`UWMadThesis` Class Manual

Troy Christopher Haskin

Table of Contents

Part I

User Guide

The `UWMadThesis` class is aimed at providing a LaTeX 2_ε 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.

`\Title`
`\Author`
`\Program`
`\Degree`

`\Title` {⟨*title*⟩}
`\Author` {⟨*author name*⟩}
`\Program` {⟨*program*⟩}
`\Degree` {⟨*degree*⟩}

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.

`\DefenseDate`
`\DefenceDate`

`\DefenseDate` {⟨*defense date*⟩}
`\DefenceDate` {⟨*defense date*⟩}

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

Since {⟨*defense date*⟩} has no parsing performed on it, it may be entered any which way and will be typeset as-entered.

`\Institution` `\Institution {⟨institution name⟩}`
`\University` `\University {⟨institution name⟩}`

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

`\CommitteeMember` `\CommitteeMember {⟨member name⟩} {⟨member position⟩}`

`\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.

`\DocumentType` `\DocumentType {⟨document type⟩}`

By default, the `\MakeTitlePage` command prints the phrase "A `{⟨document type⟩}` submitted in partial fulfillment of the requirements for the degree of'' on the title page". The default `{⟨document type⟩}` 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.

`\Dissertation` `\Dissertation`
`\DoctoralThesis` `\DoctoralThesis`
`\MastersThesis` `\MastersThesis`
`\Thesis` `\Thesis`
`\Prelim` `\Prelim`

These commands set the value of `{⟨document type⟩}` to a value similar to their command name:

- `\Dissertation` sets `{⟨document type⟩}` to "dissertation"
- `\DoctoralThesis` sets `{⟨document type⟩}` to "doctoral thesis"
- `\MastersThesis` sets `{⟨document type⟩}` to "master's thesis"
- `\Thesis` sets `{⟨document type⟩}` to "thesis"
- `\Prelim` sets `{⟨document type⟩}` to "preliminary report"

\Advisor
\Adviser

```
\Advisor {⟨advisor name⟩} {⟨advisor position⟩}
\Adviser {⟨advisor name⟩} {⟨advisor position⟩}
```

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.

\Subject
\Keywords

```
\Subject  {⟨document subject⟩}
\Keywords {⟨list of keywords⟩}
```

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.

\Producer
\Creator

```
\Producer {⟨pdf producer⟩}
\Creator  {⟨pdf creator⟩}
```

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 TeX/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 LaTeX 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:

`\LicenseOwner`	`\LicenseOwner {`⟨*owner name*⟩`}`
`\LicenseYear`	`\LicenseYear {`⟨*year*⟩`}`

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:

> 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.

\CreativeCommons

\CreativeCommons

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''.

\Attribution
\ShareAlike
\NonCommercial
\NoDerivs

\Attribution
\ShareAlike
\NonCommercial
\NoDerivs

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.

\CCVersion
\CCPorting
\CCURL
\CCURLText

\CCVersion{⟨*CC version*⟩}
\CCPorting{⟨*CC porting*⟩}
\CCURL {⟨*CC link*⟩}
\CCURLText{⟨*CC link text*⟩}

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):

Feature Set 3

Layout And Style

The `UWMadThesis` class class has several default styling differences from the standard LaTeX 2_ε 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 LaTeX 2_ε are `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, and `subparagraph`. The `UWMadThesis` class 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.

`\dedications`
`\acknowledgments`
`\abstract`
`\umiabstract`
`\preface`

```
\dedications      {⟨title⟩}
\acknowledgments  {⟨title⟩}
\abstract         {⟨title⟩}
\umiabstract      {⟨title⟩}
\preface          {⟨title⟩}
```

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 class takes a different approach to encourage good semantic mark-up in LaTeX documents and, therefore, redefines `\appendix`.

\appendix

\appendix [⟨*short title*⟩]{⟨*title*⟩}
\appendix*[⟨*short title*⟩]{⟨*title*⟩}

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.

\TableOfContentsName
\ListOfTablesName
\ListOfFiguresName

\TableOfContentsName{⟨*toc title*⟩}
\ListOfTablesName {⟨*lot title*⟩}
\ListOfFiguresName {⟨*lof title*⟩}

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”.

\TableOfContents
\ListOfTables
\ListOfFigures

\TableOfContents
\ListOfTables
\ListOfFigures

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}[`⟨*section*⟩`]{`⟨*title*⟩`}`

```
\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.

`\Entry`

```
\Entry{⟨symbol⟩}{⟨description⟩}
\Entry{⟨symbol⟩}{⟨units⟩}{⟨description⟩}
```

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).

`\Group` `\Subgroup`

```
\Group{⟨group title⟩}
\Subgroup{⟨subgroup title⟩}
```

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-separate list of key-value pairs to the function `\NomenclatureSetup`

`\NomenclatureSetup` `\NomenclatureSetup{⟨key-value CSV⟩}`

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`.

`\Acro` `\Acro{`⟨*acronym*⟩`}`

`\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 `\Arco{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.

`\AcronymSetup` `\AcronymSetup{`⟨*key-value CSV*⟩`}`

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 `\Group` command	section	section
subgroup-section	Section level for `\Subgroup` command	subsection	section
main-title	Title for the nomenclature	Nomenclature	text
group-title	Title for the `\Group` command	---	text
subgroup-title	Title for the `\Subgroup` 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.

\deriv
\pderiv
\tderiv

`\deriv  {`⟨*function*⟩`} {`⟨*variable*⟩`} {`⟨*order*⟩`}`
`\pderiv {`⟨*function*⟩`} {`⟨*variable*⟩`} {`⟨*order*⟩`}`
`\tderiv {`⟨*function*⟩`} {`⟨*variable*⟩`} {`⟨*order*⟩`}`

This function set is meant to typeset three different kinds of derivatives: ordinary, partial, and total (i.e., material or Lagragian). 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

$$\frac{\mathrm{d}^2 y}{\mathrm{d}x^2} + \frac{\mathrm{d}y}{\mathrm{d}x} + y(x) = 0 \tag{1}$$

$$\frac{\partial T}{\partial t} - \alpha \frac{\partial^2 T}{\partial z^2} = 0 \tag{2}$$

$$\frac{\mathrm{D}(\rho u)}{\mathrm{D}t} + \frac{\partial P}{\partial z} - \rho g = 0 \tag{3}$$

\derivbig
\pderivbig
\tderivbig

```
\derivbig   [⟨left delim⟩] {⟨function⟩} [⟨right delim⟩] {⟨variable⟩} {⟨order⟩}
\pderivbig  [⟨left delim⟩] {⟨function⟩} [⟨right delim⟩] {⟨variable⟩} {⟨order⟩}
\tderivbig  [⟨left delim⟩] {⟨function⟩} [⟨right delim⟩] {⟨variable⟩} {⟨order⟩}
```

This function set is identical to the non-`big` versions above, except that {⟨*function*⟩} 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} +
            q(x) (1 - \lambda) y(x)  &= 0 \\[0.50em]
    \tderivbig{ \rho{i} + \frac{1}{2} \rho u^2 }[(]{t} -
            \pderivbig[\lvert]{ \kappa \pderiv{T}{z} }{z} &= 0
\end{align}
```

and is typeset as

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

$$\frac{\mathrm{D}}{\mathrm{D}t}\left[\rho i + \frac{1}{2}\rho u^2\left(- \frac{\partial}{\partial z}\middle|\kappa\frac{\partial T}{\partial z}\right]\right. = 0 \tag{5}$$

\DerivativeGeneral
\DerivativeGeneralBig

```
\DerivativeGeneral     {⟨function⟩} {⟨variable⟩} {⟨order⟩} {⟨symbol⟩}
\DerivativeGeneralBig  {⟨function⟩} {⟨variable⟩} {⟨order⟩} {⟨symbol⟩} {⟨left delim⟩} {⟨right
delim⟩}
```

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 desire, these commands are available for usage.

\derivSymbol
\pderivSymbol
\tderivSymbol

```
\derivSymbol
```

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

\derivSymbolChange
\pderivSymbolChange
\tderivSymbolChange

`\derivSymbolChange {⟨symbol⟩}`

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 TEX group where the switch was made (more often than not for LATEX users, this means ``upon exiting an environment''). For example:

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

typesets as

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \frac{\delta Q}{\delta t} - \frac{\delta W}{\delta t} \tag{6}$$

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

\derivSymbolChangeDefault
\pderivSymbolChangeDefault
\tderivSymbolChangeDefault

`\derivSymbolChangeDefault {⟨symbol⟩}`
`\pderivSymbolChangeDefault {⟨symbol⟩}`
`\tderivSymbolChangeDefault {⟨symbol⟩}`

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

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

typesets as

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \frac{\delta Q}{\delta t} - \frac{\delta W}{\delta t} \tag{7}$$

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

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

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 Infinum 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

$$\operatorname*{Inf}_{x\in\mathbb{R}}\{0 < x < 1\} = 0 \tag{9}$$

$$\operatorname*{Sup}_{x\in\mathbb{R}}\{0 < x < 1\} = 1 \tag{10}$$

`\Lim`

The limit operator:

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

is typeset as

$$\Lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = \mathrm{e} \tag{11}$$

`\Min`
`\Max`

The maximum and minimum value operators

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

is typeset as

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

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

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

is typeset as

$$\operatorname*{ArgMin}_{x \in \mathbb{R}} \operatorname{Sin}(x) = \frac{3\pi}{2} + 2\pi n \tag{14}$$

$$\operatorname*{ArgMax}_{x \in \mathbb{R}} \operatorname{Sin}(x) = \frac{\pi}{2} + 2\pi n \tag{15}$$

`\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.

`\ArcCos`
`\ArcSin`
`\ArcTan`
`\ArcSec`
`\ArcCsc`
`\ArcCot`

Standard inverse trigonometric functions and their reciprocals.

`\ArcCosh`
`\ArcSinh`
`\ArcTanh`
`\ArcSech`
`\ArcCsch`
`\ArcCoth`

Hyperbolic inverse trigonometric functions and their reciprocals.

6.3 Miscellaneous Commands

`\Sqrt` `\Sqrt [⟨n⟩] {⟨argument⟩}`

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:

$$\sqrt[3]{\frac{f(x)}{g(x)}} = \sqrt[3]{\frac{f(x)}{g(x)}} \tag{16}$$

`\IfMathModeTF` `\IfMathModeTF {⟨math mode code⟩} {⟨text mode code⟩}`

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

\subs
\sups
\subsups

`\subs [⟨space⟩] {⟨text subscript⟩}`

`\sups [⟨space⟩] {⟨text superscript⟩}`

`\subsups [⟨subscript space⟩] {⟨text subscript⟩} [⟨superscript space⟩] {⟨text superscript⟩}`

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 `$_{\mathrm{P}}$, T_{P}, T_P', and the input `` `T\subs[\!]{P}, T\subs[\:]{P}' `` is typeset as `$_{\mathrm{P}}$, $_{\mathrm{P}}$'. T^{P}

\OneOver
\oneo

`\OneOver {⟨denominator⟩}`

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

$$\frac{1}{\sqrt{x^2+1}} \tag{17}$$

\dd

`\dd {⟨variable⟩}`

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

$$f(b) - f(a) = \int_a^b \frac{\mathrm{d}f}{\mathrm{d}t}\mathrm{d}t \tag{18}$$

`\dprime` `\tprime`

`\dprime`

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

$$q' = q''2\pi R = q'''\pi R^2 \tag{19}$$

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 LaTeX3 Programming Layer; this will be denoted as `expl3`. The `expl3` is the first piece of a new system designed to succeed LaTeX 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 LaTeX 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 TeX, 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.

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


\ExplSyntaxOn
```

1.2 Identification and Defaults

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

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

```
    LaTeX2e+~Thesis~Class~for~UW-Madison
}
\tl_const:Nn \c__UWMad_UniversityLong_tl   {University~of~Wisconsin--Madison}
\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.

```
\ProvidesExplClass
    {\c__UWMad_Class_Name_tl}   {\c__UWMad_Class_Date_tl}
    {\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 LaTeX base class `report`.

```
\tl_new:N   \g_UWMad_ParentClass_tl
\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.

```
\cs_new:Nn \__UWMad_ThrowWarnings:NTF {
    \bool_if:NTF \g__UWMad_ThrowWarnings_bool {
        \bool_if:NTF #1 {
            #2
        } {
            #3
        }
    } {}
}
\cs_new:Nn \__UWMad_ThrowWarnings:TF {
    \bool_if:NTF \g__UWMad_ThrowWarnings_bool {
        #1
    } {
        #2
    }
}
```

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

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

```
    Option~'#1'~violates~\c_UWMadUniversityShort_tl{}~
    Dissertation~Guidelines;~consider~omission
}
\cs_new:Nn \__UWMad_FrontMatter_StyleWarning:n {
    \__UWMad_ThrowWarnings:TF {
    \msg_warning:nnn {UWMadThesis}{Options/StyleViolation}
        {#1}
    } { }
   \PassOptionsToClass{#1}{\g_UWMad_ParentClass_tl}
}
```

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

```
\bool_new:N       \g__UWMad_MathTweaks_bool
\bool_gset_true:N \g__UWMad_MathTweaks_bool
\bool_new:N       \g__UWMad_ThrowWarnings_bool
\bool_gset_true:N \g__UWMad_ThrowWarnings_bool
```

Declare the options.

```
\DeclareOption{NoMath} {
    \bool_gset_false:N \g__UWMad_MathTweaks_bool
}
\DeclareOption{Quiet} {
    \bool_gset_false:N \g__UWMad_ThrowWarnings_bool
}
```

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

```
\DeclareOption{a4paper} {
    \__UWMad_ThrowWarnings:TF {
        \__UWMad_FrontMatter_StyleWarning:n {\CurrentOption}
    } { }
}
\DeclareOption{twoside} {
    \__UWMad_ThrowWarnings:TF {
        \__UWMad_FrontMatter_StyleWarning:n {\CurrentOption}
    } { }
}
```

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

```
\DeclareOption{article} {
    \tl_gset:Nn \g_UWMad_ParentClass_tl {article}
}
```

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.

```
\DeclareOption{l3doc} {
    \tl_gset:Nn \g_UWMad_ParentClass_tl {l3doc}
    \tl_const:cn {ver@thumbpdf.sty} {}
}
```

Pass all remaining options to the base class.

```
\DeclareOption*{
    \PassOptionsToClass
        {\CurrentOption}{\g_UWMad_ParentClass_tl}
}
```

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

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

1.4 Package Loads

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

```
\RequirePackage{xparse}
\RequirePackage{fixltx2e}
\RequirePackage{microtype}
\RequirePackage{array}
\RequirePackage{graphicx}
\RequirePackage{setspace}
\RequirePackage{geometry}
```

Load the $\mathcal{AMS}$ suite.

```
\RequirePackage{amsmath}
\RequirePackage{amsfonts}
\RequirePackage{amssymb}
\RequirePackage{mathtools}
```

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

```
\RequirePackage[noabbrev,nameinlink]{cleveref}
\RequirePackage[usenames,dvipsnames,svgnames,table,hyperref]{xcolor}
\RequirePackage{caption}
\RequirePackage{subcaption}
```

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

```
\RequirePackage{hyperref}
\RequirePackage{bookmark}
```

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

```
\bool_if:nTF {\xetex_if_engine_p: || \luatex_if_engine_p:} {
    \RequirePackage{fontspec}
    \setmainfont
        [SmallCapsFont = {Latin~Modern~Roman~Caps}]
        {Latin~Modern~Roman}
%
    \RequirePackage{polyglossia}
    \setmainlanguage[variant = usmax]{english}
} {
    \RequirePackage[T1]{fontenc}
    \RequirePackage{lmodern}
%
    \RequirePackage[english]{babel}
}
```

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

```
\DeclareDocumentCommand \UWMadClass { } {
    \texttt{\c__UWMad_Class_Name_tl}~class
}
\DeclareDocumentCommand \UWMadClassVersion { } {
    \c__UWMad_Class_Version_tl
}
\DeclareDocumentCommand \UWMadClassDate { } {
    \c__UWMad_Class_Date_tl
}
\DeclareDocumentCommand \UWMadLong { } {
    \c__UWMad_UniversityLong_tl
}
\DeclareDocumentCommand \UWMadShort { } {
    \c__UWMad_UniversityShort_tl
}
```

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 LaTeX 2_ε abstractions for several other features.

2.1 Utility Commands

Define some messages for the rest of the module.

```
\msg_new:nnn {UWMadThesis} {Programming/UnregisteredVariable} {
    `#1'~is~not~a~registered~#2.~~The~#2~must~be~defined~
    before~usage~by~the~function~\string\UWMad_#2_DefineLocal:n~or~
    \string\UWMad_#2_DefineGlobal:n.
}
\msg_new:nnn {UWMadThesis} {Programming/Undefined} {
    The~#2~`#1'~is~undefined.~~The~#2~must~be~defined~
    before~usage~by~the~function~\string\UWMad_#2_Define:n.
}
\msg_new:nnn {UWMadThesis} {Programming/Defined} {
    The~#2~`#1'~is~already~defined~and~will~not~altered.
}
```

`\UWMad_Hook_Prepend:nn`
`\UWMad_Hook_Append:nn`

These commands allow additional code to be prepended or appended to a specified command.

```
\cs_new:Nn \UWMad_Hook_Prepend:nn{
    \cs_new_eq:cc  {#1-Default:} {#1}
    \cs_gset:cn    {#1:}         {#2 \cs:w #1-Default:\cs_end:}
    \cs_undefine:c {#1}
    \cs_new_eq:cc  {#1}          {#1:}
}
\cs_new:Nn \UWMad_Hook_Append:nn{
    \cs_new_eq:cc  {#1-Default:} {#1}
    \cs_gset:cn    {#1:}        {\cs:w #1-Default:\cs_end: #2}
    \cs_undefine:c {#1}
    \cs_new_eq:cc  {#1}          {#1:}
}
\cs_new:Nn \UWMad_Hook_Prepend:Nn{
    \cs_new_eq:cN  {\string#1-Default:} #1
```

```
    \cs_gset:cn    {\string#1:}         {#2 \cs:w\string#1-Default:\cs_end:}
    \cs_undefine:N  #1
    \cs_new_eq:Nc   #1          {\string#1:}
}
\cs_new:Nn \UWMad_Hook_Append:Nn{
    \cs_new_eq:cN  {\string#1-Default:} #1
    \cs_gset:cn    {\string#1:}         {\cs:w\string#1-Default:\cs_end: #2}
    \cs_undefine:N  #1
    \cs_new_eq:Nc   #1          {\string#1:}
}
```

`\UWMad_Definition_Swap:nn`
`\UWMad_Definition_Reset:nn`

These commands "swap" in a new definition of a command and, when called, reset it to it's default definition.

```
\cs_new:Nn \UWMad_Definition_Swap:Nn {
    \cs_if_exist:NTF #1 {
        \cs_new_eq:cN  {\string#1-Default:}  #1
        \cs_gset_eq:Nc  #1 {#2}
    } {
        \cs_new:Nn #1 {#2}
    }
}
\cs_new:Nn \UWMad_Definition_Reset:N {
    \cs_if_exist:cTF {\string#1-Default:} {
        \cs_gset_eq:Nc  #1              {\string#1-Default:}
        \cs_undefine:c  {\string#1-Default:}
    } { }
}
\cs_generate_variant:Nn \UWMad_Definition_Swap:Nn {cn}
\cs_generate_variant:Nn \UWMad_Definition_Reset:N {c}
```

`\__UWMad_IfDefined:nnnnT`
`\__UWMad_IfUndefined:nnnnT`

These commands accept a {⟨*Prefix*⟩}, an {⟨*ID*⟩}, a {⟨*Suffix*⟩}, a {⟨*Type*⟩}, and {⟨*Code*⟩}. It determines if a command named by the concatenation of {⟨*Prefix*⟩}, {⟨*ID*⟩}, and {⟨*Suffix*⟩} is defined or not and executes {⟨*Code*⟩} depending on the existence.

Usage:

`\__UWMad_IfUndefined:nnnnT {⟨Prefix⟩}{⟨ID⟩}{⟨Suffix⟩}{⟨Type⟩}{⟨Code⟩}`

```
\cs_new:Nn \__UWMad_IfDefined:nnnnT{
    \cs_if_exist:cTF {#1#2#3} {
        #5
    }{
            \msg_error:nnnn
                {UWMadThesis}
                {Programming/Undefined}
                {#2}
                {#4}
    }
}
```

```
\cs_new:Nn \__UWMad_IfUndefined:nnnnT{
    \cs_if_free:cTF {#1#2#3} {
        #5
    }{
            \msg_warning:nnnn
                {UWMadThesis}
                {Programming/Defined}
                {#2}
                {#4}
    }
}
```

`\__UWMad_IfDefined:nT` `\__UWMad_IfUndefined:nT` These commands are simplifications of the above commands and that only take a `{`⟨*CommandName*⟩`}` and `{`⟨*TrueCode*⟩`}`.

Usage:

`\__UWMad_IfUndefined:nT {`⟨*CommandName*⟩`}{`⟨*TrueCode*⟩`}`

```
\cs_new:Nn \__UWMad_IfDefined:nT{
    \_UWMad_IfDefined:nnnnT{#1}{}{}{command}{#2}
}
\cs_new:Nn \__UWMad_IfUndefined:nT{
    \_UWMad_IfUndefined:nnnnT{#1}{}{}{command}{#2}
}
```

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), deques (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 LaTeX3 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`

```
\cs_new:Nn \__UWMad_Stack_IfDefined:nT {
    \__UWMad_IfDefined:nnnnT{g__UWMad_Stack_}{#1}{}{Stack}{#2}
}
\cs_new:Nn \__UWMad_Stack_IfUndefined:nT{
    \__UWMad_IfUndefined:nnnnT{g__UWMad_Stack_}{#1}{}{Stack}{#2}
}
```

`\UWMad_Stack_Define:n` Define a new Stack.

```
\cs_new:Nn \UWMad_Stack_Define:n {
    \__UWMad_Stack_IfUndefined:nT {#1} {
        \tl_new:c {g__UWMad_Stack_#1}
    }
}
```

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

```
\cs_new:Nn \UWMad_Stack_Clear:n {
    \__UWMad_Stack_IfDefined:nT {#1} {
        \tl_gclear:c   {g__UWMad_Stack_#1}
    }
}
```

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

```
\cs_new:Nn \UWMad_Stack_Delete:n {
    \__UWMad_Stack_IfDefined:nT {#1} {
        \tl_gclear:c   {g__UWMad_Stack_#1}
        \cs_undefine:c {g__UWMad_Stack_#1}
    }
}
```

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

```
\cs_new:Nn \UWMad_Stack_Push:nn {
    \__UWMad_Stack_IfDefined:nT {#1} {
        \tl_gput_left:cn {g__UWMad_Stack_#1} {#2}
    }
}
%
```

```
%
\cs_generate_variant:Nn \tl_head:N { c }
\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.

```
\cs_new:Nn \UWMad_Stack_Pop:n {
    \__UWMad_Stack_IfDefined:nT {#1} {
        \tl_set:Nf \l_tmpa_tl          {\tl_head:c {g__UWMad_Stack_#1}}
        \tl_set:cf {g__UWMad_Stack_#1} {\tl_tail:c {g__UWMad_Stack_#1}}
        \tl_use:N \l_tmpa_tl
    }
}
```

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

```
\cs_new:Nn \UWMad_Stack_Walk:nn {
    \tl_map_inline:cn {g__UWMad_Stack_#1} {#2}
}
```

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`

```
\cs_new:Nn \__UWMad_Queue_IfDefined:nT {
    \__UWMad_IfDefined:nnnnT{g__UWMad_Queue_}{#1}{}{Queue}{#2}
}
\cs_new:Nn \__UWMad_Queue_IfUndefined:nT{
    \__UWMad_IfUndefined:nnnnT{g__UWMad_Queue_}{#1}{}{Queue}{#2}
}
```

`\UWMad_Queue_Define:n` Define a new Queue.

```
\cs_new:Nn \UWMad_Queue_Define:n {
    \__UWMad_Queue_IfUndefined:nT {#1} {
        \tl_new:c {g__UWMad_Queue_#1}
    }
}
```

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

```
\cs_new:Nn \UWMad_Queue_Clear:n {
    \__UWMad_Queue_IfDefined:nT {#1} {
        \tl_gclear:c {g__UWMad_Queue_#1}
    }
}
```

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

```
\cs_new:Nn \UWMad_Queue_Delete:n {
    \__UWMad_Queue_IfDefined:nT {#1} {
        \tl_gclear:c    {g__UWMad_Queue_#1}
         \cs_undefine:c {g__UWMad_Queue_#1}
    }
}
```

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

```
\cs_new:Nn \UWMad_Queue_Push:nn {
    \__UWMad_Queue_IfDefined:nT {#1} {
        \tl_gput_left:cn {g__UWMad_Queue_#1} {{#2}}
    }
}
%
%
\cs_generate_variant:Nn \tl_head:N { c }
\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.

```
\cs_new:Nn \UWMad_Queue_Pop:n {
    \__UWMad_Queue_IfDefined:nT {#1} {
        \tl_reverse:c   {g__UWMad_Queue_#1}
        \tl_set:Nf \l_tmpa_tl
            {\tl_head:c {g__UWMad_Queue_#1}}
        \tl_set:cf      {g__UWMad_Queue_#1}
            {\tl_tail:c {g__UWMad_Queue_#1}}
        \tl_reverse:c   {g__UWMad_Queue_#1}
        \tl_use:N \l_tmpa_tl
    }
}
```

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

```
\cs_new:Nn \UWMad_Queue_Walk:nn {
    \__UWMad_Queue_IfDefined:nT {#1} {
        \group_begin:
            \tl_reverse:c     {g__UWMad_Queue_#1}
            \tl_map_inline:cn {g__UWMad_Queue_#1} {#2}
        \group_end:
    }
}
```

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

```
\cs_new:Nn \UWMad_Queue_IfEmpty:nTF {
    \__UWMad_Queue_IfDefined:nT {#1} {
        \tl_if_empty:cTF {g__UWMad_Queue_#1}{
            #2
        }{
            #3
        }
    }
}
```

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

```
\cs_new:Nn \__UWMad_Deque_IfDefined:nT {
    \__UWMad_IfDefined:nnnnT{g__UWMad_Deque_}{#1}{}{Deque}{#2}
}
\cs_new:Nn \__UWMad_Deque_IfUndefined:nT{
    \__UWMad_IfUndefined:nnnnT{g__UWMad_Deque_}{#1}{}{Deque}{#2}
}
```

`\UWMad_Deque_Define:n` Define a new Deque.

```
\cs_new:Nn \UWMad_Deque_Define:n {
    \__UWMad_Deque_IfUndefined:nT {#1} {
        \seq_new:c {g__UWMad_Deque_#1}
    }
}
```

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

```
\cs_new:Nn \UWMad_Deque_Clear:n {
    \__UWMad_Deque_IfDefined:nT {#1} {
        \seq_gclear:c  {g__UWMad_Deque_#1}
    }
}
```

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

```
\cs_new:Nn \UWMad_Deque_Delete:n {
    \__UWMad_Deque_IfDefined:nT {#1} {
        \seq_gclear:c  {g__UWMad_Deque_#1}
        \cs_undefine:c {g__UWMad_Deque_#1}
    }
}
```

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

```
\cs_new:Nn \UWMad_Deque_PushLeft:nn {
    \__UWMad_Deque_IfDefined:nT {#1} {
        \seq_gput_left:cn  {g__UWMad_Deque_#1} {#2}
    }
}
\cs_new:Nn \UWMad_Deque_PushRight:nn {
    \__UWMad_Deque_IfDefined:nT {#1} {
        \seq_gput_right:cn {g__UWMad_Deque_#1} {#2}
    }
}
```

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

```
\cs_new:Nn \UWMad_Deque_PopLeft:n {
    \__UWMad_Deque_IfDefined:nT {#1} {
        \seq_gpop_left:cN  {g__UWMad_Deque_#1} \l_tmpa_tl
        \tl_use:N \l_tmpa_tl
    }
}
\cs_new:Nn \UWMad_Deque_PopRight:n {
    \__UWMad_Deque_IfDefined:nT {#1} {
        \seq_gpop_right:cN {g__UWMad_Deque_#1} \l_tmpa_tl
        \tl_use:N \l_tmpa_tl
    }
}
```

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

```
\cs_new:Nn \UWMad_Deque_WalkLeftToRight:nn {
    \__UWMad_Deque_IfDefined:nT {#1} {
        \seq_map_inline:cn {g__UWMad_Deque_#1} {#2}
    }
}
\cs_generate_variant:Nn \seq_reverse:N {c}
\cs_new:Nn \UWMad_Deque_WalkRightToLeft:nn {
    \__UWMad_Deque_IfDefined:nT {#1} {
        \group_begin:
            \seq_reverse:c     {g__UWMad_Deque_#1}
            \seq_map_inline:cn {g__UWMad_Deque_#1} {#2}
        \group_end:
    }
}
```

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 usablility, 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 LaTeX3 programming.

```
\cs_generate_variant:Nn \prop_gput:Nnn   { c x n   }
\cs_generate_variant:Nn \prop_if_in:NnTF { c x TF  }
\cs_generate_variant:Nn \prop_if_in:NnTF { c f TF  }
\cs_generate_variant:Nn \prop_get:Nn     { c x     }
\cs_generate_variant:Nn \prop_get:Nn     { c f     }
\cs_generate_variant:Nn \prop_get:NnNTF  { c x N TF}
\cs_generate_variant:Nn \prop_gremove:Nn { c x     }
```

`\__UWMad_Hash_IfDefined:nT`
`\__UWMad_Hash_IfUndefined:nT`

Shortcuts for the more general commands outlined above.

```
\cs_new:Nn \__UWMad_Hash_IfDefined:nT {
    \__UWMad_IfDefined:nnnnT{g__UWMad_Hash_}{#1}{}{Hash}{#2}
}
\cs_new:Nn \__UWMad_Hash_IfUndefined:nT{
    \__UWMad_IfUndefined:nnnnT{g__UWMad_Hash_}{#1}{}{Hash}{#2}
```

```
}
```

`\UWMad_Hash_Define:n` Define a new Hash.

```
\cs_new:Nn \UWMad_Hash_Define:n {
    \__UWMad_Hash_IfUndefined:nT {#1} {
        \prop_new:c {g__UWMad_Hash_#1}
    }
}
```

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

Usage:

`\UWMad_Hash_Set:nnn{`⟨*HashID*⟩`}{`⟨*Key*⟩`}{`⟨*Value*⟩`}`

```
\cs_new:Nn \UWMad_Hash_Set:nnn {
    \__UWMad_Hash_IfDefined:nT {#1} {
        \prop_gput:cxn {g__UWMad_Hash_#1}{#2}{#3}
    }
}
```

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

```
\cs_generate_variant:Nn \prop_get:cn {cf}
\cs_new:Nn \UWMad_Hash_Get:nn {
    \__UWMad_Hash_IfDefined:nT {#1} {
        \prop_get:cf {g__UWMad_Hash_#1}{#2}
    }
}
```

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

```
\cs_new:Nn \UWMad_Hash_Unset:nn {
    \__UWMad_Hash_IfDefined:nT {#1} {
        \prop_gremove:cx {g__UWMad_Hash_#1} {#2}
    }
}
```

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

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

```
        \prop_if_in:cfTF {g__UWMad_Hash_#1} {\tl_trim_spaces:n{#2}} {
            #3
        }{
            #4
        }
    }
}
```

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

```
\cs_new:Nn \UWMad_Hash_Walk:nn {
    \__UWMad_Hash_IfDefined:nT {#1} {
        \prop_map_inline:cn {g__UWMad_Hash_#1} {#2}
    }
}
```

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

```
\cs_new:Nn \UWMad_Hash_Delete:n {
    \__UWMad_Hash_IfDefined:nT {#1} {
        \prop_gclear:c {g__UWMad_Hash_#1}
        \cs_undefine:c {g__UWMad_Hash_#1}
    }
}
```

2.3 User-Level Abstractions

The commands that follow are LaTeX 2ε-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` `\IfCommandDoesNotExist` This command pair is used instead of LaTeX's `\@ifundefined`. Since it is ε-TeX, this command will allow for a switch to `\@ifundefined` if problems arise from non-ε-TeX users in the future.

Usage:

`\IfCommandExists{`⟨*Command Name*⟩`}{`⟨*True*⟩`}{`⟨*False*⟩`}`

`\IfCommandDoesNotExist{`⟨*Command Name*⟩`}{`⟨*True*⟩`}{`⟨*False*⟩`}`

```
\DeclareDocumentCommand \IfCommandExistsTF { m +m +m } {
    \cs_if_exist:cTF {#1}{
        #2
    }{
        #3
    }
}
\DeclareDocumentCommand \IfCommandDoesNotExistTF { m +m +m } {
    \cs_if_free:cTF {#1}{
        #2
    }{
        #3
    }
}
```

`\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*⟩`}`

```
\cs_generate_variant:Nn \tl_if_blank:nTF {fTF}
\DeclareDocumentCommand \IfEmptyTF { m +m +m } {
    \tl_if_blank:fTF {#1}{
        #2
    }{
        #3
    }
}
```

`\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*⟩`}`

```
\DeclareDocumentCommand \IfCommandEmptyTF { m +m +m }{
    \tl_if_blank:oTF{#1}{
        #2
    }{
        #3
    }
}
```

2.3.2 Command Creators

\MakeCommand
\ReMakeCommand

This command pair uses the `etoolbox`'s `\csdef` to define a commands via a supplied string {⟨*Command Name*⟩} and a set of {⟨*Code*⟩}. 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{⟨Command Name⟩}{⟨Code⟩}
\ReMakeCommand{⟨Command Name⟩}{⟨Code⟩}
```

```
\DeclareDocumentCommand \MakeCommand { O{} m +m } {
    \cs_if_free:cTF {#2} {
        \cs_set:cpn {#2} #1 {#3}
    }{
        \msg_warning:nnnn
            {UWMadThesis}{Programming/Defined}{#2}{command}
    }
}
\DeclareDocumentCommand \ReMakeCommand { O{} m +m }{
    \cs_if_exist:cTF {#2} {
        \cs_set:cpn {#2} #1 {#3}
    }{
        \msg_error:nnnn
            {UWMadThesis}{Programming/Undefined}{#2}{command}
    }
}
```

\MakeGlobalCommand

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

Usage:

```
\MakeGlobalCommand{⟨Command Name⟩}{⟨Code⟩}
```

```
\DeclareDocumentCommand \MakeGlobalCommand { O{} +m m } {
    \cs_gset:cpn {#2} #1 {#3}
}
```

\MakeCommandUndefined

Globally undefines the command specified by {⟨*Command Name*⟩}.

Usage:

```
\MakeCommandUndefined{⟨Command Name⟩}
```

```
\DeclareDocumentCommand \MakeCommandUndefined { m } {
    \cs_undefine:c {#1}
}
```

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

Usage:

\CopyCommand{⟨*Command Name 1*⟩}{⟨*Command Name 2*⟩}

```
\DeclareDocumentCommand \CopyCommand { m m } {
    \cs_if_free:cTF {#1} {
        \cs_if_free:cTF {#2} {
            \cs_gset_eq:cc {#2}{#1}
        }{
            \msg_warning:nnnn
                {UWMadThesis}{Programming/Defined}{#2}{command}
        }
    }{
        \msg_warning:nnnn
            {UWMadThesis}{Programming/Defined}{#1}{command}
    }
}
```

2.3.3 Types

\CreateBoolean
\CreateBooleanTrue
\CreateBooleanFalse
\SetBooleanTrue
\SetBooleanFalse
\IfBooleanTrueTF
\IfBooleanFalseTF

LaTeX 2ε version of the Boolean Type system above.

```
\DeclareDocumentCommand \CreateBoolean { m } {
    \bool_new:c {g__UWMad_Programming_API_#1_bool}
}
\DeclareDocumentCommand \CreateBooleanTrue { m } {
    \bool_new:c       {g__UWMad_Programming_API_#1_bool}
    \bool_gset_true:c {g__UWMad_Programming_API_#1_bool}
}
\DeclareDocumentCommand \CreateBooleanFalse { m } {
    \bool_new:c        {g__UWMad_Programming_API_#1_bool}
}
\DeclareDocumentCommand \SetBooleanTrue { m } {
    \bool_gset_true:c {g__UWMad_Programming_API_#1_bool}
}
\DeclareDocumentCommand \SetBooleanFalse { m } {
    \bool_gset_false:c {g__UWMad_Programming_API_#1_bool}
}
```

```
\DeclareDocumentCommand \IfBooleanTrueTF { m +m +m } {
    \bool_if:cTF {g__UWMad_Programming_API_#1_bool} {
        #2
    } {
        #3
    }
}
\DeclareDocumentCommand \IfBooleanFalseTF { m +m +m } {
    \bool_if:cTF {g__UWMad_Programming_API_#1_bool} {
        #3
    } {
        #2
    }
}
```

\CreateLength
\AddToLength
\SetLength
\ValueOfLength
\IfLengthTF

LaTeX 2_ε version of the Boolean Type system above.

```
\DeclareDocumentCommand \CreateLength { m m } {
    \dim_new:c   {g__UWMad_Programming_API_#1_dim}
    \dim_gset:cn {g__UWMad_Programming_API_#1_dim} {#2}
}
\DeclareDocumentCommand \AddToLength { m m } {
    \dim_gadd:cn {g__UWMad_Programming_API_#1_dim} {#2}
}
\DeclareDocumentCommand \SetLength { m m } {
    \dim_gset:cn {g__UWMad_Programming_API_#1_dim} {#2}
}
\DeclareDocumentCommand \ValueOfLength { m } {
    \dim_use:c {g__UWMad_Programming_API_#1_dim}
}
\DeclareDocumentCommand \IfLengthTF { m m m +m +m } {
    \dim_compare:nNnTF {#1} #2 {#3} {
        #4
    } {
        #5
    }
}
```

\CreateCounter
\AddToCounter
\StepCounter
\SetCounter
\ValueOfCounter
\IfCounterTF
\CounterToArabic
\CounterToALPHA
\CounterToAlpha
\CounterToROMAN
\CounterToRoman

LaTeX 2_ε version of the Counter Type system above.

```
\DeclareDocumentCommand \CreateCounter { m m } {
    \int_new:c   {g__UWMad_Programming_API_#1_int}
    \int_gset:cn {g__UWMad_Programming_API_#1_int} {#2}
}
\DeclareDocumentCommand \AddToCounter { m m } {
    \int_gadd:cn {g__UWMad_Programming_API_#1_int} {#2}
}
\DeclareDocumentCommand \StepCounter { m m } {
    \int_gincr:cn {g__UWMad_Programming_API_#1_int} {#2}
}
\DeclareDocumentCommand \SetCounter { m m } {
    \int_gset:cn {g__UWMad_Programming_API_#1_int} {#2}
```

```
}
\DeclareDocumentCommand \ValueOfCounter { m m } {
    \int_use:c {g__UWMad_Programming_API_#1_int}
}
\DeclareDocumentCommand \IfCounterTF { m m m +m +m } {
    \int_compare:nNnTF {#1} {#2} {#3} {
        #4
    } {
        #5
    }
}
\DeclareDocumentCommand \CounterToArabic { m } {
    \int_to_arabic:c {g__UWMad_Programming_API_#1_int}
}
\DeclareDocumentCommand \CounterToALPHA { m } {
    \int_to_Alph:c {g__UWMad_Programming_API_#1_int}
}
\DeclareDocumentCommand \CounterToAlpha { m } {
    \int_to_alph:c {g__UWMad_Programming_API_#1_int}
}
\DeclareDocumentCommand \CounterToROMAN { m } {
    \int_to_Roman:c {g__UWMad_Programming_API_#1_int}
}
\DeclareDocumentCommand \CounterToRoman { m } {
    \int_to_roman:c {g__UWMad_Programming_API_#1_int}
}
```

Module 3

Layout And Styles

```
\geometry{
    includehead = true,
    margin      = 1.0in,
    paper       = letterpaper,
}
%
\creflabelformat{equation}{#2#1#3}
%
\captionsetup [table] {
    format        = hang                ,
    labelsep      = colon               ,
    justification = justified           ,
    labelfont     = sc                  ,
    textfont      = sl                  ,
    font          = {normal,stretch=1.1},
    width         = 0.9\textwidth       ,
    position      = above               ,
    skip          = 0.50em
}
%
\captionsetup [figure] {
    format        = hang                ,
    labelsep      = colon               ,
    justification = justified           ,
    labelfont     = sc                  ,
    textfont      = sl                  ,
    font          = {normal,stretch=1.1},
    width         = 0.9\textwidth       ,
    position      = above               ,
    skip          = 0.5em
}
%
\definecolor{UWMadGreen}{rgb}{0,0.7,0}
\hypersetup {
    colorlinks         = true       ,
    linkcolor          = blue       ,
    citecolor          = UWMadGreen ,
    urlcolor           = violet     ,
    pdfdisplaydoctitle = true       ,
    pdfview            = {FitH}     ,
    pdfstartview       = {FitH}     ,
    pdfpagelayout      = OneColumn  ,
    plainpages         = false      ,
    hypertexnames      = true       ,
    bookmarksopenlevel = 1          ,
```

```
    bookmarksopen      = true       ,
    unicode            = true
}
%
%
\doublespacing
\UWMad_Hook_Prepend:Nn \singlespacing {
    \__UWMad_FrontMatter_StyleWarning:n {
        University~guidelines~require~double-spacing.~
        If~this~is~for~temporary~use,~please~use~the~spacing~environment.
    }
}
\UWMad_Hook_Prepend:Nn \onehalfspacing {
    \__UWMad_FrontMatter_StyleWarning:n {
        University~guidelines~require~double-spacing.~
        If~this~is~for~temporary~use,~please~use~the~spacing~environment.
    }
}
%
%
%
%   Standard \LaTeX{} styles and lengths
\pagestyle{myheadings}
\setlength{\parindent}{ 0pt}
\setlength{\parskip}  {10pt}
\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.

```
\UWMad_Hook_Prepend:Nn \@chapter {
    \thispagestyle{myheadings}
    \int_compare:nNnTF {\value{chapter}} = {0} {
        \pagenumbering{arabic}
    } { }
}
\UWMad_Hook_Prepend:Nn \@schapter {
    \thispagestyle{myheadings}
}
```

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.

```
\int_new:N  \g__UWMad_Appendix_Counter_int
\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.

```
\cs_new:Nn \__UWMad_Appendix_Initialize:{
    \par
    \setcounter{section}{0}
    \cs_gset_eq:NN \@chapapp \appendixname
    \cs_gset:Npn \thechapter {
        \int_to_Alph:n {
            \g__UWMad_Appendix_Counter_int
        }
    }
}
```

Now, `\appendix` is undefined (to avoide a warning from xparse) and redefined with standard LaTeX 2_ε sectioning arguments.

```
\cs_undefine:N \appendix
\DeclareDocumentCommand \appendix { s o m } {

    \int_compare:nNnTF {\g__UWMad_Appendix_Counter_int} = {0} {
        \__UWMad_Appendix_Initialize:
    } { }
    \int_gincr:N \g__UWMad_Appendix_Counter_int

    \IfBooleanTF { #1 } {
        \chapter*{#3}
    } {
        \IfNoValueTF { #2 } {
            \chapter[#3]{#3}
        } {
            \chapter[#2]{#3}
        }
    }
}
```

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.

```
%
\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.

```
\cs_new:Nn \__UWMad_FrontMatter_Register:nn {

    \int_compare:nNnTF {\g__UWMad_FrontMatter_Counter_int} = {0} {
        \pagenumbering{roman}
    } { }

    \int_gincr:N \g__UWMad_FrontMatter_Counter_int
    \addcontentsline
        {toc}
        {#1}
        {#2}
```

```
}
```

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

```
\tl_new:N   \g__UWMad_FrontMatter_Title_Dedications_tl
\tl_new:N   \g__UWMad_FrontMatter_Title_Acknowledgments_tl
\tl_new:N   \g__UWMad_FrontMatter_Title_Abstract_tl
\tl_new:N   \g__UWMad_FrontMatter_Title_UMIAbstract_tl
\tl_new:N   \g__UWMad_FrontMatter_Title_Preface_tl
%
\tl_gset:Nn \g__UWMad_FrontMatter_Title_Dedications_tl
    {Dedications}
\tl_gset:Nn \g__UWMad_FrontMatter_Title_Acknowledgments_tl
    {Acknowledgments}
\tl_gset:Nn \g__UWMad_FrontMatter_Title_Abstract_tl
    {Abstract}
\tl_gset:Nn \g__UWMad_FrontMatter_Title_UMIAbstract_tl
    {Abstract}
\tl_gset:Nn \g__UWMad_FrontMatter_Title_Preface_tl
    {Preface}
```

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

```
\cs_undefine:N \abstract
\cs_undefine:N \endabstract

\DeclareDocumentCommand \FrontMatterSetSection { m m } {

    \tl_set_eq:Nc
        \l_tmpa_tl
        {g__UWMad_FrontMatter_Title_#2_tl}

    \IfNoValueTF { #1 } { } {
        \IfEmptyTF { #1 } { } {
            \tl_set:Nn \l_tmpa_tl {#1}
        }
    }

    \chapter*{\l_tmpa_tl}
    \__UWMad_FrontMatter_Register:nn {chapter} {
        \l_tmpa_tl
    }

}
\DeclareDocumentCommand \dedications { g } {
    \FrontMatterSetSection{#1}{Dedications}
}
\DeclareDocumentCommand \acknowledgments { g } {
    \FrontMatterSetSection{#1}{Acknowledgments}
}
\DeclareDocumentCommand \abstract { g } {
    \FrontMatterSetSection{#1}{Abstract}
```

```
}
\DeclareDocumentCommand \umiabstract { g } {
    \FrontMatterSetSection{#1}{Abstract}
}
\DeclareDocumentCommand \preface { g } {
    \FrontMatterSetSection{#1}{Preface}
}
```

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.

```
\cs_gset_eq:NN \TableOfContentsDefault \tableofcontents
\cs_gset_eq:NN \ListOfTablesDefault    \listoftables
\cs_gset_eq:NN \ListOfFiguresDefault   \listoffigures
\cs_undefine:N \tableofcontents
\cs_undefine:N \listoftables
\cs_undefine:N \listoffigures
```

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

```
\tl_new:N   \g__UWMad_TOC_Name_TOC_tl
\tl_new:N   \g__UWMad_TOC_Name_LOT_tl
\tl_new:N   \g__UWMad_TOC_Name_LOF_tl
\tl_gset:Nn \g__UWMad_TOC_Name_TOC_tl {Table~of~Contents}
\tl_gset:Nn \g__UWMad_TOC_Name_LOT_tl {List~of~Tables}
\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.

```
\DeclareDocumentCommand \tableofcontents { } {

    \tl_gset_eq:NN \contentsname \g__UWMad_TOC_Name_TOC_tl

    \group_begin:
        \setstretch{1.05}
        \phantomsection
        \ExplSyntaxOff
            \TableOfContentsDefault
        \ExplSyntaxOn
        \__UWMad_FrontMatter_Register:nn
```

```
            {chapter}
            {\contentsname}
        \clearpage
    \group_end:
}
\DeclareDocumentCommand \listoftables { } {

    \cs_set_eq:NN \listtablename \g__UWMad_TOC_Name_LOT_tl

    \group_begin:
        \setstretch{1.05}
        \ExplSyntaxOff
            \ListOfTablesDefault
        \ExplSyntaxOn
        \__UWMad_FrontMatter_Register:nn
            {chapter}
            {\listtablename}
        \clearpage
    \group_end:
}
\DeclareDocumentCommand \listoffigures { } {

    \cs_set_eq:NN \listfigurename \g__UWMad_TOC_Name_LOF_tl

    \group_begin:
        \setstretch{1.05}
        \ExplSyntaxOff
            \ListOfFiguresDefault
        \ExplSyntaxOn
        \__UWMad_FrontMatter_Register:nn
            {chapter}
            {\listfigurename}
        \clearpage
    \group_end:
}
```

Camel-cased aliases.

```
\cs_set_eq:NN \TableOfContents \tableofcontents
\cs_set_eq:NN \ListOfTables    \listoftables
\cs_set_eq:NN \ListOfFigures   \listoffigures
```

User-level commands to change the default names.

```
\DeclareDocumentCommand \TableOfContentsName { m } {
    \tl_gset:Nn \g__UWMad_TOC_Name_TOC_tl {#1}
}
\DeclareDocumentCommand \ListOfTablesName { m } {
    \tl_gset:Nn \g__UWMad_TOC_Name_LOT_tl {#1}
}
\DeclareDocumentCommand \ListOfFiguresName { m } {
    \tl_gset:Nn \g__UWMad_TOC_Name_LOF_tl {#1}
}
```

4.4 Section-Level Commands

These commands are used internally when needing to check if a user-supplied `section` is a LaTeX 2_ε-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.

```
\tl_const:Nn \c__UWMad_SectionsLevel_part_tl            {-1}
\tl_const:Nn \c__UWMad_SectionsLevel_chapter_tl         {0}
\tl_const:Nn \c__UWMad_SectionsLevel_section_tl         {1}
\tl_const:Nn \c__UWMad_SectionsLevel_subsection_tl      {2}
\tl_const:Nn \c__UWMad_SectionsLevel_subsubsection_tl   {3}
\tl_const:Nn \c__UWMad_SectionsLevel_paragraph_tl       {4}
\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.

```
\msg_new:nnn { UWMadThesis } { Sectioning / UndefinedSection } {
    Undefined~section~'#1'~used.
}
\cs_new:Nn \UWMad_IfSectionExists:nT {
    \tl_if_exist:cTF {c__UWMad_SectionsLevel_ #1 _tl} {
        #2
    } {
        \msg_error:nnn
            { UWMadThesis }
            { Sectioning / UndefinedSection }
            {#1}
    }
}
```

Variables that map a level number to a section.

```
\tl_const:cn {c__UWMad_LevelsSection_-1_tl} {part}
\tl_const:cn {c__UWMad_LevelsSection_ 0_tl} {chapter}
\tl_const:cn {c__UWMad_LevelsSection_ 1_tl} {section}
\tl_const:cn {c__UWMad_LevelsSection_ 2_tl} {subsection}
\tl_const:cn {c__UWMad_LevelsSection_ 3_tl} {subsubsection}
\tl_const:cn {c__UWMad_LevelsSection_ 4_tl} {paragraph}
\tl_const:cn {c__UWMad_LevelsSection_ 5_tl} {subparagraph}
```

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

```
\tl_const:Nn \c__UWMad_NextSection_part_tl            {chapter}
```

```
\tl_const:Nn \c__UWMad_NextSection_chapter_tl         {section}
\tl_const:Nn \c__UWMad_NextSection_section_tl         {subsection}
\tl_const:Nn \c__UWMad_NextSection_subsection_tl      {subsubsection}
\tl_const:Nn \c__UWMad_NextSection_subsubsection_tl   {paragraph}
\tl_const:Nn \c__UWMad_NextSection_paragraph_tl       {subparagraph}
```

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

```
\tl_const:Nn \c__UWMad_PreviousSection_chapter_tl         {part}
\tl_const:Nn \c__UWMad_PreviousSection_section_tl         {chapter}
\tl_const:Nn \c__UWMad_PreviousSection_subsection_tl      {section}
\tl_const:Nn \c__UWMad_PreviousSection_subsubsection_tl   {subsection}
\tl_const:Nn \c__UWMad_PreviousSection_paragraph_tl       {subsubsection}
\tl_const:Nn \c__UWMad_PreviousSection_subparagraph_tl    {paragraph}
```

Given a section, acquire its level number.

```
\cs_new:Nn \UWMad_SectionToLevel:n {
    \UWMad_IfSectionExists:nT {#1} {
        \tl_use:c {c__UWMad_SectionsLevel_ #1 _tl}
    }
}
```

Given a level number, acquire its section.

```
\cs_new:Nn \UWMad_LevelToSection:n {
    \UWMad_IfSectionExists:nT {#1} {
        \tl_use:c {c__UWMad_LevelsSection_ #1 _tl}
    }
}
```

Given a section, acquire its next lower one.

```
\cs_new:Nn \UWMad_NextSection:n {
    \UWMad_IfSectionExists:nT {#1} {
        \tl_use:c {c__UWMad_NextSection_ #1 _tl}
    }
}
```

Given a section, acquire its next higher one.

```
\cs_new:Nn \UWMad_PreviousSection:n {
    \UWMad_IfSectionExists:nT {#1} {
        \tl_use:c {c__UWMad_PreviousSection_ #1 _tl}
    }
}
```

Module 5

Math

```
%
%
%
\tex_everydisplay:D \exp_after:wN {
    \tex_the:D \tex_everydisplay:D
    \cs_set_eq:NN \frac \dfrac
}
%
%
%
%
%
% ============================================================================ %
%                              Derivative Commands                             %
% ============================================================================ %
%
\tl_new:N   \g_UWMad_Math_derivSymbol_tl
\tl_gset:Nn \g_UWMad_Math_derivSymbol_tl   {\mathrm{d}}
\tl_new:N   \g_UWMad_Math_pderivSymbol_tl
\tl_gset:Nn \g_UWMad_Math_pderivSymbol_tl  {\partial}
\tl_new:N   \g_UWMad_Math_tderivSymbol_tl
\tl_gset:Nn \g_UWMad_Math_tderivSymbol_tl  {\mathrm{D}}
\tl_new:N   \g_UWMad_Math_DelimiterDefaultLeft_tl
\tl_gset:Nn \g_UWMad_Math_DelimiterDefaultLeft_tl  {[}
\tl_new:N   \g_UWMad_Math_DelimiterDefaultRight_tl
\tl_gset:Nn \g_UWMad_Math_DelimiterDefaultRight_tl {]}
\tl_new:N   \l_UWMad_Math_DelimiterLeft_tl
\tl_new:N   \l_UWMad_Math_DelimiterRight_tl
%
%
\DeclareDocumentCommand \derivSymbol { } {
    \g_UWMad_Math_derivSymbol_tl
}
\DeclareDocumentCommand \pderivSymbol { } {
    \g_UWMad_Math_pderivSymbol_tl
}
\DeclareDocumentCommand \tderivSymbol { } {
    \g_UWMad_Math_tderivSymbol_tl
}
%
%
\DeclareDocumentCommand \derivSymbolChange { m } {
    \tl_set:Nn \g_UWMad_Math_derivSymbol_tl {#1}
}
\DeclareDocumentCommand \pderivSymbolChange { m } {
```

```
    \tl_set:Nn \g_UWMad_Math_pderivSymbol_tl {#1}
}
\DeclareDocumentCommand \tderivSymbolChange { m } {
    \tl_set:Nn \g_UWMad_Math_tderivSymbol_tl {#1}
}
%
%
\DeclareDocumentCommand \derivSymbolChangeDefault { m } {
    \tl_gset:Nn \g_UWMad_Math_derivSymbol_tl {#1}
}
\DeclareDocumentCommand \pderivSymbolChangeDefault { m } {
    \tl_gset:Nn \g_UWMad_Math_pderivSymbol_tl {#1}
}
\DeclareDocumentCommand \tderivSymbolChangeDefault { m } {
    \tl_gset:Nn \g_UWMad_Math_tderivSymbol_tl {#1}
}
%
%
\DeclareDocumentCommand \DelimiterChangeDefault { m m } {
    \tl_gset:Nn  \g_UWMad_Math_DelimiterDefaultLeft_tl  {#1}
    \tl_gset:Nn  \g_UWMad_Math_DelimiterDefaultRight_tl {#2}
}
%
%
\DeclareDocumentCommand \DerivativeGeneral { +m +m m m } {
    \frac{ #4^{#3} #1      }
         { #4      #2^{#3} }
}
\DeclareDocumentCommand \DerivativeGeneralBig { +m +m m m m m} {

    \IfNoValueTF {#5} {
        \tl_set_eq:NN
            \l_UWMad_Math_DelimiterLeft_tl
            \g_UWMad_Math_DelimiterDefaultLeft_tl
    } {
        \tl_set:Nn \l_UWMad_Math_DelimiterLeft_tl {#5}
    }

    \IfNoValueTF {#6} {
        \tl_set_eq:NN
            \l_UWMad_Math_DelimiterRight_tl
            \g_UWMad_Math_DelimiterDefaultRight_tl
    } {
        \tl_set:Nn \l_UWMad_Math_DelimiterRight_tl {#6}
    }

    \frac{ #4^{#3}    }
         { #4 #2^{#3} }
    \!\!
    \left\l_UWMad_Math_DelimiterLeft_tl
        #1
    \right\l_UWMad_Math_DelimiterRight_tl
}
%
%
\DeclareDocumentCommand \deriv { +m +m G{} } {
```

```
    \DerivativeGeneral
        {#1}{#2}{#3}{\derivSymbol}
}
\DeclareDocumentCommand \pderiv { +m +m G{} } {
    \DerivativeGeneral
        {#1}{#2}{#3}{\pderivSymbol}
}
\DeclareDocumentCommand \tderiv { +m +m G{} } {
    \DerivativeGeneral
        {#1}{#2}{#3}{\tderivSymbol}
}
%
%
\DeclareDocumentCommand \derivbig { o +m o +m G{} } {
    \DerivativeGeneralBig
        {#2}{#4}{#5}{\derivSymbol}{#1}{#3}
}
\DeclareDocumentCommand \pderivbig { o +m o +m G{} } {
    \DerivativeGeneralBig
        {#2}{#4}{#5}{\pderivSymbol}{#1}{#3}
}
\DeclareDocumentCommand \tderivbig { o +m o +m G{} } {
    \DerivativeGeneralBig
        {#2}{#4}{#5}{\tderivSymbol}{#1}{#3}
}
%
%
%
%
\DeclareMathOperator*{\Sup}    {Sup}
\DeclareMathOperator*{\Inf}    {Inf}
\DeclareMathOperator*{\Lim}    {Lim}
\DeclareMathOperator*{\Min}    {Min}
\DeclareMathOperator*{\Max}    {Max}
\DeclareMathOperator*{\ArgMin} {ArgMin}
\DeclareMathOperator*{\ArgMax} {ArgMax}
\DeclareMathOperator{\Abs}     {Abs}
\DeclareMathOperator{\Ln}      {Ln}
\DeclareMathOperator{\Log}     {Log}
\DeclareMathOperator{\Exp}     {Exp}
\DeclareMathOperator{\Cos}     {Cos}
\DeclareMathOperator{\Sin}     {Sin}
\DeclareMathOperator{\Tan}     {Tan}
\DeclareMathOperator{\Sec}     {Sec}
\DeclareMathOperator{\Csc}     {Csc}
\DeclareMathOperator{\Cot}     {Cot}
\DeclareMathOperator{\Cosh}    {Cosh}
\DeclareMathOperator{\Sinh}    {Sinh}
\DeclareMathOperator{\Tanh}    {Tanh}
\DeclareMathOperator{\Sech}    {Sech}
\DeclareMathOperator{\Csch}    {Csch}
\DeclareMathOperator{\Coth}    {Coth}
\DeclareMathOperator{\ArcCos}  {ArcCos}
\DeclareMathOperator{\ArcSin}  {ArcSin}
\DeclareMathOperator{\ArcTan}  {ArcTan}
\DeclareMathOperator{\ArcSec}  {ArcSec}
```

```
\DeclareMathOperator{\ArcCsc}  {ArcCsc}
\DeclareMathOperator{\ArcCot}  {ArcCot}
\DeclareMathOperator{\ArcCosh} {ArcCosh}
\DeclareMathOperator{\ArcSinh} {ArcSinh}
\DeclareMathOperator{\ArcTanh} {ArcTanh}
\DeclareMathOperator{\ArcSech} {ArcSech}
\DeclareMathOperator{\ArcCsch} {ArcCsch}
\DeclareMathOperator{\ArcCoth} {ArcCoth}
%
%
%
%
%
% ============================================================================ %
%                              Miscellaneous Commands                          %
% ============================================================================ %
\cs_new:Nn \UWMad_Math_RootWithTail:nn {

    \hbox_set:Nn \l_tmpa_box {
        $
            \mathchoice
                {\root #1 \of {#2\:\!}}
                {\root #1 \of {#2\:\!}}
                {\root #1 \of {#2\:\!}}
                {\root #1 \of {#2\:\!}}
        $
    }
    %
    \dim_set:Nn \l_tmpa_dim {\box_ht:N \l_tmpa_box}
    \dim_set:Nn \l_tmpb_dim {0.8\l_tmpa_dim}
    %
    \hbox_set:Nn \l_tmpb_box {
        \tex_vrule:D height \l_tmpa_dim depth -\l_tmpb_dim
    }
    %
    \box_use:N \l_tmpa_box
    \box_move_down:nn {0.40pt}{\box_use:N \l_tmpb_box}
}
\DeclareDocumentCommand \Sqrt { O{} m } {
    \UWMad_Math_RootWithTail:nn{#1}{#2}
}
%
%
\DeclareExpandableDocumentCommand \IfMathModeTF { +m +m } {
    \mode_if_math:TF {
        #1
    }{
        $#2$
    }
}
\cs_gset_eq:NN \supsipa \sups
\cs_undefine:N \sups
\ExplSyntaxOff
    \DeclareDocumentCommand \subs { O{} +m } {%
        \IfMathModeTF{%
            _{\!\!\:#1\text{\scriptsize #2}}%
```

```
        }{%
            _{\!#1\text{\scriptsize #2}}%
        }%
    }%
    \DeclareDocumentCommand \sups { O{} +m } {%
        \IfMathModeTF{%
            ^{#1\text{\scriptsize #2}}%
        }{%
            ^{#1\text{\scriptsize #2}}%
        }%
    }%
    \DeclareDocumentCommand \subsups { O{} +m O{} +m } {%
        \IfMathModeTF{%
            _{#1\text{\scriptsize #2}}^{\!\!\:#3\text{\scriptsize #4}}%
        }{%
            _{#1\text{\scriptsize #2}}^{\!\!\!#3\text{\scriptsize #4}}%
        }%
    }%
\ExplSyntaxOn
%
\DeclareDocumentCommand \OneOver { +m } {
    \frac{1}{#1}
}
\DeclareDocumentCommand \oneo { +m } {
    \OneOver{#1}
}
\DeclareDocumentCommand \dd { m } {
    \mathrm{d}{#1}
}
\DeclareDocumentCommand \dprime { } {
    {\prime\prime}
}
\DeclareDocumentCommand \tprime { } {
    {\prime\prime\prime}
}
\DeclareDocumentCommand \LessThan        { } {<}
\DeclareDocumentCommand \GreaterThanThan { } {>}
%
```

Module 6

ListOf

The ListOf Module is a collection of commands that enables the easy creation and typsetting 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 entires. The `ListOf` commands create a system specifically for this scenario.

Of course, as the commands description will show, lists can be much more complicated that 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 `{`⟨*ID*⟩`}`. 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 contentst (regardless of numbering), a hash to hold of the user-defined hooks for the `ListOf`, and a queue to store the entries for typesetting.

```
\cs_new:Nn \UWMad_ListOf_Define:n {
    \tl_const:cn {c__UWMad_ListOf#1_IsDefined_tl}{}
%
    \tl_new:c {g__UWMad_ListOf#1_Section_Main_tl}
    \tl_new:c {g__UWMad_ListOf#1_Section_Group_tl}
    \tl_new:c {g__UWMad_ListOf#1_Section_Subgroup_tl}
%
    \tl_new:c {g__UWMad_ListOf#1_Title_Main_tl}
    \tl_new:c {g__UWMad_ListOf#1_Title_Group_tl}
    \tl_new:c {g__UWMad_ListOf#1_Title_Subgroup_tl}
%
    \bool_new:c       {g__UWMad_ListOf#1_ClearAfterPrint_bool}
    \bool_gset_true:c {g__UWMad_ListOf#1_ClearAfterPrint_bool}
    \bool_new:c       {g__UWMad_ListOf#1_IsNumbered_bool}
    \bool_gset_true:c {g__UWMad_ListOf#1_IsNumbered_bool}
    \bool_new:c       {g__UWMad_ListOf#1_IncludeInTOC_bool}
    \bool_gset_true:c {g__UWMad_ListOf#1_IncludeInTOC_bool}
    \UWMad_Queue_Define:n               {g__ListOf#1_EntryQueue}
    \UWMad_Hash_Define:n                {g__ListOf#1_Hook}
}
```

`\UWMad_ListOf_Delete:n` Simply undefines all of the commands created in the `Define` command above for the given `{`⟨*ID*⟩`}`.

```
\cs_new:Nn \UWMad_ListOf_Delete:n {
    \cs_undefine:c {c__UWMad_ListOf#1_IsDefined_tl}
%
    \cs_undefine:c {g__UWMad_ListOf#1_Section_Main_tl}
    \cs_undefine:c {g__UWMad_ListOf#1_Section_Group_tl}
    \cs_undefine:c {g__UWMad_ListOf#1_Section_Subgroup_tl}
%
    \cs_undefine:c {g__UWMad_ListOf#1_Title_Main_tl}
    \cs_undefine:c {g__UWMad_ListOf#1_Title_Group_tl}
    \cs_undefine:c {g__UWMad_ListOf#1_Title_Subgroup_tl}
%
    \cs_undefine:c {g__UWMad_ListOf#1_ClearAfterPrint_bool}
    \cs_undefine:c {g__UWMad_ListOf#1_IsNumbered_bool}
    \cs_undefine:c {g__UWMad_ListOf#1_IncludeInTOC_bool}
    \UWMad_Queue_Delete:n   {g__ListOf#1_EntryQueue}
    \UWMad_Hash_Delete:n    {g__ListOf#1_Hook}
}
```

`\UWMad_ListOf_IfDefined:nT` Checks to see if a `ListOf` with `{`⟨*ID*⟩`}` has been created and errors if not.

```
\cs_new:Nn \UWMad_ListOf_IfDefined:nT {
    \__UWMad_IfDefined:nnnnT
        {c__UWMad_ListOf}
        {#1}
        {_IsDefined_tl}
        {ListOf}
        {#2}
}
```

`\UWMad_ListOf_MakeNumbered:n` `\UWMad_ListOf_MakeNotNumbered:n` Makes the current section of the `ListOf` with `{`⟨*ID*⟩`}` numbered or unnumbered (i.e., a star version).

```
\cs_new:Nn \UWMad_ListOf_MakeNumbered:n {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \bool_set_true:c {g__UWMad_ListOf#1_IsNumbered_bool}
    }
}
\cs_new:Nn \UWMad_ListOf_MakeNotNumbered:n {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \bool_set_false:c {g__UWMad_ListOf#1_IsNumbered_bool}
    }
}
```

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

```
\cs_new:Nn \UWMad_ListOf_IfNumbered:nTF {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \bool_if:cTF {g__UWMad_ListOf#1_IsNumbered_bool} {
            #2
```

```
        }{
            #3
        }
    }
}
```

\UWMad_ListOf_IncludeInTOC:n
\UWMad_ListOf_DoNotIncludeInTOC:n

Makes the current section of the `ListOf` with {⟨*ID*⟩} appear in the Table of Contents (TOC) or not, regardless of if it is numbered/unnumbered.

```
\cs_new:Nn \UWMad_ListOf_IncludeInTOC:n {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \bool_set_true:c {c__UWMad_ListOf#1_IncludeInTOC_bool}
    }
}
\cs_new:Nn \UWMad_ListOf_DoNotIncludeInTOC:n {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \bool_set_false:c {c__UWMad_ListOf#1_IncludeInTOC_bool}
    }
}
```

\UWMad_ListOf_IfIncludeInTOC:n

Branches to {⟨*True Code*⟩} or {⟨*False Code*⟩} depending on whether the `ListOf` with {⟨*ID*⟩} is to be included or not.

```
\cs_new:Nn \UWMad_ListOf_IfIncludeInTOC:nTF {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \bool_if:cTF {c__UWMad_ListOf#1_IncludeInTOC_bool} {
            #2
        }{
            #3
        }
    }
}
```

`\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*⟩}

```
\cs_new:Nn \UWMad_ListOf_SetTitle_Main:nn {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_set:cn {g__UWMad_ListOf#1_Title_Main_tl}{#2}
    }
}
\cs_new:Nn \UWMad_ListOf_SetTitle_Group:nn {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_set:cn {g__UWMad_ListOf#1_Title_Group_tl}{#2}
    }
}
\cs_new:Nn \UWMad_ListOf_SetTitle_Subgroup:nn {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_set:cn {g__UWMad_ListOf#1_Title_Subgroup_tl}{#2}
    }
}
```

`\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*⟩`}`

Retrieces the value of the title of the section for the `ListOf` with {⟨*ID*⟩}.

```
\cs_new:Nn \UWMad_ListOf_GetTitle_Main:n {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_use:c {g__UWMad_ListOf#1_Title_Main_tl}
    }
}
\cs_new:Nn \UWMad_ListOf_GetTitle_Group:n {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_use:c {g__UWMad_ListOf#1_Title_Group_tl}
    }
}
\cs_new:Nn \UWMad_ListOf_GetTitle_Subgroup:n {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_use:c {g__UWMad_ListOf#1_Title_Subgroup_tl}
    }
}
```

`\UWMad_ListOf_SetSection_Main:nn`	`\UWMad_ListOf_SetSection_Main:nn{⟨ID⟩}{⟨Section⟩}`
`\UWMad_ListOf_SetSection_Group:nn`	`\UWMad_ListOf_SetSection_Group:nn{⟨ID⟩}{⟨Section⟩}`
`\UWMad_ListOf_SetSection_Subgroup:nn`	`\UWMad_ListOf_SetSection_Subgroup:nn{⟨ID⟩}{⟨Section⟩}`

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

```
\cs_new:Nn \UWMad_ListOf_SetSection_Main:nn {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_set:cn {g__UWMad_ListOf#1_Section_Main_tl}{#2}
    }
}
\cs_new:Nn \UWMad_ListOf_SetSection_Group:nn {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_set:cn {g__UWMad_ListOf#1_Section_Group_tl}{#2}
    }
}
\cs_new:Nn \UWMad_ListOf_SetSection_Subgroup:nn {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_set:cn {g__UWMad_ListOf#1_Section_Subgroup_tl}{#2}
    }
}
```

`\UWMad_ListOf_GetSection_Main:n`	`\UWMad_ListOf_GetSection_Main:n{⟨ID⟩}`
`\UWMad_ListOf_GetSection_Group:n`	`\UWMad_ListOf_GetSection_Group:n{⟨ID⟩}`
`\UWMad_ListOf_GetSection_Subgroup:n`	`\UWMad_ListOf_GetSection_Subgroup:n{⟨ID⟩}`

Gets the value of the section level for a particular group of the `ListOf` with {⟨*ID*⟩}.

```
\cs_new:Nn \UWMad_ListOf_GetSection_Main:n {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_use:c {g__UWMad_ListOf#1_Section_Main_tl}
    }
}
\cs_new:Nn \UWMad_ListOf_GetSection_Group:n {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_use:c {g__UWMad_ListOf#1_Section_Group_tl}
    }
}
\cs_new:Nn \UWMad_ListOf_GetSection_Subgroup:n {
    \UWMad_ListOf_IfDefined:nT {#1} {
        \tl_use:c {g__UWMad_ListOf#1_Section_Subgroup_tl}
    }
}
```

\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`.

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

\UWMad_ListOf_PushEntry:nn `\UWMad_ListOf_PushEntry:nn {⟨ID⟩}{⟨Entry⟩}`

Pushes {⟨*Entry*⟩} on to the entry queue of the `ListOf` with {⟨*ID*⟩}.

```
\cs_new:Nn \UWMad_ListOf_PushEntry:nn {
    \UWMad_Hash_Get:nn   {g__ListOf#1_Hook}{PrePush}
    \UWMad_Queue_Push:nn {g__ListOf#1_EntryQueue}{#2}
    \UWMad_Hash_Get:nn   {g__ListOf#1_Hook}{PostPush}
}
```

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.

```
\cs_new:Nn \UWMad_ListOf_PrintEntries:n {
    \UWMad_Hash_Get:nn   {g__ListOf#1_Hook}{PrePrint}
    \UWMad_Queue_Walk:nn {g__ListOf#1_EntryQueue}{##1}
    \UWMad_Queue_Clear:n {g__ListOf#1_EntryQueue}
    \UWMad_Hash_Get:nn   {g__ListOf#1_Hook}{PostPrint}
}
```

\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.

```
\cs_new:Nn \__UWMad_ListOf_CurrentSectioningCommmand:n {}
\cs_new:Nn \UWMad_ListOf_PrintTitle:nn {

    \cs_set_eq:Nc
        \__UWMad_ListOf_CurrentSectioningCommmand:n
        {\tl_use:c{g__UWMad_ListOf#1_Section_#2_tl}}

    \UWMad_ListOf_IfNumbered:nTF {#1} {

        \tl_if_eq:nnTF {#2} {Main} {
            \UWMad_ListOf_IfIncludeInTOC:nTF {#1} { } {
                \int_set_eq:NN \l_tmpa_int \c@tocdepth
                \setcounter{tocdepth}{-1}
            }
        } {
            \int_set_eq:NN \l_tmpa_int \c@tocdepth
            \setcounter{tocdepth}{-1}
        }


        \__UWMad_ListOf_CurrentSectioningCommmand:n
            {\tl_use:c {g__UWMad_ListOf#1_Title_#2_tl}}


        \tl_if_eq:nnTF #2 {Main} {
            \UWMad_ListOf_IfIncludeInTOC:nTF {#1} { } {
                \setcounter{tocdepth}{\l_tmpa_int}
            }
        } {
            \setcounter{tocdepth}{\l_tmpa_int}
        }

    } {

        \cs_generate_variant:Nn \tl_if_eq:nnTF {onTF}
        \tl_set:Nn \l_tmpa_tl {Main}

        \phantomsection
        \__UWMad_ListOf_CurrentSectioningCommmand:n*
        {\tl_use:c {g__UWMad_ListOf#1_Title_#2_tl}}
        \tl_if_eq:onTF {#2} {Main} {
            \UWMad_ListOf_IfIncludeInTOC:nTF {#1} {
                \addcontentsline
                    {toc}
                    {\tl_use:c {g__UWMad_ListOf#1_Section_#2_tl}}
                    {\tl_use:c {g__UWMad_ListOf#1_Title_#2_tl}}
            } { }
        } { }
    }

}
```

\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.

```
\cs_new:Nn \UWMad_ListOf_StartGroup:nn {
    \UWMad_ListOf_PrintEntries:n{#1}
    \UWMad_ListOf_PrintTitle:nn {#1}{#2}
}
```

6.1 Nomenclature

Dimensions that are calculated are declared first.

```
\dim_new:N \l__UWMad_Nomenclature_WidestSymbol_dim
\dim_new:N \l__UWMad_Nomenclature_WidestUnit_dim
\dim_new:N \l__UWMad_Nomenclature_Entry_WidthSymbol_dim
\dim_new:N \l__UWMad_Nomenclature_Entry_WidthUnits_dim
\dim_new:N \l__UWMad_Nomenclature_Entry_WidthDescription_dim
```

Then user-adjustable dimensions are declared.

```
\dim_new:N \l__UWMad_Nomenclature_TitleSkip_dim
\dim_new:N \l__UWMad_Nomenclature_PrintSkip_dim
\dim_new:N \l__UWMad_Nomenclature_Entry_MarginLeft_dim
\dim_new:N \l__UWMad_Nomenclature_Entry_MarginBottom_dim
\dim_new:N \l__UWMad_Nomenclature_Entry_Padding_dim
```

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

```
\tl_new:N \l__UWMad_Nomenclature_Section_Main_tl
\tl_new:N \l__UWMad_Nomenclature_Section_Group_tl
\tl_new:N \l__UWMad_Nomenclature_Section_Subgroup_tl
\tl_new:N \l__UWMad_Nomenclature_Title_Main_tl
\tl_new:N \l__UWMad_Nomenclature_Title_Group_tl
\tl_new:N \l__UWMad_Nomenclature_Title_Subgroup_tl
```

Now the keys for user-customization are defined:

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

Adjustable dimensions:

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

```
    print-skip .dim_set:N = \l__UWMad_Nomenclature_PrintSkip_dim,
    entry-margin-left .dim_set:N =
        \l__UWMad_Nomenclature_Entry_MarginLeft_dim,
    entry-margin-bottom .dim_set:N =
        \l__UWMad_Nomenclature_Entry_MarginBottom_dim,
    entry-padding .dim_set:N =
        \l__UWMad_Nomenclature_Entry_Padding_dim,
```

Adjustable dimension defaults:

```
    title-skip          .default:n  = {0.00pt},
    print-skip          .default:n  = {1.00em},
    entry-margin-left   .default:n  = {1.00em},
    entry-margin-bottom .default:n  = {0.25em},
    entry-padding       .default:n  = {0.75em},
```

Group section adjustments:

```
    main-section .code:n = {
        \tl_set:Nn
            \l__UWMad_Nomenclature_Section_Main_tl {#1}
    },
    group-section .code:n = {
        \tl_set:Nn
            \l__UWMad_Nomenclature_Section_Group_tl {#1}
    },
    subgroup-section .code:n = {
        \tl_set:Nn
            \l__UWMad_Nomenclature_Section_Subgroup_tl {#1}
    },
```

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

```
    main-section .default:n = chapter,
```

Group title adjustments:

```
    main-title .code:n = {
        \tl_set:Nn
            \l__UWMad_Nomenclature_Title_Main_tl {#1}
    },
    group-title .code:n = {
        \tl_set:Nn
            \l__UWMad_Nomenclature_Title_Group_tl {#1}
    },
    subgroup-title .code:n = {
        \tl_set:Nn
            \l__UWMad_Nomenclature_Title_Subgroup_tl {#1}
    },
```

Group title default for main group only:

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

Miscellaneous options:

```
    numbered       .bool_gset:N =
        \g__UWMad_Nomenclature_IsNumbered_bool,
    include-in-toc .bool_gset:N =
        \g__UWMad_Nomenclature_IncludeInTOC_bool,
    with-units .bool_gset:N =
        \g__UWMad_Nomenclature_IncludeUnitsColumn_bool,
```

Miscellaneous option defaults:

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

And the defaults for all keys are now set.

```
\keys_set:nn { UWMad/Nomenclature } {
    title-skip          ,
    print-skip          ,
    entry-margin-left   ,
    entry-margin-bottom ,
    entry-padding       ,
    numbered            ,
    include-in-toc      ,
    with-units          ,
    main-section        ,
    main-title
}
```

And the defaults for all keys are now set.

\UWMad_Nomenclature_UpdateWidest:Nn	\UWMad_Nomenclature_UpdateWidest:Nn⟨*dim*⟩{⟨*object*⟩}
\UWMad_Nomenclature_UpdateWidest_Symbol:n	\UWMad_Nomenclature_UpdateWidest_Symbol:n{⟨*symbol*⟩}
\UWMad_Nomenclature_UpdateWidest_Units:n	\UWMad_Nomenclature_UpdateWidest_Units:n{⟨*units*⟩}

These commands update the widest symbol and widest unit lengths.

```
\cs_new:Nn \UWMad_Nomenclature_UpdateWidest:Nn {
    \hbox_set:Nn \l_tmpa_box {#2}
    \dim_set:Nn  \l_tmpa_dim {\box_wd:N \l_tmpa_box}
    \dim_compare:nNnTF {#1} < {\l_tmpa_dim} {
        \dim_set:Nn #1 {\l_tmpa_dim}
    } { }
}
\cs_new:Nn \UWMad_Nomenclature_UpdateWidest_Symbol:n {
    \UWMad_Nomenclature_UpdateWidest:Nn
        \l__UWMad_Nomenclature_WidestSymbol_dim {#1}
}
%
\cs_new:Nn \UWMad_Nomenclature_UpdateWidest_Units:n {
    \UWMad_Nomenclature_UpdateWidest:Nn
        \l__UWMad_Nomenclature_WidestUnit_dim {#1}
}
%
%
\cs_new:Nn \UWMad_Nomenclature_ZeroWidest_Symbol: {
    \dim_set:Nn \l__UWMad_Nomenclature_WidestSymbol_dim {0pt}
}
\cs_new:Nn \UWMad_Nomenclature_ZeroWidest_Unit: {
    \dim_set:Nn \l__UWMad_Nomenclature_WidestUnit_dim {0pt}
}


%
%
% ========================================================================= %
%                               Set Entry Widths                            %
% ========================================================================= %
\cs_new:Nn \UWMad_Nomenclature_SetEntryWidths_NoUnits: {
    %
    % Define symbol width
    \dim_set:Nn \l__UWMad_Nomenclature_Entry_WidthSymbol_dim {
        1.01\l__UWMad_Nomenclature_WidestSymbol_dim
    }
    %
    % Define description width
    \dim_set:Nn \l__UWMad_Nomenclature_Entry_WidthDescription_dim {
        0.999\textwidth -
        \l__UWMad_Nomenclature_Entry_MarginLeft_dim  -
        \l__UWMad_Nomenclature_Entry_WidthSymbol_dim -
        \l__UWMad_Nomenclature_Entry_Padding_dim
    }
}
%
\cs_new:Nn \UWMad_Nomenclature_SetEntryWidths_Units: {
```

```
    %
    % Define symbol width
    \dim_set:Nn \l__UWMad_Nomenclature_Entry_WidthSymbol_dim {
        1.01\l__UWMad_Nomenclature_WidestSymbol_dim
    }
    %
    % Define unit width
    \dim_set:Nn \l__UWMad_Nomenclature_Entry_WidthUnit_dim {
        1.01\l__UWMad_Nomenclature_WidestUnit_dim
    }
    %
    % Define description width
    \dim_set:Nn \l__UWMad_Nomenclature_Entry_WidthDescription_dim {
        0.999\textwidth -
         \l__UWMad_Nomenclature_Entry_MarginLeft_dim  -
         \l__UWMad_Nomenclature_Entry_WidthSymbol_dim -
         \l__UWMad_Nomenclature_Entry_WidthUnit_dim   -
        2\l__UWMad_Nomenclature_Entry_Padding_dim
    }
}
%
\cs_new:Nn \UWMad_Nomenclature_SetEntryWidths: {
    \bool_if:NTF \g__UWMad_Nomenclature_IncludeUnitsColumn_bool {
        \UWMad_Nomenclature_SetEntryWidths_Units:
    } {
        \UWMad_Nomenclature_SetEntryWidths_NoUnits:
    }
}
%
%
%
%
% ========================================================================== %
%                                Typeset Entry                               %
% ========================================================================== %
%
\coffin_new:N \l_tmpc_coffin
%
\cs_new:Nn \UWMad_Nomenclature_SetEntry_NoUnits:nn {
% Set the entry material in the temporary coffins
    \vcoffin_set:Nnn
        \l_tmpa_coffin
        {\l__UWMad_Nomenclature_Entry_WidthSymbol_dim}
        {#1}
    \vcoffin_set:Nnn
        \l_tmpb_coffin
        {\l__UWMad_Nomenclature_Entry_WidthDescription_dim}
        {#2}
%
% Typeset the material and skips
    \group_begin:
        \setstretch{1.1}
        \skip_horizontal:n {\l__UWMad_Nomenclature_Entry_MarginLeft_dim}
        \coffin_typeset:Nnnnn \l_tmpa_coffin {l}{t}{0pt}{0pt}
        \skip_horizontal:n {\l__UWMad_Nomenclature_Entry_Padding_dim}
        \coffin_typeset:Nnnnn \l_tmpb_coffin {l}{t}{0pt}{0pt}
```

```
        \skip_vertical:n {\l__UWMad_Nomenclature_Entry_MarginBottom_dim}
    \group_end:
}
%
\cs_new:Nn \UWMad_Nomenclature_SetEntry_Units:nnn {
% Set the entry material in the temporary coffins
    \vcoffin_set:Nnn
        \l_tmpa_coffin
        {\l__UWMad_Nomenclature_Entry_WidthSymbol_dim}
        {#1}
    \vcoffin_set:Nnn
        \l_tmpb_coffin
        {\l__UWMad_Nomenclature_Entry_WidthUnit_dim}
        {#2}
    \vcoffin_set:Nnn
        \l_tmpc_coffin
        {\l__UWMad_Nomenclature_Entry_WidthDescription_dim}
        {#3}
%
% Typeset the material and skips
    \group_begin:
        \setstretch{1.1}
        \skip_horizontal:n {\l__UWMad_Nomenclature_Entry_MarginLeft_dim}
        \coffin_typeset:Nnnnn \l_tmpa_coffin {l}{t}{0pt}{0pt}
        \skip_horizontal:n {\l__UWMad_Nomenclature_Entry_Padding_dim}
        \coffin_typeset:Nnnnn \l_tmpb_coffin {l}{t}{0pt}{0pt}
        \skip_horizontal:n {\l__UWMad_Nomenclature_Entry_Padding_dim}
        \coffin_typeset:Nnnnn \l_tmpc_coffin {l}{t}{0pt}{0pt}
        \skip_vertical:n {\l__UWMad_Nomenclature_Entry_MarginBottom_dim}
    \group_end:
}
%
\cs_new:Nn \UWMad_Nomenclature_SetEntry: {
    \bool_if:NTF \g__UWMad_Nomenclature_IncludeUnitsColumn_bool {
        \UWMad_Nomenclature_SetEntry_Units:
    } {
        \UWMad_Nomenclature_SetEntry_NoUnits:
    }
}
%
%
%
%
%
% ============================================================================ %
%                                User Front-Ends                               %
% ============================================================================ %
\DeclareDocumentCommand \NomenclatureSetup { m } {
    \keys_set:nn { UWMad/Nomenclature } { #1 }
}
%
%
\DeclareDocumentEnvironment {Nomenclature} { o g } {
%
%   Create the ListOf
    \UWMad_ListOf_Define:n {Nomenclature}
```

```
%
%
%   Check for an optional section declaration and
%   set Main section token list.
    \IfNoValueTF {#1} { } {
        \UWMad_IfSectionExists:nT {#1} { }
        \tl_set:Nn \l__UWMad_Nomenclature_Section_Main_tl {#1}
    }
%
%
%   Check for an optional section declaration and
%   set Main section token list.
    \IfNoValueTF {#2} { } {
        \tl_set:Nf
            \l__UWMad_Nomenclature_Title_Main_tl {#2}
    }
    \UWMad_ListOf_SetTitle_Main:nn {Nomenclature}
        {\l__UWMad_Nomenclature_Title_Main_tl}
%
%
   \bool_if:NTF \g__UWMad_Nomenclature_IsNumbered_bool {
        \UWMad_ListOf_MakeNumbered:n    {Nomenclature}
    }{
        \UWMad_ListOf_MakeNotNumbered:n {Nomenclature}
    }
%
%
%
%   If Group section token list is empty, set it to the following
%   section after Main in the LaTeX sectioning hierarchy.
%   Otherwise, take the value at its word.
    \tl_if_empty:NTF \l__UWMad_Nomenclature_Section_Group_tl {
        \tl_set:Nf \l__UWMad_Nomenclature_Section_Group_tl {
            \UWMad_NextSection:n {
                \l__UWMad_Nomenclature_Section_Main_tl
            }
        }
    } { }
%
%   If Subgroup section token list is empty, set it to the following
%   section after Group in the LaTeX sectioning hierarchy.
%   Otherwise, take the value at its word.
    \tl_if_empty:NTF \l__UWMad_Nomenclature_Section_Subgroup_tl {
        \tl_set:Nf \l__UWMad_Nomenclature_Section_Subgroup_tl {
            \UWMad_NextSection:n {
                \l__UWMad_Nomenclature_Section_Group_tl
            }
        }
    } { }
%
%   Set the sections with the Nomenclature ListOf instance
    \UWMad_ListOf_SetSection_Main:nn
        {Nomenclature} {\l__UWMad_Nomenclature_Section_Main_tl}
    \UWMad_ListOf_SetSection_Group:nn
        {Nomenclature} {\l__UWMad_Nomenclature_Section_Group_tl}
    \UWMad_ListOf_SetSection_Subgroup:nn
```

```
        {Nomenclature} {\l__UWMad_Nomenclature_Section_Subgroup_tl}
%
%
%   Determine if this nomenclature should be in the Table of Contents
    \bool_if:NTF \g__UWMad_Nomenclature_IncludeInTOC_bool {
        \UWMad_ListOf_IncludeInTOC:n {Nomenclature}
    } {
        \UWMad_ListOf_DoNotIncludeInTOC:n {Nomenclature}
    }
%
%   Set some hooks in the Nomenclature ListOf instance
    \UWMad_ListOf_SetHook:nnn {Nomenclature} {PrePrint} {
        \UWMad_Nomenclature_SetEntryWidths:
    }
    \UWMad_ListOf_SetHook:nnn {Nomenclature} {PostPrint} {
        \UWMad_Nomenclature_ZeroWidest_Symbol:
    }
%
%
%   User front-end for creating a Group
    \DeclareDocumentCommand \Group { G{} } {
        \IfNoValueTF {##1} { } {
            \tl_set:Nn
                \l__UWMad_Nomenclature_Title_Group_tl {##1}
        }
        \UWMad_ListOf_SetTitle_Group:nn {Nomenclature}
            {\l__UWMad_Nomenclature_Title_Group_tl}
        \UWMad_ListOf_StartGroup:nn{Nomenclature}{Group}
    }
%
%   User front-end for creating a Subgroup
    \DeclareDocumentCommand \Subgroup { G{} } {
        \IfNoValueTF {##1} { } {
            \tl_set:Nn
                \l__UWMad_Nomenclature_Title_Subgroup_tl {##1}
        }
        \UWMad_ListOf_SetTitle_Subgroup:nn {Nomenclature}
            {\l__UWMad_Nomenclature_Title_Subgroup_tl}
        \UWMad_ListOf_StartGroup:nn{Nomenclature}{Subgroup}
    }
%
%   User front-end for creating an entry
    \bool_if:NTF \g__UWMad_Nomenclature_IncludeUnitsColumn_bool {
        \DeclareDocumentCommand \Entry { m m m } {
            \UWMad_ListOf_PushEntry:nn {Nomenclature} {
                \UWMad_Nomenclature_SetEntry_Units:nnn
                    {##1} {##2} {##3}
            }
            \UWMad_Nomenclature_UpdateWidest_Symbol:n{##1}
        }
    } {
        \DeclareDocumentCommand \Entry { m m } {
            \UWMad_ListOf_PushEntry:nn {Nomenclature} {
                \UWMad_Nomenclature_SetEntry_NoUnits:nn
                    {##1} {##2}
            }
```

```
            \UWMad_Nomenclature_UpdateWidest_Symbol:n{##1}
        }
    }
%
%   User front-end for reseting the column width
    \DeclareDocumentCommand  \ResetColumnWidth { } {
        \UWMad_Nomenclature_ZeroWidest_Symbol:
        \UWMad_Nomenclature_ZeroWidest_Unit:
    }
%
%
%   Print the main section title
    \UWMad_ListOf_PrintTitle:nn {Nomenclature}{Main}
%
} {
%   Flush the remaining entries from the ListOf queue and
%   delete the Nomenclature ListOf instance.
    \UWMad_ListOf_PrintEntries:n {Nomenclature}
    \UWMad_ListOf_Delete:n{Nomenclature}
}
%
%
%
%
\DeclareDocumentEnvironment {Acronym} { o G{Acronym} } {

    \begin{Nomenclature}[#1]{#2}
%
%
    \UWMad_Hash_Define:n{Acronyms}
    \UWMad_Hash_Define:n{AcronymMeanings}
%
%
    \cs_undefine:N \Entry
    \DeclareDocumentCommand \Entry { o m m } {
        \IfNoValueTF {##1} {

            \UWMad_Hash_Set:nnn{Acronyms}       {##2}{##2}
            \UWMad_Hash_Set:nnn{AcronymMeanings}{##2}{##3}
            \bool_new:c {g__UWMad_Acronym_WasSet_##2_bool}
            %
            \UWMad_ListOf_PushEntry:nn {Nomenclature} {
                \hypertarget{Acronym:##2}{}
                \UWMad_Nomenclature_SetEntry_NoUnits:nn
                    {##2} {##3}
            }

        } {

            \UWMad_Hash_Set:nnn{Acronyms}       {##1}{##2}
            \UWMad_Hash_Set:nnn{AcronymMeanings}{##1}{##3}
            \bool_new:c {g__UWMad_Acronym_WasSet_##1_bool}
            %
            \UWMad_ListOf_PushEntry:nn {Nomenclature} {
                \hypertarget{Acronym:##1}{}
                \UWMad_Nomenclature_SetEntry_NoUnits:nn
```

```
                    {##2} {##3}
            }

        }
        \UWMad_Nomenclature_UpdateWidest_Symbol:n{##2}
    }
} {

    \end{Nomenclature}

}

\DeclareDocumentCommand \Acro { m } {
    \UWMad_Hash_IfKeySet:nnTF {Acronyms} {#1} {
        \bool_if:cTF {g__UWMad_Acronym_WasSet_#1_bool} {
            \hyperlink{Acronym:#1}{
                \UWMad_Hash_Get:nn{Acronyms}{#1}
            }
        } {
            \UWMad_Hash_Get:nn{AcronymMeanings}{#1}~
                (
                    \UWMad_Hash_Get:nn{Acronyms}{#1}
                )
            \bool_gset_true:c {g__UWMad_Acronym_WasSet_#1_bool}
        }
    } { }
}
%
%
\cs_new_eq:NN \AcronymSetup \NomenclatureSetup
%
%
%
```

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`.

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

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

```
\cs_new:Nn \UWMad_MetaData_PushToList:nn {
   \clist_gput_right:Nn \g__UWMad_MetaDataList_clist {
        #1={#2}
    }
}
```

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

```
\bool_new:N \g__UWMad_MetaData_GenerateAux_bool
\bool_new:N \g__UWMad_MetaData_IsDocument_bool
```

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

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

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.

```
\AtBeginDocument{
    \clist_if_empty:NTF \g__UWMad_MetaDataList_clist { } {
        \exp_args:Nx \hypersetup {
            \clist_use:Nn\g__UWMad_MetaDataList_clist{,}
        }
    } { }
    \bool_gset_true:N \g__UWMad_MetaData_IsDocument_bool
}
```

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

```
\AtEndDocument{
    \bool_if:NTF \g__UWMad_MetaData_GenerateAux_bool {
        \clist_if_empty:NTF \g__UWMad_MetaDataList_clist { } {
            \iow_new:N   \g__UWMad_PDFMetaData_HyperSetup_io
            \iow_open:Nn \g__UWMad_PDFMetaData_HyperSetup_io {
                \c_job_name_tl.UWMad.PDFMetaData.aux
            }
            \iow_now:Nx  \g__UWMad_PDFMetaData_HyperSetup_io {
                \noexpand\ExplSyntaxOff
                    \noexpand\hypersetup
                    {\clist_use:Nn\g__UWMad_MetaDataList_clist{,}}
                \noexpand\ExplSyntaxOn
            }
            \iow_close:N \g__UWMad_PDFMetaData_HyperSetup_io
        } { }
    } { }
}
```

7.2 Thesis Information

Declare the `ThesisInfo` token list variables.

```
\tl_new:N \g__UWMad_ThesisInfo_Title_tl
\tl_new:N \g__UWMad_ThesisInfo_Author_tl
\tl_new:N \g__UWMad_ThesisInfo_DefenseDate_tl
\tl_new:N \g__UWMad_ThesisInfo_Department_tl
\tl_new:N \g__UWMad_ThesisInfo_Program_tl
\tl_new:N \g__UWMad_ThesisInfo_Degree_tl
\tl_new:N \g__UWMad_ThesisInfo_DocumentType_tl
\tl_new:N \g__UWMad_ThesisInfo_AdvisorName_tl
\tl_new:N \g__UWMad_ThesisInfo_AdvisorPosition_tl
\tl_new:N \g__UWMad_ThesisInfo_AdvisorAssociation_tl
\tl_new:N \g__UWMad_ThesisInfo_AdvisorMarker_tl
\tl_new:N \g__UWMad_ThesisInfo_Institution_tl
```

Set the document type default.

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

Define some booleans for required information/

```
\bool_new:N \g__UWMad_ThesisInfo_IsSet_Title_bool
\bool_new:N \g__UWMad_ThesisInfo_IsSet_Author_bool
\bool_new:N \g__UWMad_ThesisInfo_IsSet_DefenseDate_bool
\bool_new:N \g__UWMad_ThesisInfo_IsSet_Program_bool
\bool_new:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
\bool_new:N \g__UWMad_ThesisInfo_IsSet_Institution_bool
\bool_new:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool
```

Declare the user front-end for the title.

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

Set the associated token list variable

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

Pass it to the default LaTeX `\title` command.

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

Push the value to the MetaData `clist`.

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

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

```
    \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
        \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
    } { }
```

Tell the class this variable is now set.

```
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Title_bool
}
```

Similar flow to the `\Title` defintion.

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

```
    \UWMad_MetaData_PushToList:nn{pdfauthor}   {#1}
    \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
        \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
    } { }
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Author_bool
}
```

A simple setter command.

```
\DeclareDocumentCommand \Program { m } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_Program_tl {#1}
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Program_bool
}
```

A simple setter command.

```
\DeclareDocumentCommand \Degree { m } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_Degree_tl {#1}
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
}
```

A simple setter command.

```
\DeclareDocumentCommand \DocumentType { m } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {#1}
}
```

A semantic setter command.

```
\DeclareDocumentCommand \Dissertation { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
        dissertation
    }
}
```

A semantic setter command.

```
\DeclareDocumentCommand \DoctoralThesis { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
        doctoral~thesis
    }
}
```

A semantic setter command.

```
\DeclareDocumentCommand \MastersThesis { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
```

```
        master's~thesis
    }
}
```

A semantic setter command.

```
\DeclareDocumentCommand \Thesis { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
        thesis
    }
}
```

A semantic setter command.

```
\DeclareDocumentCommand \Prelim { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
        preliminary~report
    }
}
```

A simple setter command and aliases.

```
\DeclareDocumentCommand \DefenseDate { m } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DefenseDate_tl {#1}
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_DefenseDate_bool
}
\cs_gset_eq:NN \DefenceDate \DefenseDate
```

A simple setter command and aliases.

```
\DeclareDocumentCommand \Institution { m } {
    \tl_gset:Nn       \g__UWMad_ThesisInfo_Institution_tl {#1}
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Institution_bool
}
\cs_set_eq:NN \University \Institution
%
%
%
%
%
%   User front-ends (Optional)
\DeclareDocumentCommand \Department { m } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_Department_tl {#1}
}
\DeclareDocumentCommand \Advisor { m m m } {
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool
    \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorName_tl        {#1}
    \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorPosition_tl    {#2}
    \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorAssociation_tl {#3}
    \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorMarker_tl      {Advisor}
```

```
 }
\DeclareDocumentCommand \Adviser { m m m } {
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool
    \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorName_tl        {#1}
    \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorPosition_tl    {#2}
    \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorAssociation_tl {#3}
    \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorMarker_tl      {Adviser}
}
%
%
%
\msg_new:nnn { UWMadThesis } { ThesisInfo / UnsetInformation } {
    The~required~information~for~the~#1~is~not~set.
}
\DeclareDocumentCommand \IfInfoIsSetT { m +m } {
    \bool_if:cTF {g__UWMad_ThesisInfo_IsSet_ #1 _bool} {
        #2
    } {
        \msg_error:nnn
            { UWMadThesis }
            { ThesisInfo / UnsetInformation }
            {#1}
    }
}
%
%
%
%
%   User front-end accessors
\DeclareDocumentCommand \TheTitle { } {
    \g__UWMad_ThesisInfo_Title_tl
}
\DeclareDocumentCommand \TheAuthor { } {
    \g__UWMad_ThesisInfo_Author_tl
}
\DeclareDocumentCommand \TheProgram { } {
    \g__UWMad_ThesisInfo_Program_tl
}
\DeclareDocumentCommand \TheDegree { } {
    \g__UWMad_ThesisInfo_Degree_tl
}
\DeclareDocumentCommand \TheDocumentType { } {
    \g__UWMad_ThesisInfo_DocumentType_tl
}
\DeclareDocumentCommand \TheDefenseDate { } {
    \g__UWMad_ThesisInfo_DefenseDate_tl
}
\cs_gset_eq:NN \TheDefenceDate \TheDefenseDate
\DeclareDocumentCommand \TheInstitution { } {
    \g__UWMad_ThesisInfo_Institution_tl
}
\cs_set_eq:NN \TheUniversity \TheInstitution
%
\DeclareDocumentCommand \TheDepartment { } {
    \g__UWMad_ThesisInfo_Department_tl
}
```

```
\DeclareDocumentCommand \TheAdvisor { } {
    \g__UWMad_ThesisInfo_AdvisorName_tl
}
%
%
%
\int_new:N \l__UWMad_ThesisInfo_CommitteeCount_int
\UWMad_ListOf_Define:n {CommitteeList}
\DeclareDocumentCommand \CommitteeMember { m m m } {
    \int_incr:N \l__UWMad_ThesisInfo_CommitteeCount_int
    \UWMad_ListOf_PushEntry:nn {CommitteeList} {
        #1,~#2,~#3\skip_vertical:n{-1em}
    }
}
\DeclareDocumentCommand \PrintCommitteeMemberList { } {
    \bool_if:NTF \g__UWMad_ThesisInfo_IsSet_Advisor_bool {
         \g__UWMad_ThesisInfo_AdvisorName_tl{},~
         \g__UWMad_ThesisInfo_AdvisorPosition_tl{},~
         \g__UWMad_ThesisInfo_AdvisorAssociation_tl{}~
        (\g__UWMad_ThesisInfo_AdvisorMarker_tl{})
        \skip_vertical:n{-1em}
    } { }
    \UWMad_ListOf_PrintEntries:n {CommitteeList}
}
%
%
%
%
%
\tl_new:N \g__UWMad_PDFMetaData_Subject_tl
\tl_new:N \g__UWMad_PDFMetaData_Keywords_tl
\tl_new:N \g__UWMad_PDFMetaData_Producer_tl
\tl_new:N \g__UWMad_PDFMetaData_Creator_tl
%
%
%
%   User front-end (Optional)
\DeclareDocumentCommand \Subject { m } {
    \tl_gset:Nn \g__UWMad_PDFMetaData_Subject_tl {#1}
    \UWMad_MetaData_PushToList:nn{pdfsubject}  {#1}
    \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
        \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
    } { }
}
\DeclareDocumentCommand \Keywords { m } {
    \tl_gset:Nn \g__UWMad_PDFMetaData_Keywords_tl {#1}
    \UWMad_MetaData_PushToList:nn{pdfproducer} {#1}
    \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
        \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
    } { }
}
\DeclareDocumentCommand \Producer { m } {
    \tl_gset:Nn \g__UWMad_PDFMetaData_Producer_tl {#1}
    \UWMad_MetaData_PushToList:nn{pdfcreator}  {#1}
    \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
        \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
```

```
    } { }
}
\DeclareDocumentCommand \Creator { m } {
    \tl_gset:Nn \g__UWMad_PDFMetaData_Creator_tl {#1}
    \UWMad_MetaData_PushToList:nn{pdfkeywords} {#1}
    \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
        \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
    } { }
}
%
%
%
%   User front-end accessors.
\DeclareDocumentCommand \TheSubject { } {
    \g__UWMad_PDFMetaData_Subject_tl
}
\DeclareDocumentCommand \TheKeywords { } {
    \g__UWMad_PDFMetaData_Keywords_tl
}
\DeclareDocumentCommand \TheProducer { } {
    \g__UWMad_PDFMetaData_Producer_tl
}
\DeclareDocumentCommand \TheCreator { } {
    \g__UWMad_PDFMetaData_Creator_tl
}
%
```

Module 8

Special Pages

8.1 MakeTitlePage

```
% That phrase that occurs on every title page design the class author has seen
\DeclareDocumentCommand \FulfillmentClause { } {
    {
    \setstretch{1.1}
    A~\TheDocumentType{}~submitted~in~partial~fulfillment~of~the~
    requirements~for~the~degree~of
    }
}

\DeclareDocumentCommand \TitlePageTitle { } {
    \IfInfoIsSetT {Title} {
        {
            \LARGE
            \textsc {\TheTitle}
        }
    }
}

\DeclareDocumentCommand \TitlePageAuthor { } {
    \IfInfoIsSetT {Author} {
        {
            \large
            by  \\[0.50em]
            \TheAuthor{}
        }
    }
}

\DeclareDocumentCommand \TitlePageFulFillment { } {
    \FulfillmentClause{}
}

\DeclareDocumentCommand \TitlePageDegree { } {
    \IfInfoIsSetT {Degree} {
        \TheDegree{}
    }
}

\DeclareDocumentCommand \TitlePageProgram { } {
```

```
    \IfInfoIsSetT {Program} {
        \TheProgram{}
    }
}

\DeclareDocumentCommand \TitlePageInstitution { } {
    \IfInfoIsSetT {Institution} {
        \textsc{\TheInstitution{}}
    }
}

\DeclareDocumentCommand \TitlePageDefenseDate { } {
    \IfInfoIsSetT {DefenseDate} {
        Date~of~final~oral~examination:~\TheDefenseDate{}
    }
}


\DeclareDocumentCommand \MakeTitlePage { } {
    \clearpage
    \thispagestyle{empty}
    \begin{center}
        \TitlePageTitle{}       \\[1.0em]
        \TitlePageAuthor{}      \\[1.0em]
        \vfill
        \TitlePageFulFillment{} \\[1.0em]
        \TitlePageDegree{}      \\[1.0em]
        \TitlePageProgram{}     \\[1.0em]
        \vfill
        \TitlePageInstitution{}
        \vfill
    \end{center}
    \TitlePageDefenseDate{}\\[1.0em]
    \PrintCommitteeMemberList{}
    \cleardoublepage
}
```

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.

```
\cs_new:Nn \__UWMad_LicensePage_StartPage: {
    \clearpage
    \thispagestyle{empty}
```

```
    \tex_hbox:D{}
    \tex_vfill:D
    \phantomsection
}
%
```

8.2.1 Copyright

```
\bool_new:N   \l__UWMad_Copyright_UseCopyright_bool
\cs_set_eq:NN \CopyrightSymbol \copyright

\cs_set:Nn \__UWMad_Copyright_LicenseText: {
    \begin{center}
        Copyright~\CopyrightSymbol{}~
        \l__UWMad_LicensePage_Year_tl{}~
        by~
        \l__UWMad_LicensePage_Owner_tl{}
    \end{center}
}
%
%
%
%
```

8.2.2 Creative Commons

```
%   Token lists
\tl_new:N    \l__UWMad_CCLicense_Porting_tl
\tl_new:N    \l__UWMad_CCLicense_Version_tl
\tl_new:N    \l__UWMad_CCLicense_TypeAbbreviation_tl
\tl_new:N    \l__UWMad_CCLicense_TypeWords_tl
\tl_new:N    \l__UWMad_CCLicense_URL_Front_tl
\tl_new:N    \l__UWMad_CCLicense_URL_Middle_tl
\tl_new:N    \l__UWMad_CCLicense_URL_Back_tl
\tl_new:N    \l__UWMad_CCLicense_URL_tl
\tl_new:N    \l__UWMad_CCLicense_http_tl
\tl_new:N    \l__UWMad_CCLicense_URLText_tl
%
%   Booleans
\bool_new:N \l__UWMad_CCLicense_UseCreativeCommons_bool
\bool_new:N \l__UWMad_CCLicense_UseAttribution_bool
\bool_new:N \l__UWMad_CCLicense_UseShareAlike_bool
\bool_new:N \l__UWMad_CCLicense_UseNoDerivatives_bool
\bool_new:N \l__UWMad_CCLicense_UseNonCommercial_bool
\bool_new:N \l__UWMad_CCLicense_IsValid_bool
```

```
\bool_set_true:N \l__UWMad_CCLicense_UseAttribution_bool
%
%   Valid license types
\cs_new:cn {l__UWMad_CCLicense_Valid_ by :}      {}
\cs_new:cn {l__UWMad_CCLicense_Valid_ by-sa :}   {}
\cs_new:cn {l__UWMad_CCLicense_Valid_ by-nd :}   {}
\cs_new:cn {l__UWMad_CCLicense_Valid_ by-nc :}   {}
\cs_new:cn {l__UWMad_CCLicense_Valid_ by-nc-sa :}{}
\cs_new:cn {l__UWMad_CCLicense_Valid_ by-nc-nd :}{}
%
%   Defaults
\tl_gset:Nn \l__UWMad_CCLicense_Porting_tl {
    International
}
\tl_gset:Nn \l__UWMad_CCLicense_Version_tl {
    4.0
}
%
%   URL definitions
\tl_set:Nn \l__UWMad_CCLicense_URL_Front_tl {
    creativecommons.org/licenses
}
\tl_set:Nn \l__UWMad_CCLicense_URL_Middle_tl {
    /\l__UWMad_CCLicense_TypeAbbreviation_tl
}
\tl_set:Nn \l__UWMad_CCLicense_URL_Back_tl {
    /\l__UWMad_CCLicense_Version_tl
}
\tl_set:Nn \l__UWMad_CCLicense_URL_tl {
    http://
    \l__UWMad_CCLicense_URL_Front_tl
    \l__UWMad_CCLicense_URL_Middle_tl
    \l__UWMad_CCLicense_URL_Back_tl
}
\tl_set:Nn \l__UWMad_CCLicense_http_tl {
    http://
}
%
%
\tl_set:Nn \l__UWMad_CCLicense_URLText_tl {
    Creative~Commons~
    \l__UWMad_CCLicense_TypeWords_tl{}~
    \l__UWMad_CCLicense_Version_tl{}~
    \l__UWMad_CCLicense_Porting_tl{}
}
%
%
%
%   Type Creator
\cs_new:Nn \__UWMad_CCLicense_CreateType: {

        \bool_if:NTF \l__UWMad_CCLicense_UseAttribution_bool {

            \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
                by
            }
```

```
            \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
                Attribution
            }

        } { }

        \bool_if:NTF \l__UWMad_CCLicense_UseNonCommercial_bool {

            \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
                -nc
            }
            \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
                -NonCommercial
            }

        } { }

        \bool_if:NTF \l__UWMad_CCLicense_UseShareAlike_bool {

            \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
                -sa
            }
            \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
                -ShareAlike
            }

        } { }

        \bool_if:NTF \l__UWMad_CCLicense_UseNoDerivatives_bool {

            \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
                -nd
            }
            \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
                -NoDerivatives
            }

        } { }
}
%
%
%
%   Type Validator
\cs_new:Nn \__UWMad_CCLicense_CheckTypeValidity: {
    \cs_if_exist:cTF {
        l__UWMad_CCLicense_Valid_
        \l__UWMad_CCLicense_TypeAbbreviation_tl :
    } {

        \bool_set_true:N \l__UWMad_CCLicense_IsValid_bool

    } {

        \msg_new:nnn {UWMadThesis} {CCLicense / InvalidLicenseType} {
            The~license~type~`\l__UWMad_CCLicense_TypeAbbreviation_tl'~
            is~not~a~valid~Creative~Commons~license.
```

```
        }
        \msg_error:nn {UWMadThesis} {CCLicense / InvalidLicenseType}

    }
}
%
%
%
%   Page Printer
\cs_new:Nn \__UWMad_CCLicense_LicenseText: {
    \begin{center}
        \setstretch{1.05}
        This~work~is~released~under~a~
        \href {\l__UWMad_CCLicense_URL_tl} {
            \l__UWMad_CCLicense_URLText_tl
        }~
        license.\\[0.1em]
        \l__UWMad_LicensePage_Owner_tl{},~
        \l__UWMad_LicensePage_Year_tl{}
    \end{center}
}
%
```

8.2.3 LicensePage Proper

```
%
    \tl_new:N  \l__UWMad_LicensePage_Year_tl
    \tl_new:N  \l__UWMad_LicensePage_Owner_tl
%
    \tl_set:Nn \l__UWMad_LicensePage_Owner_tl {
        \g__UWMad_ThesisInfo_Author_tl
    }
    \tl_set:Nn \l__UWMad_LicensePage_Year_tl {
        \the\year
    }
%
%
%
\DeclareDocumentEnvironment {LicensePage} { } {
%
%
%
    \DeclareDocumentCommand \LicenseOwner { m } {
        \tl_set:Nn \l__UWMad_LicensePage_Owner_tl {
            ##1
        }
    }
    \DeclareDocumentCommand \TheLicenseOwner { } {
        \l__UWMad_LicensePage_Owner_tl
    }
%
```

```
    \DeclareDocumentCommand \LicenseYear { m } {
        \tl_set:Nn \l__UWMad_LicensePage_Year_tl {
            ##1
        }
    }
    \DeclareDocumentCommand \TheLicenseYear { } {
        \l__UWMad_LicensePage_Year_tl
    }
%
%
%
\DeclareDocumentCommand \Copyright { } {
    \bool_set_true:N \l__UWMad_Copyright_UseCopyright_bool
}
\cs_set_eq:NN \AllRightsReserved \Copyright
%
%
%
%   User front ends
\DeclareDocumentCommand \CreativeCommons { } {
    \bool_set_true:N \l__UWMad_CCLicense_UseCreativeCommons_bool
}
\DeclareDocumentCommand \Attribution { } {
    \bool_set_true:N \l__UWMad_CCLicense_UseAttribution_bool
}
\DeclareDocumentCommand \NonCommercial { } {
    \bool_set_true:N \l__UWMad_CCLicense_UseNonCommercial_bool
}
\DeclareDocumentCommand \ShareAlike { } {
    \bool_set_true:N \l__UWMad_CCLicense_UseShareAlike_bool
}
\DeclareDocumentCommand \NoDerivs { } {
    \bool_set_true:N \l__UWMad_CCLicense_UseNoDerivatives_bool
}
%
%
\DeclareDocumentCommand \CCVersion { m } {
    \tl_set:Nn \l__UWMad_CCLicense_Version_tl {##1}
}
%
\DeclareDocumentCommand \CCPorting { m } {
    \tl_set:Nn \l__UWMad_CCLicense_Porting_tl {##1}
}
%
\DeclareDocumentCommand \CCURL { m } {
    \tl_set:Nn \l__UWMad_CCLicense_URL_Front_tl  {##1}
    \tl_set:Nn \l__UWMad_CCLicense_URL_Middle_tl {/.}
    \tl_set:Nn \l__UWMad_CCLicense_URL_Back_tl   {}
}
%
\DeclareDocumentCommand \CCURLText { m } {
    \tl_set:Nn \l__UWMad_CCLicense_URLText_tl {##1}
}
%
%
} {
```

```
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

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.

D

E

F

G

I

K

L

M

N

O

P

R

S

V

X