_if:NTF \g__UWMad_ThesisInfo_IsSet_Advisor_bool {
1988         \g__UWMad_ThesisInfo_AdvisorName_tl{ },~
1989         \g__UWMad_ThesisInfo_AdvisorPosition_tl{ },~
1990         \g__UWMad_ThesisInfo_AdvisorAssociation_tl{ }~
1991         (\g__UWMad_ThesisInfo_AdvisorMarker_tl{ })
1992         \skip_vertical:n{-1em}
1993     } { }
1994     \UWMad_ListOf_PrintEntries:n {CommitteeList}
1995 }
1996 %
1997 %
1998 %
1999 %
2000 %
2001 \tl_new:N \g__UWMad_PDFMetaData_Subject_tl
2002 \tl_new:N \g__UWMad_PDFMetaData_Keywords_tl
2003 \tl_new:N \g__UWMad_PDFMetaData_Producer_tl
2004 \tl_new:N \g__UWMad_PDFMetaData_Creator_tl
2005 %
2006 %
2007 %
2008 % User front-end (Optional)
2009 \DeclareDocumentCommand \Subject { m } {
2010     \tl_gset:Nn \g__UWMad_PDFMetaData_Subject_tl {#1}
2011     \UWMad_MetaData_PushToList:nn{pdfsubject} {#1}
2012     \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
2013         \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
2014     } { }
2015 }
2016 \DeclareDocumentCommand \Keywords { m } {
2017     \tl_gset:Nn \g__UWMad_PDFMetaData_Keywords_tl {#1}
2018     \UWMad_MetaData_PushToList:nn{pdfproducer} {#1}
2019     \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
2020         \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
2021     } { }
2022 }
2023 \DeclareDocumentCommand \Producer { m } {
2024     \tl_gset:Nn \g__UWMad_PDFMetaData_Producer_tl {#1}
2025     \UWMad_MetaData_PushToList:nn{pdfcreator} {#1}
2026     \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
2027         \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool

```

```

2028     } { }
2029 }
2030 \DeclareDocumentCommand \Creator { m } {
2031     \tl_gset:Nn \g__UWMad_PDFMetaData_Creator_tl {#1}
2032     \UWMad_MetaData_PushToList:nn{pdfkeywords} {#1}
2033     \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
2034         \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
2035     } { }
2036 }
2037 %
2038 %
2039 %
2040 % User front-end accessors.
2041 \DeclareDocumentCommand \TheSubject { } {
2042     \g__UWMad_PDFMetaData_Subject_tl
2043 }
2044 \DeclareDocumentCommand \TheKeywords { } {
2045     \g__UWMad_PDFMetaData_Keywords_tl
2046 }
2047 \DeclareDocumentCommand \TheProducer { } {
2048     \g__UWMad_PDFMetaData_Producer_tl
2049 }
2050 \DeclareDocumentCommand \TheCreator { } {
2051     \g__UWMad_PDFMetaData_Creator_tl
2052 }
2053 %

```


Module 8

Special Pages

8.1 MakeTitlePage

```

2054 % That phrase that occurs on every title page design the class author has seen
2055 \DeclareDocumentCommand \FulfillmentClause { } {
2056   {
2057     \setstretch{1.1}
2058     A~\TheDocumentType{}~submitted~in~partial~fulfillment~of~the~
2059     requirements~for~the~degree~of
2060   }
2061 }
2062
2063 \DeclareDocumentCommand \TitlePageTitle { } {
2064   \IfInfoIsSetT {Title} {
2065     {
2066       \LARGE
2067       \textsc {\TheTitle}
2068     }
2069   }
2070 }
2071
2072 \DeclareDocumentCommand \TitlePageAuthor { } {
2073   \IfInfoIsSetT {Author} {
2074     {
2075       \large
2076       by \[0.50em]
2077       \TheAuthor{}
2078     }
2079   }
2080 }
2081
2082 \DeclareDocumentCommand \TitlePageFulFillment { } {
2083   \FulfillmentClause{}
2084 }
2085
2086 \DeclareDocumentCommand \TitlePageDegree { } {
2087   \IfInfoIsSetT {Degree} {
2088     \TheDegree{}
2089   }
2090 }
2091
2092 \DeclareDocumentCommand \TitlePageProgram { } {

```

```

2093     \IfInfoIsSetT {Program} {
2094         \TheProgram{}
2095     }
2096 }
2097
2098 \DeclareDocumentCommand \TitlePageInstitution { } {
2099     \IfInfoIsSetT {Institution} {
2100         \textsc{\TheInstitution{}}
2101     }
2102 }
2103
2104 \DeclareDocumentCommand \TitlePageDefenseDate { } {
2105     \IfInfoIsSetT {DefenseDate} {
2106         Date-of~final~oral~examination:~\TheDefenseDate{}
2107     }
2108 }
2109
2110
2111 \DeclareDocumentCommand \MakeTitlePage { } {
2112     \clearpage
2113     \thispagestyle{empty}
2114     \begin{center}
2115         \TitlePageTitle{}          \\\[1.0em]
2116         \TitlePageAuthor{}         \\\[1.0em]
2117         \vfill
2118         \TitlePageFulFillment{}    \\\[1.0em]
2119         \TitlePageDegree{}         \\\[1.0em]
2120         \TitlePageProgram{}        \\\[1.0em]
2121         \vfill
2122         \TitlePageInstitution{}
2123         \vfill
2124     \end{center}
2125     \TitlePageDefenseDate{}\\\[1.0em]
2126     \PrintCommitteeMemberList{}
2127     \cleardoublepage
2128 }
2129
2130
2131
2132
2133

```

8.2 LicensePage

First, the support code for defining `\Copyright` and `\CreativeCommons` will be given. Then the user front-end will be given through the `LicensePage` environment.

```

2134 \cs_new:Nn \__UWMad_LicensePage_StartPage: {
2135     \clearpage
2136     \thispagestyle{empty}

```

```

2137 \tex_hbox:D{}
2138 \tex_vfill:D
2139 \phantomsection
2140 }
2141 %

```

8.2.1 Copyright

```

2142 \bool_new:N \l__UWMad_Copyright_UseCopyright_bool
2143 \cs_set_eq:NN \CopyrightSymbol \copyright
2144
2145 \cs_set:Nn \__UWMad_Copyright_LicenseText: {
2146   \begin{center}
2147     Copyright~\CopyrightSymbol{}~
2148     \l__UWMad_LicensePage_Year_tl{}~
2149     by~
2150     \l__UWMad_LicensePage_Owner_tl{}
2151   \end{center}
2152 }
2153 %
2154 %
2155 %
2156 %

```

8.2.2 Creative Commons

```

2157 % Token lists
2158 \tl_new:N \l__UWMad_CCLicense_Porting_tl
2159 \tl_new:N \l__UWMad_CCLicense_Version_tl
2160 \tl_new:N \l__UWMad_CCLicense_TypeAbbreviation_tl
2161 \tl_new:N \l__UWMad_CCLicense_TypeWords_tl
2162 \tl_new:N \l__UWMad_CCLicense_URL_Front_tl
2163 \tl_new:N \l__UWMad_CCLicense_URL_Middle_tl
2164 \tl_new:N \l__UWMad_CCLicense_URL_Back_tl
2165 \tl_new:N \l__UWMad_CCLicense_URL_tl
2166 \tl_new:N \l__UWMad_CCLicense_http_tl
2167 \tl_new:N \l__UWMad_CCLicense_URLText_tl
2168 %
2169 % Booleans
2170 \bool_new:N \l__UWMad_CCLicense_UseCreativeCommons_bool
2171 \bool_new:N \l__UWMad_CCLicense_UseAttribution_bool
2172 \bool_new:N \l__UWMad_CCLicense_UseShareAlike_bool
2173 \bool_new:N \l__UWMad_CCLicense_UseNoDerivatives_bool
2174 \bool_new:N \l__UWMad_CCLicense_UseNonCommercial_bool
2175 \bool_new:N \l__UWMad_CCLicense_IsValid_bool

```

```

2176 \bool_set_true:N \l__UWMad_CCLicense_UseAttribution_bool
2177 %
2178 %   Valid license types
2179 \cs_new:cn {l__UWMad_CCLicense_Valid_by :} {}
2180 \cs_new:cn {l__UWMad_CCLicense_Valid_by-sa :} {}
2181 \cs_new:cn {l__UWMad_CCLicense_Valid_by-nd :} {}
2182 \cs_new:cn {l__UWMad_CCLicense_Valid_by-nc :} {}
2183 \cs_new:cn {l__UWMad_CCLicense_Valid_by-nc-sa :} {}
2184 \cs_new:cn {l__UWMad_CCLicense_Valid_by-nc-nd :} {}
2185 %
2186 %   Defaults
2187 \tl_gset:Nn \l__UWMad_CCLicense_Porting_tl {
2188   International
2189 }
2190 \tl_gset:Nn \l__UWMad_CCLicense_Version_tl {
2191   4.0
2192 }
2193 %
2194 %   URL definitions
2195 \tl_set:Nn \l__UWMad_CCLicense_URL_Front_tl {
2196   creativecommons.org/licenses
2197 }
2198 \tl_set:Nn \l__UWMad_CCLicense_URL_Middle_tl {
2199   /\l__UWMad_CCLicense_TypeAbbreviation_tl
2200 }
2201 \tl_set:Nn \l__UWMad_CCLicense_URL_Back_tl {
2202   /\l__UWMad_CCLicense_Version_tl
2203 }
2204 \tl_set:Nn \l__UWMad_CCLicense_URL_tl {
2205   http://
2206   \l__UWMad_CCLicense_URL_Front_tl
2207   \l__UWMad_CCLicense_URL_Middle_tl
2208   \l__UWMad_CCLicense_URL_Back_tl
2209 }
2210 \tl_set:Nn \l__UWMad_CCLicense_http_tl {
2211   http://
2212 }
2213 %
2214 %
2215 \tl_set:Nn \l__UWMad_CCLicense_URLText_tl {
2216   Creative~Commons~
2217   \l__UWMad_CCLicense_TypeWords_tl{ }~
2218   \l__UWMad_CCLicense_Version_tl{ }~
2219   \l__UWMad_CCLicense_Porting_tl{ }
2220 }
2221 %
2222 %
2223 %
2224 %   Type Creator
2225 \cs_new:Nn \__UWMad_CCLicense_CreateType: {
2226
2227   \bool_if:NTF \l__UWMad_CCLicense_UseAttribution_bool {
2228
2229     \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
2230       by
2231     }

```

```

2232         \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
2233             Attribution
2234         }
2235
2236     } { }
2237
2238     \bool_if:NTF \l__UWMad_CCLicense_UseNonCommercial_bool {
2239
2240         \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
2241             -nc
2242         }
2243         \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
2244             -NonCommercial
2245         }
2246
2247     } { }
2248
2249     \bool_if:NTF \l__UWMad_CCLicense_UseShareAlike_bool {
2250
2251         \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
2252             -sa
2253         }
2254         \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
2255             -ShareAlike
2256         }
2257
2258     } { }
2259
2260     \bool_if:NTF \l__UWMad_CCLicense_UseNoDerivatives_bool {
2261
2262         \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
2263             -nd
2264         }
2265         \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
2266             -NoDerivatives
2267         }
2268
2269     } { }
2270 }
2271 %
2272 %
2273 %
2274 % Type Validator
2275 \cs_new:Nn \__UWMad_CCLicense_CheckTypeValidity: {
2276     \cs_if_exist:cTF {
2277         l__UWMad_CCLicense_Valid_
2278         \l__UWMad_CCLicense_TypeAbbreviation_tl :
2279     } {
2280
2281         \bool_set_true:N \l__UWMad_CCLicense_IsValid_bool
2282
2283     } {
2284
2285         \msg_new:nnn {UWMadThesis} {CCLicense / InvalidLicenseType} {
2286             The~license~type~`\l__UWMad_CCLicense_TypeAbbreviation_tl'~
2287             is~not~a~valid~Creative~Commons~license.

```

```

2288     }
2289     \msg_error:nn {UWMadThesis} {CCLicense / InvalidLicenseType}
2290
2291   }
2292 }
2293 %
2294 %
2295 %
2296 %   Page Printer
2297 \cs_new:Nn \__UWMad_CCLicense_LicenseText: {
2298   \begin{center}
2299     \setstretch{1.05}
2300     This~work~is~released~under~a~
2301     \href {\l__UWMad_CCLicense_URL_tl} {
2302       \l__UWMad_CCLicense_URLText_tl
2303     }~
2304     license.\\[0.1em]
2305     \l__UWMad_LicensePage_Owner_tl{},~
2306     \l__UWMad_LicensePage_Year_tl{}
2307   \end{center}
2308 }
2309 %

```

8.2.3 LicensePage Proper

```

2310 %
2311 \tl_new:N \l__UWMad_LicensePage_Year_tl
2312 \tl_new:N \l__UWMad_LicensePage_Owner_tl
2313 %
2314 \tl_set:Nn \l__UWMad_LicensePage_Owner_tl {
2315   \g__UWMad_ThesisInfo_Author_tl
2316 }
2317 \tl_set:Nn \l__UWMad_LicensePage_Year_tl {
2318   \the\year
2319 }
2320 %
2321 %
2322 %
2323 \DeclareDocumentEnvironment {LicensePage} { } {
2324 %
2325 %
2326 %
2327 \DeclareDocumentCommand \LicenseOwner { m } {
2328   \tl_set:Nn \l__UWMad_LicensePage_Owner_tl {
2329     ##1
2330   }
2331 }
2332 \DeclareDocumentCommand \TheLicenseOwner { } {
2333   \l__UWMad_LicensePage_Owner_tl
2334 }
2335 %

```

```

2336 \DeclareDocumentCommand \LicenseYear { m } {
2337     \tl_set:Nn \l__UWMad_LicensePage_Year_tl {
2338         ##1
2339     }
2340 }
2341 \DeclareDocumentCommand \TheLicenseYear { } {
2342     \l__UWMad_LicensePage_Year_tl
2343 }
2344 %
2345 %
2346 %
2347 \DeclareDocumentCommand \Copyright { } {
2348     \bool_set_true:N \l__UWMad_Copyright_UseCopyright_bool
2349 }
2350 \cs_set_eq:NN \AllRightsReserved \Copyright
2351 %
2352 %
2353 %
2354 % User front ends
2355 \DeclareDocumentCommand \CreativeCommons { } {
2356     \bool_set_true:N \l__UWMad_CCLicense_UseCreativeCommons_bool
2357 }
2358 \DeclareDocumentCommand \Attribution { } {
2359     \bool_set_true:N \l__UWMad_CCLicense_UseAttribution_bool
2360 }
2361 \DeclareDocumentCommand \NonCommercial { } {
2362     \bool_set_true:N \l__UWMad_CCLicense_UseNonCommercial_bool
2363 }
2364 \DeclareDocumentCommand \ShareAlike { } {
2365     \bool_set_true:N \l__UWMad_CCLicense_UseShareAlike_bool
2366 }
2367 \DeclareDocumentCommand \NoDerivs { } {
2368     \bool_set_true:N \l__UWMad_CCLicense_UseNoDerivatives_bool
2369 }
2370 %
2371 %
2372 \DeclareDocumentCommand \CCVersion { m } {
2373     \tl_set:Nn \l__UWMad_CCLicense_Version_tl {##1}
2374 }
2375 %
2376 \DeclareDocumentCommand \CCPorting { m } {
2377     \tl_set:Nn \l__UWMad_CCLicense_Porting_tl {##1}
2378 }
2379 %
2380 \DeclareDocumentCommand \CCURL { m } {
2381     \tl_set:Nn \l__UWMad_CCLicense_URL_Front_tl {##1}
2382     \tl_set:Nn \l__UWMad_CCLicense_URL_Middle_tl {/.}
2383     \tl_set:Nn \l__UWMad_CCLicense_URL_Back_tl {}
2384 }
2385 %
2386 \DeclareDocumentCommand \CCURLText { m } {
2387     \tl_set:Nn \l__UWMad_CCLicense_URLText_tl {##1}
2388 }
2389 %
2390 %
2391 } {

```

```

2392
2393 \bool_if:nTF {
2394     \l__UWMad_CCLicense_UseCreativeCommons_bool &&
2395     \l__UWMad_Copyright_UseCopyright_bool
2396 } {
2397     \msg_new:nnn { UWMadThesis } { SpecialPages / MultipleLicenses } {
2398         Both~Creative~Commons~and~Copyright~have~been~declared.~
2399         Please,~pick~one.
2400     }
2401     \msg_error:nn { UWMadThesis } { SpecialPages / MultipleLicenses }
2402 } { }
2403
2404
2405
2406 \bool_if:NTF \l__UWMad_CCLicense_UseCreativeCommons_bool {
2407
2408     \__UWMad_CCLicense_CreateType:
2409     \__UWMad_CCLicense_CheckTypeValidity:
2410     \bool_if:NTF \l__UWMad_CCLicense_IsValid_bool {
2411         \cs_new_eq:NN
2412             \__UWMad_LicensePage_LicenseText:
2413             \__UWMad_CCLicense_LicenseText:
2414     } { }
2415
2416 } { }
2417
2418
2419
2420 \bool_if:NTF \l__UWMad_Copyright_UseCopyright_bool {
2421     \cs_new_eq:NN
2422         \__UWMad_LicensePage_LicenseText:
2423         \__UWMad_Copyright_LicenseText:
2424 } { }
2425
2426
2427
2428 \cs_if_exist:NTF \__UWMad_LicensePage_LicenseText: {
2429     \__UWMad_LicensePage_StartPage:
2430     \vbox_to_ht:nn {0.3333\textheight} {
2431         \__UWMad_LicensePage_LicenseText:
2432     }
2433 } { }
2434
2435
2436 }
2437 %

```

2438 \ExplSyntaxOff

Change History

1.0

General: Hello [1](#)

Index

The italic numbers denote the pages where the corresponding entry is described, numbers underlined point to the definition, all others indicate the places where it is used.

Symbols

_UWMad_IfDefined:nnnnT	197
_UWMad_IfUndefined:nnnnT	200
_UWMad_Appendix_Initialize:	649, 663
_UWMad_CCLicense_CheckTypeValidity:	2269, 2402
_UWMad_CCLicense_CreateType:	2219, 2401
_UWMad_CCLicense_LicenseText:	2291, 2305
_UWMad_CCLicense_PrintPage:	2303, 2404
_UWMad_Copyright_LicenseText:	2134, 2145
_UWMad_Copyright_PrintPage:	2143, 2409
_UWMad_Deque_IfDefined:nT	302, 302, 314, 319, 325, 330, 335, 341, 347, 353
_UWMad_Deque_IfUndefined:nT	302, 305, 309
_UWMad_FrontMatter_Register:nn	679, 723, 765, 780, 795
_UWMad_FrontMatter_StyleWarning:n	44, 63, 68, 619, 625
_UWMad_Hash_IfDefined:nT	367, 367, 379, 385, 390, 396, 405, 410
_UWMad_Hash_IfUndefined:nT	367, 370, 374
_UWMad_IfDefined:nT	196, 196
_UWMad_IfDefined:nnnnT	174, 174, 203, 244, 303, 368, 1159
_UWMad_IfUndefined:nT	196, 199
_UWMad_IfUndefined:nnnnT	174, 185, 206, 247, 306, 371
_UWMad_LicensePage_StartPage:	2123, 2144, 2304
_UWMad_ListOf_CurrentSectioningCommand:n	1278, 1282, 1298, 1316
_UWMad_Queue_IfDefined:nT	243, 243, 255, 260, 266, 275, 286, 294
_UWMad_Queue_IfUndefined:nT	243, 246, 250
_UWMad_Stack_IfDefined:nT	202, 202, 214, 219, 225, 234
_UWMad_Stack_IfUndefined:nT	202, 205, 209
_UWMad_ThrowWarnings:NTF	24
_UWMad_ThrowWarnings:TF	33, 45, 62, 67

A

\Abs	27, 1009
\abstract	13, 707, 734
\acknowledgments	13, 731
\Acro	19, 1749
\AcronymSetup	19, 1766
\addcontentsline	686, 1320
\AddToCounter	529, 533
\AddToLength	509, 513

\Adviser	4, 1906
\Advisor	4, 1899
\AllRightsReserved	2348
\appendix	14, 659, 660
\appendixname	652
\ArcCos	28, 1025
\ArcCosh	28, 1031
\ArcCot	28, 1030
\ArcCoth	28, 1036
\ArcCsc	28, 1029
\ArcCsch	28, 1035
\ArcSec	28, 1028
\ArcSech	28, 1034
\ArcSin	28, 1026
\ArcSinh	28, 1032
\ArcTan	28, 1027
\ArcTanh	28, 1033
\ArgMax	27, 1008
\ArgMin	27, 1007
\Attribution	10, 2356
\Author	2, 1835
\author	1837

C

\c_UWMad_Class_Date_tl	13, 20, 114
\c_UWMad_Class_Description_tl	14, 21
\c_UWMad_Class_Name_tl	11, 20, 108
\c_UWMad_Class_Version_tl	12, 21, 111
\c_UWMad_NextSection_chapter_tl	841
\c_UWMad_NextSection_paragraph_tl	845
\c_UWMad_NextSection_part_tl	840
\c_UWMad_NextSection_section_tl	842
\c_UWMad_NextSection_subsection_tl	843
\c_UWMad_NextSection_subsubsection_tl	844
\c_UWMad_PreviousSection_chapter_tl	846
\c_UWMad_PreviousSection_paragraph_tl	850
\c_UWMad_PreviousSection_section_tl	847
\c_UWMad_PreviousSection_subparagraph_tl	851
\c_UWMad_PreviousSection_subsection_tl	848
\c_UWMad_PreviousSection_subsubsection_tl	849

\g__UWMad_PDFMetaData_Subject_tl 1990, 1999, 2031
 \g__UWMad_ThesisInfo_AdvisorAssociation_tl
 1815, 1903, 1910, 1979
 \g__UWMad_ThesisInfo_AdvisorMarker_tl .. 1816, 1904, 1911, 1980
 \g__UWMad_ThesisInfo_AdvisorName_tl 1813, 1901, 1908, 1962, 1977
 \g__UWMad_ThesisInfo_AdvisorPosition_tl 1814, 1902, 1909, 1978
 \g__UWMad_ThesisInfo_Author_tl 1807, 1836, 1938, 2313
 \g__UWMad_ThesisInfo_DefenseDate_tl 1808, 1881, 1950
 \g__UWMad_ThesisInfo_Degree_tl 1811, 1849, 1944
 \g__UWMad_ThesisInfo_Department_tl 1809, 1897, 1959
 \g__UWMad_ThesisInfo_DocumentType_tl
 1812, 1818, 1853, 1856, 1861, 1866, 1871, 1876, 1947
 \g__UWMad_ThesisInfo_Institution_tl 1817, 1886, 1954
 \g__UWMad_ThesisInfo_IsSet_Advisor_bool 1825, 1900, 1907, 1976
 \g__UWMad_ThesisInfo_IsSet_Author_bool 1820, 1842
 \g__UWMad_ThesisInfo_IsSet_DefenseDate_bool 1821, 1882
 \g__UWMad_ThesisInfo_IsSet_Degree_bool 1823, 1850
 \g__UWMad_ThesisInfo_IsSet_Institution_bool 1824, 1887
 \g__UWMad_ThesisInfo_IsSet_Program_bool 1822, 1846
 \g__UWMad_ThesisInfo_IsSet_Title_bool 1819, 1833
 \g__UWMad_ThesisInfo_Program_tl 1810, 1845, 1941
 \g__UWMad_ThesisInfo_Title_tl 1806, 1827, 1935
 \g__UWMad_ThrowWarnings_bool 25, 34, 53, 54, 59
 \g__UWMad_TOC_Name_LOF_tl 751, 754, 788, 811
 \g__UWMad_TOC_Name_LOT_tl 750, 753, 773, 808
 \g__UWMad_TOC_Name_TOC_tl 749, 752, 757, 805
 \g__UWMad_Math_DelimiterDefaultLeft_tl 894, 895, 936, 950
 \g__UWMad_Math_DelimiterDefaultRight_tl 896, 897, 937, 958
 \g__UWMad_Math_derivSymbol_tl 888, 889, 903, 914, 925
 \g__UWMad_Math_pderivSymbol_tl 890, 891, 906, 917, 928
 \g__UWMad_Math_tderivSymbol_tl 892, 893, 909, 920, 931
 \g__UWMad_ParentClass_tl 22, 23, 49, 72, 75, 80, 84
 \GreaterThanThan 1119
 \Group 16, 1645

I

\IfBooleanFalseTF 479, 502
 \IfBooleanTrueTF 479, 495
 \IfCommandDoesNotExist 415
 \IfCommandDoesNotExistTF 422
 \IfCommandEmpty 437
 \IfCommandEmptyTF 437
 \IfCommandExists 415
 \IfCommandExistsTF 415
 \IfCounterTF 529, 545
 \IfEmptyTF 430, 717

\IfInfoIsSetT 1919, 2053, 2062, 2076, 2082, 2088, 2094
 \IfLengthTF 509, 522
 \IfMathModeTF 28, 1072, 1081, 1088, 1095
 \IfStringEmpty 429
 \Inf 25, 1003
 \Institution 3, 1885, 1889
 \int_gincr:cn 537
 \int_to_Alph:c 556
 \int_to_alph:c 559
 \int_to_arabic:c 553
 \int_to_Roman:c 562
 \int_to_roman:c 565

K

\Keywords 4, 2005

L

\l__UWMad_CCLicense_http_tl 2160, 2204
 \l__UWMad_CCLicense_IsValid_bool 2169, 2275, 2403
 \l__UWMad_CCLicense_Porting_tl 2152, 2181, 2213, 2375
 \l__UWMad_CCLicense_TypeAbbreviation_tl
 2154, 2193, 2223, 2234, 2245, 2256, 2272, 2280
 \l__UWMad_CCLicense_TypeWords_tl
 2155, 2211, 2226, 2237, 2248, 2259
 \l__UWMad_CCLicense_URL_Back_tl 2158, 2195, 2202, 2381
 \l__UWMad_CCLicense_URL_Front_tl 2156, 2189, 2200, 2379
 \l__UWMad_CCLicense_URL_Middle_tl 2157, 2192, 2201, 2380
 \l__UWMad_CCLicense_URL_tl 2159, 2198, 2295
 \l__UWMad_CCLicense_URLText_tl 2161, 2209, 2296, 2385
 \l__UWMad_CCLicense_UseAttribution_bool 2165, 2170, 2221, 2357
 \l__UWMad_CCLicense_UseCreativeCommons_bool
 2164, 2354, 2391, 2400
 \l__UWMad_CCLicense_UseNoDerivatives_bool .. 2167, 2254, 2366
 \l__UWMad_CCLicense_UseNonCommercial_bool .. 2168, 2232, 2360
 \l__UWMad_CCLicense_UseShareAlike_bool 2166, 2243, 2363
 \l__UWMad_CCLicense_Version_tl ... 2153, 2184, 2196, 2212, 2371
 \l__UWMad_Copyright_UseCopyright_bool .. 2131, 2346, 2392, 2408
 \l__UWMad_LicensePage_Owner_tl 2139, 2299, 2310, 2312, 2326, 2331
 \l__UWMad_LicensePage_Year_tl 2137, 2300, 2309, 2315, 2335, 2340
 \l__UWMad_Nomenclature_Entry_MarginBottom_dim
 1341, 1355, 1513, 1541
 \l__UWMad_Nomenclature_Entry_MarginLeft_dim
 1340, 1353, 1450, 1471, 1509, 1535
 \l__UWMad_Nomenclature_Entry_Padding_dim
 1342, 1357, 1452, 1474, 1511, 1537, 1539

\l__UWMad_Nomenclature_Entry_WidthDescription_dim	\MastersThesis	3, 1865
.....	1337, 1448, 1469, 1503, 1529	\Max	26, 1006
\l__UWMad_Nomenclature_Entry_WidthSymbol_dim	\Min	26, 1005
.....	1335, 1443, 1451, 1459, 1472, 1499, 1521			
\l__UWMad_Nomenclature_Entry_WidthUnit_dim	..	N		
.....	1464, 1473, 1525	\NoDerivs	10, 2365
\l__UWMad_Nomenclature_Entry_WidthUnits_dim	\noexpand	1797, 1798, 1800
.....	1339, 1351	\NomenclatureSetup	18, 1560, 1766
\l__UWMad_Nomenclature_Section_Group_tl	\NonCommercial	10, 2359
.....	1344, 1369, 1600, 1601, 1614, 1623			
\l__UWMad_Nomenclature_Section_Main_tl	O		
.....	1343, 1365, 1575, 1603, 1621	\onehalfspacing	624
\l__UWMad_Nomenclature_Section_Subgroup_tl	\oneo	29, 1106
.....	1345, 1373, 1611, 1612, 1625	\OneOver	29, 1103, 1107
\l__UWMad_Nomenclature_Title_Group_tl	..			
.....	1347, 1382, 1648, 1651	P		
\l__UWMad_Nomenclature_Title_Main_tl	...	\pderiv	22, 976
.....	1346, 1378, 1583, 1586	\pderivbig	23, 990
\l__UWMad_Nomenclature_Title_Subgroup_tl	\pderivSymbol	23, 905, 978, 992
.....	1348, 1386, 1659, 1662	\pderivSymbolChange	24, 916
\l__UWMad_Nomenclature_TitleSkip_dim	\pderivSymbolChangeDefault	24, 927
.....	1338, 1350	\preface	13, 740
\l__UWMad_Nomenclature_WidestSymbol_dim	\Prelim	3, 1875
.....	1333, 1420, 1430, 1444, 1460	\PrintCommitteeMemberList	1975, 2115
\l__UWMad_Nomenclature_WidestUnit_dim	..	\Producer	4, 2012
.....	1334, 1425, 1433, 1465	\Program	2, 1844
\l__UWMad_ThesisInfo_CommitteeCount_int			
.....	1967, 1970	R		
\l_tmpc_coffin	\ReMakeCommand	444, 452
.....	1493, 1528, 1540	\ResetColumnWidth	1686
\l_UWMad_Math_DelimiterLeft_tl			
.....	898, 949, 952, 966	S		
\l_UWMad_Math_DelimiterRight_tl	\Sec	27, 1016
.....	899, 957, 960, 968	\Sech	27, 1022
\LARGE	\SetBooleanFalse	479, 492
.....	2055	\SetBooleanTrue	479, 489
\LaTeX	\SetCounter	529, 539
.....	633	\SetLength	509, 516
\LessThan	\ShareAlike	10, 2362
.....	1118	\Sin	27, 1014
\LicenseOwner	\singlespacing	618
.....	7, 2325	\Sinh	27, 1020
\LicenseYear	\Sqrt	28, 1067
.....	7, 2334	\StepCounter	529, 536
\Lim	\Subgroup	16, 1656
.....	26, 1004	\Subject	4, 1998
\listfigurename	\subs	29, 1080
.....	788, 797			
\ListOfFigures			
.....	14, 803			
\ListOfFiguresDefault			
.....	745, 793			
\ListOfFiguresName			
.....	14, 810			
\ListOfTables			
.....	14, 802			
\ListOfTablesDefault			
.....	744, 778			
\ListOfTablesName			
.....	14, 807			
\listtablename			
.....	773, 782			
\Ln			
.....	27, 1010			
\Log			
.....	27, 1011			
M				
\MakeCommand			
.....	444, 444			
\MakeCommandUndefined			
.....	463, 463			
\MakeGlobalCommand			
.....	460, 460			
\MakeTitlePage			
.....	2100			

\subsup_s 29, 1094
 \Sup 25, 1002
 \sup_s 29, 1087

T

\TableOfContents 14, 801
 \TableOfContentsDefault 743, 763
 \TableOfContentsName 14, 804
 \Tan 27, 1015
 \Tanh 27, 1021
 \tderiv 22, 980
 \tderivbig 23, 994
 \tderivSymbol 23, 908, 982, 996
 \tderivSymbolChange 24, 919
 \tderivSymbolChangeDefault 24, 930
 \texttt 108
 \TheAdvisor 5, 1961
 \TheAuthor 5, 1937, 2066
 \TheCreator 5, 2039
 \TheDefenceDate 5, 1952
 \TheDefenseDate 5, 1949, 1952, 2095
 \TheDegree 5, 1943, 2077
 \TheDepartment 1958
 \TheDocumentType 5, 1946, 2047
 \TheInstitution 5, 1953, 1956, 2089
 \TheKeywords 5, 2033
 \TheLicenseOwner 2330
 \TheLicenseYear 2339
 \TheProducer 5, 2036
 \TheProgram 5, 1940, 2083
 \Thesis 3, 1870
 \TheSubject 5, 2030
 \TheTitle 5, 1934, 2056
 \TheUniversity 1956
 \Title 2, 1826
 \TitlePageAuthor 2061, 2105
 \TitlePageDefenseDate 2093, 2114
 \TitlePageDegree 2075, 2108
 \TitlePageFulFillment 2071, 2107
 \TitlePageInstitution 2087, 2111
 \TitlePageProgram 2081, 2109
 \TitlePageTitle 2052, 2104
 \tl_gset_eq:NN 757
 \tl_if_blank:fTF 431
 \tl_set_eq:Nc 712
 \tprime 30, 1115

U

\umiabstract 13, 737
 \University 3, 1889
 \UWMad_Definition_Reset:N 166, 173
 \UWMad_Definition_Reset:nn 158
 \UWMad_Definition_Swap:Nn 158, 172
 \UWMad_Definition_Swap:nn 158
 \UWMad_Deque_Clear:n 313, 313, 318
 \UWMad_Deque_Define:n 308, 308
 \UWMad_Deque_Delete:n 318
 \UWMad_Deque_PopLeft:n 334
 \UWMad_Deque_PopRight:n 340
 \UWMad_Deque_PushLeft:nn 324, 324, 334
 \UWMad_Deque_PushRight:nn 324, 329, 334
 \UWMad_Deque_WalkLeftToRight:nn 346, 346
 \UWMad_Deque_WalkRightToLeft:nn 346, 352
 \UWMad_Hash_Define:n 373, 373, 1139, 1710, 1711
 \UWMad_Hash_Delete:n 409, 409, 1156
 \UWMad_Hash_Get:nn 383, 384, 1268, 1270, 1273, 1276, 1753, 1756, 1758
 \UWMad_Hash_IfKeySet:nnTF 394, 395, 1750
 \UWMad_Hash_Set:nnn 378, 378, 1265, 1718, 1719, 1730, 1731
 \UWMad_Hash_Unset:nn 389, 389
 \UWMad_Hash_Walk:nn 404, 404
 \UWMad_Hook_Append:Nn 152
 \UWMad_Hook_Append:nn 134, 140
 \UWMad_Hook_Prepnd:Nn 146, 618, 624, 638, 644
 \UWMad_Hook_Prepnd:nn 134, 134
 \UWMad_IfSectionExists:nT 823, 853, 858, 863, 868, 1574
 \UWMad_LevelToSection:n 857
 \UWMad_ListOf_Define:n 1121, 1121, 1568, 1968
 \UWMad_ListOf_Delete:n 1141, 1141, 1699
 \UWMad_ListOf_DoNotIncludeInTOC:n 1185, 1190, 1632
 \UWMad_ListOf_GetSection_Group:n 72, 1254
 \UWMad_ListOf_GetSection_Main:n 72, 1249
 \UWMad_ListOf_GetSection_Subgroup:n 72, 1259
 \UWMad_ListOf_GetTitle_Group:n 1224
 \UWMad_ListOf_GetTitle_Group:nn 71
 \UWMad_ListOf_GetTitle_Main:n 1219
 \UWMad_ListOf_GetTitle_Main:nn 71
 \UWMad_ListOf_GetTitle_Subgroup:n 1229
 \UWMad_ListOf_GetTitle_Subgroup:nn 71
 \UWMad_ListOf_IfDefined:nT 1158,
 1158, 1167, 1172, 1177, 1186, 1191, 1196, 1205, 1210,
 1215, 1220, 1225, 1230, 1235, 1240, 1245, 1250, 1255, 1260
 \UWMad_ListOf_IfIncludeInTOC:n 1195

\UWMad_ListOf_IfIncludeInTOC:nTF	1195, 1288, 1303, 1319	\UWMad_Nomenclature_UpdateWidest_Units:n	78, 1423
\UWMad_ListOf_IfNumbered:nTF	1176, 1176, 1285	\UWMad_Nomenclature_ZeroWidest_Symbol:	1429, 1640, 1687
\UWMad_ListOf_IncludeInTOC:n	1185, 1185, 1630	\UWMad_Nomenclature_ZeroWidest_Unit:	1432, 1688
\UWMad_ListOf_MakeNotNumbered:n	1166, 1171, 1592	\UWMad_PreviousSection:n	867
\UWMad_ListOf_MakeNumbered:n	1166, 1166, 1590	\UWMad_Queue_Clear:n	254, 254, 1275
\UWMad_ListOf_PrintEntries:n	73, 1272, 1330, 1698, 1983	\UWMad_Queue_Define:n	249, 249, 1138
\UWMad_ListOf_PrintTitle:nn	74, 1279, 1331, 1693	\UWMad_Queue_Delete:n	259, 259, 1155
\UWMad_ListOf_PushEntry:nn	73, 1267, 1669, 1677, 1722, 1734, 1971	\UWMad_Queue_IfEmpty:nTF	293, 293
\UWMad_ListOf_SetHook:nnn	73, 1264, 1636, 1639	\UWMad_Queue_Pop:n	274, 274
\UWMad_ListOf_SetSection_Group:nn	72, 1239, 1622	\UWMad_Queue_Pop:nn	265
\UWMad_ListOf_SetSection_Main:nn	72, 1234, 1620	\UWMad_Queue_Push:nn	265, 1269
\UWMad_ListOf_SetSection_Subgroup:nn	72, 1244, 1624	\UWMad_Queue_Walk:nn	285, 285, 1274
\UWMad_ListOf_SetTitle_Group:nn	71, 1209, 1650	\UWMad_SectionToLevel:n	852
\UWMad_ListOf_SetTitle_Main:nn	71, 1204, 1585	\UWMad_Stack_Clear:n	213, 213
\UWMad_ListOf_SetTitle_Subgroup:nn	71, 1214, 1661	\UWMad_Stack_Define:n	208, 208
\UWMad_ListOf_StartGroup:nn	75, 1329, 1652, 1663	\UWMad_Stack_Delete:n	218, 218
\UWMad_Math_RootWithTail:nn	1045, 1068	\UWMad_Stack_Pop:n	233, 233
\UWMad_MetaData_PushToList:nn		\UWMad_Stack_Push:nn	224, 224
	1771, 1829, 1838, 2000, 2007, 2014, 2021	\UWMad_Stack_Walk:nn	240, 240
\UWMad_NextSection:n	862, 1602, 1613	\UWMadClass	107
\UWMad_Nomenclature_SetEntry:	1545	\UWMadClassDate	113
\UWMad_Nomenclature_SetEntry_NoUnits:	1549	\UWMadClassVersion	110
\UWMad_Nomenclature_SetEntry_NoUnits:nn	1495, 1678, 1724, 1736	\UWMadLong	116
\UWMad_Nomenclature_SetEntry_Units:	1547	\UWMadShort	119
\UWMad_Nomenclature_SetEntry_Units:nnn	1517, 1670		
\UWMad_Nomenclature_SetEntryWidths:	1478, 1637	V	
\UWMad_Nomenclature_SetEntryWidths_NoUnits:	1440, 1482	\ValueOfCounter	529, 542
\UWMad_Nomenclature_SetEntryWidths_Units:	1456, 1480	\ValueOfLength	509, 519
\UWMad_Nomenclature_UpdateWidest:Nn	78, 1411, 1419, 1424		
\UWMad_Nomenclature_UpdateWidest_Symbol:n		X	
	78, 1418, 1673, 1681, 1741	\xetex_if_engine_p:	102