`UWMadThesis` Class Manual

Troy Christopher Haskin

Table of Contents

Part I

User Guide

The `UWMadThesis` class is aimed at providing a $\LaTeX\,2_\varepsilon$ 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 PDFs are available

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. If any of these commands is not present, usage of the Title Page command will throw an error. It is encouraged to use these commands in the preamble of the document.

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

Each of these commands must be used once; if not, their respective variables will be empty and usage of the They can, of course, be used more than once, but the additional uses would only redefine the value of the associated variable.

```
\Degree      \Degree  {⟨degree⟩}
\Doctorate   \Doctorate
\Masters     \Masters
\Bachelors   \Bachelors
```

Only one of these commands is required to define the `{`⟨*degree*⟩`}` variable. The generic `\Degree` function will accept any valid text or expandable content for defining the degree variable.

The other three commands take no argument and are semantic commands for defining the degree variable:

- `\Doctorate` sets `{`⟨*degree*⟩`}` to “Doctor of Philosophy”
- `\Masters` sets `{`⟨*degree*⟩`}` to “Master’s”
- `\Bachelors` sets `{`⟨*degree*⟩`}` to “Bachelor’s”

```
\DefenseDate  \DefenseDate {⟨defense date⟩}
\DefenceDate  \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, any valid text or expandable argument may be entered 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⟩}  {⟨member program⟩}
\Advisor          \Advisor         {⟨advisor name⟩} {⟨advisor position⟩} {⟨advisor program⟩}
\Adviser          \Adviser         {⟨advisor name⟩} {⟨advisor position⟩} {⟨advisor program⟩}
```

`\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.

Using either the `\Advisor` or `\Adviser` 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. Either of these commands are not required but semantic in nature.

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⟩}
\Dissertation     \Dissertation
\DoctoralThesis   \DoctoralThesis
\MastersThesis    \MastersThesis
\Thesis           \Thesis
\Prelim           \Prelim
```

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.

To facilitate good semantic mark-up, some prepared commands to set the document type were made. These commands take no argument and set the value of `{`⟨*document type*⟩`}` to something 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"

```
\Subject   \Subject  {⟨document subject⟩}
\Keywords  \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 replacement for the default `\maketitle`. Per the example provided by the UW–Madison Graduate School's sample, the title page flows (in order): report 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.

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
\LicenseYear

```
\LicenseOwner {⟨owner name⟩}
\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 licenses are copyright licenses in that (aside from CC0) the 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. Of course, this choice can be circumvented.

`\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` `\Attribution`
`\ShareAlike` `\ShareAlike`
`\NonCommercial` `\NonCommercial`
`\NoDerivs` `\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, 2018

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

```
\CCVersion   \CCVersion{⟨CC version⟩}
\CCPorting   \CCPorting{⟨CC porting⟩}
\CCURL       \CCURL    {⟨CC link⟩}
\CCURLText   \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):

This work is released under a Creative Commons Attribution 3.0 United States license.
Troy Christopher Haskin, 2018

Feature Set 3

Layout And Style

The `UWMadThesis` class has several default styling differences from the standard LaTeX 2$_\varepsilon$ 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 styles are list in this section or the references manuals.

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 showcased below by using the packages' utilities outlined 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 changed using the commands below or 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.

3.2.1 Link Colors

To more easily facilitate color changes to links, several user interface commands have been defined.

`\MakeLinksTheseColors`

`\MakeLinksTheseColors{⟨link color⟩}{⟨cite color⟩}{⟨url color⟩}`

Redefines the colors used for (internal) links, cites, and URLs. Any valid color, including those defined by the xcolor package, is allowed for all three, required arguments.

`\MakeLinksThisColor`

`\MakeLinksThisColor{⟨color⟩}`

Redefines the colors used for (internal) links, cites, and URLs to be the single indicated color. Any valid color, including those defined by the xcolor package, is allowed for the one required arguments.

`\MakeLinksBlack`
`\MakeLinksBlue`
`\MakeLinksRed`

`\MakeLinksBlack`
`\MakeLinksBlue`
`\MakeLinksRed`

These commands take no argument and define all links to have the color indicated in the command name.

3.2.2 References

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). The user is referred to their respective manuals for more options and feature descriptions.

3.3 Paragraph Spacing

In general, there are two dominant methods for indicating separate paragraphs: no indentation with extra space between paragraphs (compared to between lines) and indentation with no extra space between paragraphs. The default of the `UWMadThesis` class is the former but some may prefer the latter. To facilitate either, two options for the have been created.

Feature Set 4

Getting Started

4.1 Options On-Load

4.2 Feature Options

4.3 Identification Commands

Feature Set 5

Sectioning

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

5.1 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       \dedications      {⟨title⟩}
\acknowledgments   \acknowledgments {⟨title⟩}
\abstract          \abstract         {⟨title⟩}
\umiabstract       \umiabstract      {⟨title⟩}
\preface           \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".

5.2 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` command. The `UWMadThesis` 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 act like `\chapter` commands and are typeset in the Table of Contents as such.

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.

5.3 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 6

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.

6.1 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.

6.1.1 Command Descriptions

A sketch of the `Nomenclature` implementation would be:

```
\begin{Nomenclature}[⟨toc title⟩][⟨title⟩]
    \Entry{⟨symbol⟩}{⟨description⟩}
    \Group[⟨toc group⟩]{⟨group⟩}
        \Entry{⟨symbol⟩}{⟨description⟩}
        \Subgroup[⟨toc subgroup⟩]{⟨subgroup⟩}
            \Entry{⟨symbol⟩}{⟨description⟩}
\end{Nomenclature}
```

The square brace-delimited `[`⟨*toc title*⟩`]` is OPTIONAL and the overrides `[`⟨*title*⟩`]` argument for insertion into the table of contents. The square brace-delimited `[`⟨*title*⟩`]` is OPTIONAL and temporarily overrides the default title used for the nomenclature environment (Nomenclature). If only one optional argument is given, it is assumed that `[`⟨*title*⟩`]` was given and `[`⟨*toc title*⟩`]` is equal to the `[`⟨*title*⟩`]`. The curly brace-delimited `{`⟨*group*⟩`}` and `{`⟨*subgroup*⟩`}` are REQUIRED; the optional these arguments will override the titles in the table of contents.

`\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` `\Group{⟨group title⟩}`
`\Subgroup` `\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).

6.1.2 Examples

As an example, the following input

```
\begin{Nomenclature}[Symbol Table]
    \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{$\rho$}{Density}
    \Entry{$\mu$}{Viscosity}
\end{Nomenclature}
```

would be typeset as:

Symbol Table

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.
ρ	Density
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 adding the line `\Group{Greek Letters}` below the first entry:

Symbol Table

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.

Greek Letters

ρ	Density
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.

6.1.3 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 `\UWMadSetup` with the module name `Nomenclature`:

```
\UWMadSetup {
    Nomenclature / {
        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.

6.2 Acronym

6.2.1 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.

6.2.2 Example

The following input

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

```
\end{Acronym}
```

is typeset as

Acronym Table

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’.

6.2.3 Acronym Customization

Since this feature is an extension of the `Nomenclature` feature, it is customized in a similar fashion: using `\UWMadSetup` and the `Acronym` module name. It shares all of the same keys with some additional ones outline in table 3.

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

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

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 7

Math

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

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

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

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

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

7.2 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

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

\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

$$\operatorname*{Min}_{x\in\mathbb{R}} \operatorname{Sin}(x) = -1 \tag{12}$$

$$\operatorname*{Max}_{x\in\mathbb{R}} \operatorname{Sin}(x) = +1 \tag{13}$$

\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
        \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.

7.3 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 'T_P, T_P, T_P', and the input `` `T\subs[\!]{P}, T\subs[\:]{P}' `` is typeset as '$\mathrm{T}_{\!\mathrm{P}}$, $\mathrm{T}_{\:\mathrm{P}}$'. 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 8

Programming

The Programming 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.

Feature Set 9

Relative Directory Includes

LaTeX provides two commands for importing external files:

`\input` Simply adds the contents of the file to the input stream

`\include` Performs a `\clearpage` before and after the file inclusion; also allows selective inclusion through the `\includeonly` command.

They work well but do have one deficiency for longer documents: they lack directory awareness. For example, if a chapter file named `Chapter-1.tex` existed a sub-directory named `Chapter-1`, the required markup would b:

```
\input{Chapter-1/Chapter-1}
```

This seems reasonable. However, the complexity (or possibly annoyance) increases if other files are imported from `Chapter-1.tex`. If there was a section file `Section.tex` in the `Chapter-1` directory that was desired to be included by `Chapter-1.tex` (a somewhat intuitive idea: chapter files include section files), the markup would need to be

```
\input{Chapter-1/Section-1}
```

WITHIN the `Chapter-1.tex` file itself. For large documents where sections, or even subsections, become large enough that they require their own files, adding these directory trees can be become burdensome and lead to poor-looking markup.

The `UWMadThesis` class Relative Directory feature provides a mechanism to make this process easier and cleaner. Commands are added to form a ⟨*search stack*⟩ that is separate from the default LaTeX search path. These commands and the convention built into the system are discussed below.

9.1 File Inclusion

For including text files (i.e., not graphics files) the system operates through the usage of the following three commands.

```
\IncludeChapter        \IncludeChapter   [⟨path⟩]{⟨filename⟩}
\IncludeSection        \IncludeSection   [⟨path⟩]{⟨filename⟩}
\IncludeSubsection     \IncludeSubsection[⟨path⟩]{⟨filename⟩}
```

These commands will augment the class's current search path according the conventions outlined in the next section. The `{⟨filename⟩}` will then be searched for and, if found, added to the input stream. These commands are meant to be used following the standard LaTeX sectioning conventions: chapters then sections then subsections. While the system may work if used out-of-order, the behavior is not tested and should be avoided.

An optional `[⟨path⟩]` can be input to override the current Naming Conventions and is present for special circumstances.

At first, these commands seem to be simple renamings of the LaTeX system but with the path and file name having separate inputs. This stance is entirely true if directory Naming Conventions aren't used. But it is highly recommended that they are.

9.2 Naming Conventions

By default, there is no naming convention (referred to a `none` in the implementation). A naming convention is a pattern that tells the Relative Directory system how the directories that hold document files are named. Naming conventions are defined by the user through the `\UWMadSetup` function and the `RelativeDirectory` module name (see examples below).

By default, there are currently two supported naming conventions: increment and same.

More maybe added in the future.

9.2.1 Increment

Suppose a user has a LaTeX document that is to be compiled from a file named `Main.tex` that exists in the directory `Main`. The user also has several chapters and and sections with the directory structure seen in table 4a. Each of the directory names is prefixed with `Chapter-` or `Section-` and ended with an Arabic number. This directory structure exemplifies the Increment naming convention.

The user can easily tell the Relative Directory system of this convention using the following input

```
\UWMadSetup{
    RelativeDirectory / {
        chapter-directory-prefix = Chapter-,
        chapter-directory-name   = increment,
        section-directory-prefix = Section-,
        section-directory-name   = increment
    }
}
```

Then, using the commands above, the user can include the files by adding the following input to `Main.tex`:

```
\IncludeChapter{Chapter}
    \IncludeSection{Section-1}
    \IncludeSection{Section-2}
\IncludeChapter{Chapter}
    \IncludeSection{Section}
    \IncludeSection{Section}
```

Or, the user can choose to only add the chapters in `Main.tex` while putting the section includes in their respective `Chapter.tex` files. The `UWMadThesis` class ⟨*search stack*⟩ will handle either.

9.2.2 Same

Suppose a user has a LaTeX document that is to be compiled from a file named `Main.tex` that exists in the directory `Main`. The user also has several chapters and and sections with the directory structure seen in table 4b. Each of the directory names is suffixed with `-Chapter` or `-Section` and begins with the file name of at least one of its files. This directory structure exemplifies the Same naming convention. The user can easily tell the Relative Directory system of this convention using the following input

```
\UWMadSetup{
    RelativeDirectory / {
        chapter-directory-name   = same,
        chapter-directory-suffix = -Chapter,
        section-directory-name   = same,
        section-directory-suffix = -Section,
    }
}
```

Then, using the commands above, the user can include the files by adding the following input to `Main.tex`:

```
\IncludeChapter{Introduction}
    \IncludeSection{Motivation}
    \IncludeSection{LiteratureReview}
\IncludeChapter{Theory}
    \IncludeSection{LinearSystems}
    \IncludeSection{NewtonsMethod}
```

Or, the user can choose to only add the chapters in `Main.tex` while putting the section includes in their respective `Chapter.tex` files. The `UWMadThesis` class ⟨*search stack*⟩ will handle either.

9.2.3 None

`None` is the default naming convention used. This convention only forms a path from a concatenation of a section's prefix and suffix only. Without setting up one of the naming

Table 4: *Directory structure examples for naming conventions*

(a) *Increment*

```
/Main/
|-- Main.tex
|-- Chapter-1
| |-- Chapter.tex
| |-- Section-1.tex
| |-- Section-2.tex
|-- Chapter-2
| |-- Chapter.tex
| |-- Section-1
| | |-- Section.tex
| |-- Section-2
| | |-- Section.tex
|----
```

(b) *Same*

```
/Main/
|-- Main.tex
|-- Introduction-Chapter
| |-- Introduction.tex
| |-- Motivation.tex
| |-- LiteratureReview.tex
|-- Theory-Chapter
| |-- Theory.tex
| |-- LinearSystems-Section
| | |-- LinearSystems.tex
| |-- NewtonsMethod-Section
| | |-- NewtonsMethods.tex
|----
```

conventions described above, the system will require the optional argument for dynamic file searching to be possible. The `UWMadThesis` class ⟨*search stack*⟩ will be updated according to these given optional paths, so relative definitions are required. Also, this option can be used to create a static container directory be defining either the prefix or suffix (or both) to the static directory name; this can be useful for placing all section files into one directory instead of nesting them, for example.

9.3 Including Graphics

For including graphics files (i.e., not text files), the system operates through the usage of one of the following commands.

\IncludeGraphics
\includegraphics

`\IncludeGraphics[⟨options⟩]{⟨graphic name⟩}`
`\includegraphics[⟨options⟩]{⟨graphic name⟩}`

The `UWMadThesis` class augments the definition of the `\includegraphics` command (while adding 100% equivalent camel-cased version) to use the ⟨*search stack*⟩. The one difference in using these commands from the default behavior is that THE EXTENSION IS REQUIRED. These commands will follow the defined naming conventions and search the directories (from the lowest to highest) for the graphics file and input the first extant graphic matching the `{`⟨*graphic name*⟩`}`.

If a dedicated graphics directory is desired at MULTIPLE LEVELS, one can be defined through the `graphics-directory-name` option. If a dedicated graphics directory is desired at A SINGLE LEVEL, one can be defined through the `the-only-graphics-directory` option.

9.4 Search Controls

As mentioned above, the Relative Directory system builds a stack of directory paths and then searches them. The default behavior is different for files and graphics.

By default, files are only searched for in the lowest (i.e., most recently added) directory path and the main search path. This default was chosen such that similarly named files at higher directory levels are not mistakenly included. The default can be changed by setting the `cycle-file-paths` to `true`.

By default, graphics are searched for from lowest to highest directory and, if not found, in the main search path. This default was chosen such that the same graphic can be included across many input levels. The default can be changed by setting the `cycle-graphic-paths` to `false`.

9.5 User Options

Options for this feature are set using the `\UWMadSetup` command and `RelativeDirectory` module name. The input syntax has the form

```
\UWMadSetup {
        RelativeDirectory / {
            key-one = value-one,
            key-two = value-two,
            ...
            key-n   = value-n,
        }
    }
```

Table 5 lists all of the valid keys for this feature set.

TABLE 5: *List of key-value pairs for Relative Directory system.*

Key	Meaning	Default	Allowed value
chapter-directory-prefix	Directory prefix used for `\IncludeChapter`	—	text
chapter-directory-suffix	Directory suffix used for `\IncludeChapter`	—	text
chapter-directory-name	Naming convention used for `\IncludeChapter`	none	valid choice†
section-directory-prefix	Directory prefix used for `\IncludeSection`	—	text
section-directory-suffix	Directory suffix used for `\IncludeSection`	—	text
section-directory-name	Naming convention used for `\IncludeSection`	none	valid choice†
subsection-directory-prefix	Directory prefix used for `\IncludeSubsection`	—	text
subsection-directory-suffix	Directory suffix used for `\IncludeSubsection`	—	text
subsection-directory-name	Naming convention used for `\IncludeSubsection`	none	valid choice†
graphics-directory-name	Graphics directory name for multiple directories	—	text
the-only-graphics-directory	Graphics directory name for a single directory	—	text
cycle-file-paths	Search the entire file stack or only the lowest level	false	boolean
cycle-graphics-paths	Search the entire graphic stack or only the lowest level	true	boolean

† Valid choices: none, same, increment

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$_\varepsilon$ 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$_\varepsilon$ 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}[2018/05/13]
\@ifpackagelater{expl3}{2018/05/13} {} {%
    \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        {2018/05/29}
\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

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

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

Declare the options.

```
\DeclareOption{NoMath} {
    \bool_gset_false:N \g__UWMad_MathTweaks_bool
}
\DeclareOption{Quiet} {
    \msg_redirect_module:nnn { UWMadThesis } { warning } { none }
}
```

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

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

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

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

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.

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

1.4 Package Loads

Since the philosophy behind this class is to stand on the shoulders of giants, we now load packages that are either commonly loaded by others, deemed useful for the class user, or needed for the class author.

1.4.1 Hyperref Prior

Load some packages that give nice features and can be loaded before hyperref.

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

Load the $\mathcal{A}\mathcal{M}\mathcal{S}$ suite.

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

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

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

1.4.2 Hyperref Now

Load hyperref and bookmark.

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

1.4.3 Hyperref Forever

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}
```

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
}
%
```

1.5 Key-Value Interface

`\UWMadSetup` `\UWMadSetup{⟨option list⟩}`

This simple command creates a user interface for the key-value system used for several feature set options.

```
\cs_generate_variant:Nn \keys_set:nn {nf}
\tl_new:N \l__UWMad_Setup_ModuleName_tl
\clist_new:N \l__UWMad_Setup_OptionList_clist
\cs_new:Nn \__UWMad_Setup_ProcessInput:n {
    \seq_set_split:Nnn \l_tmpa_seq {,} {#1}
    \seq_map_inline:Nn \l_tmpa_seq {
        \seq_set_split:Nnn \l_tmpb_seq {/} {##1}
        \seq_pop:NN \l_tmpb_seq \l__UWMad_Setup_ModuleName_tl
        \seq_pop:NN \l_tmpb_seq \l__UWMad_Setup_OptionList_clist
        \clist_map_inline:Nn \l__UWMad_Setup_OptionList_clist {
            \tl_set:Nx \l_tmpa_tl {
                \l__UWMad_Setup_ModuleName_tl / \tl_trim_spaces:n{####1}
            }
            \exp_args:Nnf
                \keys_set:nn { UWMadThesis } { \l_tmpa_tl }
        }
    }
}
\DeclareDocumentCommand \UWMadSetup { m } {
    \__UWMad_Setup_ProcessInput:n{#1}
}
```

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$_\varepsilon$ 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:cn
\UWMad_Hook_Prepend:Nn

\UWMad_Hook_Prepend:cn {⟨*command name*⟩} {⟨*prepend code*⟩}
\UWMad_Hook_Prepend:Nn ⟨*command*⟩ {⟨*prepend code*⟩}

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

```
\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_generate_variant:Nn \UWMad_Hook_Prepend:Nn { cn }
```

\UWMad_Hook_Append:cn
\UWMad_Hook_Append:Nn

\UWMad_Hook_Append:cn {⟨*command name*⟩} {⟨*append code*⟩}
\UWMad_Hook_Append:Nn ⟨*command*⟩ {⟨*append code*⟩}

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

```
\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:}
}
\cs_generate_variant:Nn \UWMad_Hook_Append:Nn { cn }
```

`\UWMad_Definition_Swap:Nn`	`\UWMad_Definition_Swap:Nn ⟨command⟩ {⟨replacement code⟩}`
`\UWMad_Definition_Swap:cn`	`\UWMad_Definition_Reset:N ⟨command⟩`
`\UWMad_Definition_Reset:N`	`\UWMad_Definition_Swap:cn {⟨command name⟩} {⟨replacement code⟩}`
`\UWMad_Definition_Reset:c`	`\UWMad_Definition_Reset:c {⟨command name⟩}`

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_File_GetExtension:nNN` `\UWMad_File_GetExtension:nNN{⟨path⟩}⟨tl var 1⟩⟨tl var 2⟩`

Searches through the given {⟨*file path*⟩} for an extension identifier (. by default) in the path. If one is found, the path sans extension is assigned to ⟨*tl var 1*⟩ with the extension assigned to ⟨*tl var 2*⟩.

Initializations of variables and booleans used in the function

```
\tl_gset:Nn \g__UWMad_File_ExtensionMarker_tl {.}
\tl_gset:Nn \g__UWMad_File_DirectoryMarker_tl {/}
```

Define the body of the function.

```
\cs_new:Nn \UWMad_File_PathFileName:NNNn {

    %   Break possible directory chain and pop file name from right
    \seq_set_split:Nnn \l_tmpa_seq {g__UWMad_File_DirectoryMarker_tl} {#4}
    \seq_pop_right:NN  \l_tmpa_seq #2

    %   Check if a path was given
    \int_compare:nNnTF {\seq_count:N \l_tmpa_seq} > {0} {
        \tl_set:Nf #1 {
            \seq_use:Nn \l_tmpa_seq {\g__UWMad_File_DirectoryMarker_tl}
            \g__UWMad_File_DirectoryMarker_tl
        }
    } {
        \tl_clear:N #1
    }

    %   Check if an extension was given
    \tl_if_in:NnTF #2 {\g__UWMad_File_ExtensionMarker_tl} {
        \seq_set_split:NnV \l_tmpa_seq {\g__UWMad_File_ExtensionMarker_tl} {#2}
        \seq_get_right:NN  \l_tmpa_seq #3
        \tl_set:Nf #2 {
            \seq_use:Nn \l_tmpa_seq {\g__UWMad_File_DirectoryMarker_tl}
        }
    } {
        \tl_clear:N #3
    }
}
\cs_generate_variant:Nn \UWMad_File_PathFileName:NNNn {NNNx}
```

__UWMad_IfDefined:nnnnT
__UWMad_IfUndefined:nnnnT

__UWMad_IfDefined:nnnnT{⟨*Prefix*⟩}{⟨*ID*⟩}{⟨*Suffix*⟩}{⟨*Type*⟩}{⟨*Code*⟩}
__UWMad_IfUndefined:nnnnT{⟨*Prefix*⟩}{⟨*ID*⟩}{⟨*Suffix*⟩}{⟨*Type*⟩}{⟨*Code*⟩}

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.

```
\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

__UWMad_IfDefined:nT{⟨*CommandName*⟩}{⟨*TrueCode*⟩}
__UWMad_IfUndefined:nT{⟨*CommandName*⟩}{⟨*TrueCode*⟩}

These commands are simplifications of the above commands and that only take a {⟨*CommandName*⟩} and {⟨*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    \__UWMad_Stack_IfDefined:nT{⟨stack name⟩}{⟨true code⟩}
\__UWMad_Stack_IfUndefined:nT  \__UWMad_Stack_IfUndefined:nT{⟨stack name⟩}{⟨true code⟩}
```

Shortcuts for the more general commands outlined above.

```
\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`
`\__UWMad_Queue_IfUndefined:nT`

Shortcuts for the more general commands outlined above.

```
\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_Delete: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}
    }
}
```

JWMad_Deque_PopLeft:nn
JWMad_Deque_PopRight: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_item:Nn    { c x     }
\cs_generate_variant:Nn \prop_item: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` `\UWMad_Hash_Set:nnn{⟨HashID⟩}{⟨Key⟩}{⟨Value⟩}`

Set the value of a key of a defined Hash.

```
\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_item:cn {cf}
\cs_new:Nn \UWMad_Hash_Get:nn {
    \__UWMad_Hash_IfDefined:nT {#1} {
        \prop_item: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_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 2$_\varepsilon$-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

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

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.

```
\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

\IfStringEmpty{⟨*String*⟩}{⟨*True*⟩}{⟨*False*⟩}

Checks if a given string is composed of no characters or just blank spaces.

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

\IfCommandEmpty

\IfCommandEmpty{⟨*Command*⟩}{⟨*True*⟩}{⟨*False*⟩}

Determines if a commands contains no or only space after one expansion.

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

2.3.2 Command Creators

\MakeCommand
\ReMakeCommand

\MakeCommand{⟨*Command Name*⟩}{⟨*Code*⟩}
\ReMakeCommand{⟨*Command Name*⟩}{⟨*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.

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

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

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

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

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

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

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

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.

```
\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 2$_\varepsilon$ 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$_\varepsilon$ 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 2ε 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

3.1 Float Styles

Make equation references of the form (#).

```
\creflabelformat{equation}{#2#1#3}
```

Default table style

```
\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
}
```

Default figure style.

```
\captionsetup [figure] {
    format        = hang                ,
    labelsep      = colon               ,
    justification = justified           ,
    labelfont     = sl                  ,
    textfont      = sl                  ,
    font          = {normal,stretch=1.1},
    width         = 0.9\textwidth       ,
    position      = above               ,
    skip          = 0.5em
}
```

3.2 Links

Define a darker green than |green|.

```
\definecolor{UWMadGreen}{rgb}{0,0.7,0}
```

Define the default colors for the (internal) links, cites, and URLs.

```
\tl_new:N  \l_UWMad_LayoutStyle_Color_Link_tl
\tl_set:Nn \l_UWMad_LayoutStyle_Color_Link_tl {blue}
\tl_new:N  \l_UWMad_LayoutStyle_Color_Cite_tl
\tl_set:Nn \l_UWMad_LayoutStyle_Color_Cite_tl {UWMadGreen}
\tl_new:N  \l_UWMad_LayoutStyle_Color_URL_tl
\tl_set:Nn \l_UWMad_LayoutStyle_Color_URL_tl  {violet}
```

Define a new color and hyperlink defaults

```
\hypersetup {
    colorlinks         = true,
    linkcolor          = \l_UWMad_LayoutStyle_Color_Link_tl,
    citecolor          = \l_UWMad_LayoutStyle_Color_Cite_tl,
    urlcolor           = \l_UWMad_LayoutStyle_Color_URL_tl,
    pdfdisplaydoctitle = true,
    pdfview            = {FitH},
    pdfstartview       = {FitH},
    pdfpagelayout      = OneColumn,
    plainpages         = false,
    hypertexnames      = true,
    bookmarksopenlevel = 1,
    bookmarksopen      = true,
    unicode            = true
}
```

Define a helper commands to redefine all of the hyperref link colors using this class's color token lists.

```
\cs_new:Nn \UWMad_LayoutStyle_ResetLinkColors: {
    \hypersetup {
        colorlinks = true,
        linkcolor  = \l_UWMad_LayoutStyle_Color_Link_tl,
        citecolor  = \l_UWMad_LayoutStyle_Color_Cite_tl,
        urlcolor   = \l_UWMad_LayoutStyle_Color_URL_tl
    }
}
```

Define user interfaces to setting link colors.

```
\DeclareDocumentCommand \MakeLinksTheseColors { m m m } {
    \tl_set:Nn \l_UWMad_LayoutStyle_Color_Link_tl {#1}
    \tl_set:Nn \l_UWMad_LayoutStyle_Color_Cite_tl {#2}
    \tl_set:Nn \l_UWMad_LayoutStyle_Color_URL_tl  {#3}
    \UWMad_LayoutStyle_ResetLinkColors:
}
\DeclareDocumentCommand \MakeLinksThisColor { m } {
    \tl_set:Nn \l_UWMad_LayoutStyle_Color_Link_tl {#1}
    \tl_set:Nn \l_UWMad_LayoutStyle_Color_Cite_tl {#1}
    \tl_set:Nn \l_UWMad_LayoutStyle_Color_URL_tl  {#1}
    \UWMad_LayoutStyle_ResetLinkColors:
}
```

Define user interfaces to specialized color commands.

```
\keys_define:nn { UWMadThesis / LayoutStyle } {
    link-color .code:n = {
        \hypersetup {
            colorlinks = true,
            linkcolor  = #1,
        }
    },
    cite-color .code:n = {
        \hypersetup {
            colorlinks = true,
            citecolor  = #1,
        }
    },
    url-color .code:n = {
        \hypersetup {
            colorlinks = true,
            urlcolor  = #1,
        }
    },
    link-color .default:n = blue,
    cite-color .default:n = UWMadGreen,
    url-color  .default:n  = blue,
    all-link-color .code:n = {
        \hypersetup {
            colorlinks = true,
            linkcolor  = #1,
            citecolor  = #1,
            urlcolor   = #1,
        }
    },
    make-links-blue .meta:n = {
        all-link-color = blue
    },
    make-links-black .meta:n = {
        all-link-color = black
    },
    make-links-red .meta:n = {
        all-link-color = red
    }
}
\keys_set:nn { UWMadThesis / LayoutStyle } {
    link-color,
    cite-color,
    url-color
}
\DeclareDocumentCommand \MakeLinksBlack { } {
    \keys_set:nn { UWMadThesis / LayoutStyle } {
        make-links-black = true
    }
}
\DeclareDocumentCommand \MakeLinksBlue { } {
    \keys_set:nn { UWMadThesis / LayoutStyle } {
        make-links-blue = true
    }
}
\DeclareDocumentCommand \MakeLinksRed { } {
    \keys_set:nn { UWMadThesis / LayoutStyle } {
        make-links-red = true
    }
}
```

3.3 Page Layout

One inche magrins and letter (paper size) are set.

```
\geometry {
    includehead = true,
    margin      = 1.0in,
    paper       = letterpaper,
}
```

Invoke ‘doublespacing’ and set a warning in case any others invoke the ‘not cool’ commands according to the UW–Madison Guidelines.

```
\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.
    }
}
```

This setting puts the page numbers in the upper right-hand corner and atleast one inch from the top and right sides of the page (per the UW–Madison guidelines).

```
\pagestyle{myheadings}
\setlength{\headsep}  {1.15em}
```

Define user interface for defining different indentation styles.

```
\keys_define:nn { UWMadThesis / LayoutStyle } {
    indent-length .code:n = {
        \setlength{\parindent}{#1}
    },
    skip-length   .code:n = {
        \setlength{\parskip}{#1}
    },
    indent-length .default:n = 0pt,
    skip-length .default:n   = 1em,
    paragraph-style .choice:,
    paragraph-style / indent .code:n = {
        \setlength{\parindent}{1.50em}
        \setlength{\parskip}{0pt}
    },
    paragraph-style / pad    .code:n = {
        \setlength{\parindent}{0pt}
        \setlength{\parskip}{1em}
    }
}
\keys_set:nn { UWMadThesis / LayoutStyle } {
    paragraph-style = pad
}
```

3.4 Page Number

To avoid identifier problems with hyperref, the page counter is set to a large negative value here. It is reset to 1 after content begins.

```
\setcounter{page}{-100}
```

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} {
        \bool_if:NTF \g__UWmad_Appendix_Active_bool { } {
            \pagenumbering{arabic}
        }
    } { }
}
\UWMad_Hook_Prepend:Nn \@schapter {
    \thispagestyle{myheadings}
    \int_compare:nNnTF {\g__UWMad_FrontMatter_Counter_int} = {0} {
        \setcounter{page}{1}
        \pagenumbering{roman}
    } { }
}
\cs_new_eq:NN \StarredChapter \@schapter
\DeclareDocumentCommand \@schapter { m } {
    \StarredChapter{#1}
    \__UWMad_FrontMatter_Register:nn{chapter}{#1}
}
```

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 and activation flag

```
\int_new:N          \g__UWMad_Appendix_Counter_int
\int_set:Nn         \g__UWMad_Appendix_Counter_int {0}
\bool_new:N         \g__UWmad_Appendix_Active_bool
\bool_gset_false:N  \g__UWmad_Appendix_Active_bool
```

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_eq:NN  \__UWMad_Appendix_Initialize: \appendix
\cs_undefine:N \appendix
```

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

```
\DeclareDocumentCommand \appendix { s o m } {

    \int_compare:nNnTF {\g__UWMad_Appendix_Counter_int} = {0} {
        \bool_gset_true:N \g__UWmad_Appendix_Active_bool
        \__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} {
        \setcounter{page}{1}
        \pagestyle{myheadings}
        \pagenumbering{roman}
    } { }

    \int_gincr:N \g__UWMad_FrontMatter_Counter_int
    \bool_gset_true:N \g__UWMad_TOC_Registering_LOF_bool
    %\bool_if:nTF {
        %\g__UWMad_TOC_Registering_LOT_bool ||
        %\g__UWMad_TOC_Registering_LOF_bool } { } {
            \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}

}
\DeclareDocumentCommand \dedications { g } {
    \IfNoValueTF{#1} {
        \FrontMatterSetSection{}{Dedications}
    } {
       \FrontMatterSetSection{#1}{Dedications}
    }
}
\DeclareDocumentCommand \acknowledgments { g } {
    \IfNoValueTF{#1} {
        \FrontMatterSetSection{}{Acknowledgments}
    } {
       \FrontMatterSetSection{#1}{Acknowledgments}
    }
}
\DeclareDocumentCommand \abstract { g } {
    \IfNoValueTF{#1} {
        \FrontMatterSetSection{}{Abstract}
    } {
       \FrontMatterSetSection{#1}{Abstract}
    }
}
\DeclareDocumentCommand \umiabstract { g } {
    \IfNoValueTF{#1} {
        \FrontMatterSetSection{}{Abstract}
    } {
       \FrontMatterSetSection{#1}{Abstract}
    }
}
\DeclareDocumentCommand \preface { g } {
    \IfNoValueTF{#1} {
        \FrontMatterSetSection{}{Preface}
    } {
       \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_new_eq:NN \TableOfContentsDefault \tableofcontents
\cs_new_eq:NN \ListOfTablesDefault    \listoftables
\cs_new_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}
```

Create flags to indicate that the LOT or LOF are being registered. Since they are put into the TOC anyway, we need a guard to avoid double entires.

```
\bool_new:N \g__UWMad_TOC_Registering_LOT_bool
\bool_new:N \g__UWMad_TOC_Registering_LOF_bool
\bool_gset_false:N \g__UWMad_TOC_Registering_LOT_bool
\bool_gset_false:N \g__UWMad_TOC_Registering_LOF_bool
```

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:
        \cleardoublepage
        \phantomsection
        \ExplSyntaxOff
            \begin{spacing}{1.2}
                \TableOfContentsDefault
            \end{spacing}
        \ExplSyntaxOn
        \cleardoublepage
    \group_end:
}
\DeclareDocumentCommand \listoftables { } {

    \cs_set_eq:NN \listtablename \g__UWMad_TOC_Name_LOT_tl

    \group_begin:
        \ExplSyntaxOff
            \begin{spacing}{1.2}
                \ListOfTablesDefault
            \end{spacing}
        \ExplSyntaxOn
        \clearpage
    \group_end:
}
\DeclareDocumentCommand \listoffigures { } {

    \cs_set_eq:NN \listfigurename \g__UWMad_TOC_Name_LOF_tl

    \group_begin:
        \ExplSyntaxOff
            \begin{spacing}{1.2}
                \ListOfFiguresDefault
            \end{spacing}
        \ExplSyntaxOn
        \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 $\LaTeX\,2_\varepsilon$-defined section and also easily acquired/use the relationships among section levels when needed.

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

```
\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:nN {
    \UWMad_IfSectionExists:nT {#1} {
        \tl_set_eq:Nc #2 {c__UWMad_SectionsLevel_ #1 _tl}
    }
}
```

Given a level number, acquire its section.

```
\cs_new:Nn \UWMad_LevelToSection:nN {
    \UWMad_IfSectionExists:nT {#1} {
        \tl_set_eq:Nc #2 {c__UWMad_LevelsSection_ #1 _tl}
    }
}
```

Given a section, acquire its next lower one.

```
\cs_new:Nn \UWMad_NextSection:nN {
    \UWMad_IfSectionExists:nT {#1} {
        \tl_set_eq:Nc #2 {c__UWMad_NextSection_ #1 _tl}
    }
}
```

Given a section, acquire its next higher one.

```
\cs_new:Nn \UWMad_PreviousSection:nN {
    \UWMad_IfSectionExists:nT {#1} {
        \tl_set_eq:Nc #2 {c__UWMad_PreviousSection_ #1 _tl}
    }
}
```

Module 5

Math

We default the `\frac` command to a display style for all display environments.

```
\tex_everydisplay:D \exp_after:wN {
    \tex_the:D \tex_everydisplay:D
    \cs_set_eq:NN \frac \dfrac
}
```

5.1 Derivative Commands

Define the token list variables for the three supported derivative types.

```
\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
```

Define the user interface accessors.

```
\DeclareDocumentCommand \derivSymbol { } {
    \g_UWMad_Math_derivSymbol_tl
}
\DeclareDocumentCommand \pderivSymbol { } {
    \g_UWMad_Math_pderivSymbol_tl
}
\DeclareDocumentCommand \tderivSymbol { } {
    \g_UWMad_Math_tderivSymbol_tl
}
```

Define the user interface local mutators.

```
\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}
}
```

Define the user interface global mutators.

```
\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}
}
```

Define the `\left` and `\right` delimiter global mutators.

```
\DeclareDocumentCommand \DelimiterChangeDefault { m m } {
    \tl_gset:Nn  \g_UWMad_Math_DelimiterDefaultLeft_tl  {#1}
    \tl_gset:Nn  \g_UWMad_Math_DelimiterDefaultRight_tl {#2}
}
```

Define the generic regular and big derivative functions.

```
\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
}
```

Define the three supported derivative types' small forms.

```
\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}
}
```

Define the three supported derivative types' big forms.

```
\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}
}
```

5.2 Operators and Functions

Define all of the operators and function described in the user manual.

```
\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}
```

5.3 Miscallaneous Functions

Define the root function that has a tail.

```
\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}
}
```

User interface math mode check.

```
\DeclareExpandableDocumentCommand \IfMathModeTF { +m +m } {
    \mode_if_math:TF {
        #1
    }{
        $#2$
    }
}
```

Undefine the `\sups` commands defined by the IPA package.

```
\cs_gset_eq:NN \supsipa \sups
\cs_undefine:N \sups
```

Then define the `\subs`, `\sups`, and `\subsups` commands as described in the manual.

```
\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
\cs_gset_eq:NN \supsubs \subsups
```

The one-over functions discussed in the manual.

```
\DeclareDocumentCommand \OneOver { +m } {
    \frac{1}{#1}
}
\DeclareDocumentCommand \oneo { +m } {
    \OneOver{#1}
}
```

The non-math ‘d’ discussed in the manual.

```
\DeclareDocumentCommand \dd { m } {
    \mathrm{d}{#1}
}
```

The prime commands discussed in the manual.

```
\DeclareDocumentCommand \dprime { } {
    {\prime\prime}
}
\DeclareDocumentCommand \tprime { } {
    {\prime\prime\prime}
}
```

Two commands that were necessary for proper typesetting.

```
\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).

`ListOf` variable declarations for section levels.

```
\tl_new:N   \l__UWMad_ListOf_Section_Main_tl
\tl_new:N   \l__UWMad_ListOf_Section_Group_tl
\tl_new:N   \l__UWMad_ListOf_Section_Subgroup_tl
```

Boolean declarations for numbering and Table of Contents-inclusions.

```
\bool_new:N \l__UWMad_ListOf_MakeNumbered_Main_bool
\bool_new:N \l__UWMad_ListOf_MakeNumbered_Group_bool
\bool_new:N \l__UWMad_ListOf_MakeNumbered_Subgroup_bool
\bool_new:N \l__UWMad_ListOf_IncludeInTOC_Main_bool
\bool_new:N \l__UWMad_ListOf_IncludeInTOC_Group_bool
\bool_new:N \l__UWMad_ListOf_IncludeInTOC_Subgroup_bool
```

Entry queue and and Hook key-value initialization

```
\seq_new:N \l__UWMad_ListOf_EntryQueue_seq
\prop_new:N \l__UWMad_ListOf_Hooks_prop
\cs_new:Nn \__UWMad_Listof_SetHooks_Blank: {
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {PreTitle-Main}       {}
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {PostTitle-Main}      {}
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {PreTitle-Group}      {}
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {PostTitle-Group}     {}
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {PreTitle-Subgroup}   {}
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {PostTitle-Subgroup}  {}
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {PrePush}             {}
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {PostPush}            {}
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {PrePrint}            {}
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {PostPrint}           {}
}
```

Function initializations for sectioning and title print commands.

```
\cs_new:Nn \__UWMad_ListOf_SectioningCommand_Main:     {}
\cs_new:Nn \__UWMad_ListOf_SectioningCommand_Group:    {}
\cs_new:Nn \__UWMad_ListOf_SectioningCommand_Subgroup: {}
\cs_new:Nn \UWMad_ListOf_PrintTitle_Main:nn     {}
\cs_new:Nn \UWMad_ListOf_PrintTitle_Group:nn    {}
\cs_new:Nn \UWMad_ListOf_PrintTitle_Subgroup:nn {}
```

UWMad_ListOf_SetHook:nn

`\UWMad_ListOf_SetHook:nn{`⟨*Hook name*⟩`}{`⟨*Hook code*⟩`}`

Sets `{`⟨*Hook name*⟩`}` to `{`⟨*Hook code*⟩`}` for the `ListOf`. There are hooks when pushing to the queue: `PrePush` and `PostPush`. There are hooks when printing entires: `PrePrint` and `PostPrint`. There are also hooks for all section titles: `PreTitle-*` and `PostTitle-*`.

```
\cs_new:Nn \UWMad_ListOf_SetHook:nn {
    \prop_put:Nnn \l__UWMad_ListOf_Hooks_prop {#1} {#2}
}
```

These function initialize the sectioning commands for the associated `ListOf` level.

```
\cs_new:Nn \__UWMad_ListOf_Initialize_SectioningCommands: {
    \UWMad_IfSectionExists:nT {\l__UWMad_ListOf_Section_Main_tl} {
        \cs_set_eq:cc
            {__UWMad_ListOf_SectioningCommand_Main:w}
            {\l__UWMad_ListOf_Section_Main_tl}

        \UWMad_NextSection:nN{\l__UWMad_ListOf_Section_Main_tl} \l_tmpa_tl
        \cs_set_eq:cc
```

```
                {__UWMad_ListOf_SectioningCommand_Group:w}
                {\l_tmpa_tl}

            \UWMad_NextSection:nN{\l_tmpa_tl} \l_tmpb_tl
            \cs_set_eq:cc
                {__UWMad_ListOf_SectioningCommand_Subgroup:w}
                {\l_tmpb_tl}
        }
}
```

This function initializes the list of using the helper functions above.

```
\cs_new:Nn \__UWMad_ListOf_Initialize_TitlePrinter:n {
    \cs_set:cn {UWMad_ListOf_PrintTitle_ #1 :nn} {

        \prop_item:Nn \l__UWMad_ListOf_Hooks_prop {PreTitle-#1}

        \bool_if:cTF {l__UWMad_ListOf_MakeNumbered_ #1 _bool} {
            \bool_if:cTF {l__UWMad_ListOf_IncludeInTOC_ #1 _bool} {
                \use:c{__UWMad_ListOf_SectioningCommand_ #1 :w}[##1]{##2}
            } {
                \int_gset_eq:NN \l_tmpa_int \c@tocdepth
                \int_gset:Nn \c@tocdepth {-1}
                \use:c{__UWMad_ListOf_SectioningCommand_ #1 :w}{##2}
                \int_gset:Nn \c@tocdepth {\l_tmpa_int}
            }
        } {
            \use:c{__UWMad_ListOf_SectioningCommand_ #1 :w} * {##2}
            \bool_if:cTF {l__UWMad_ListOf_IncludeInTOC_ #1 _bool} {
                \tl_if_in:cnTF
                    {l__UWMad_ListOf_Section_ #1 _tl} {chapter} { } {
                    \addcontentsline
                        {toc}
                        {\tl_use:c{l__UWMad_ListOf_Section_ #1 _tl}}
                        {##1}
                }
            } { }
        }
        \prop_item:Nn \l__UWMad_ListOf_Hooks_prop {PostTitle-#1}
    }
}
\cs_new:Nn \__UWMad_ListOf_Initialize_TitlePrinters: {
    \__UWMad_ListOf_Initialize_TitlePrinter:n {Main}
    \__UWMad_ListOf_Initialize_TitlePrinter:n {Group}
    \__UWMad_ListOf_Initialize_TitlePrinter:n {Subgroup}
}
```

This function initializes the list of using the helper functions above.

```
\cs_new:Nn \__UWMad_ListOf_Clear: {
    \seq_clear:N \l__UWMad_ListOf_EntryQueue_seq
    \__UWMad_Listof_SetHooks_Blank:
}
\cs_new:Nn \UWMad_ListOf_Initialize: {
    \__UWMad_ListOf_Clear:
    \__UWMad_ListOf_Initialize_SectioningCommands:
    \__UWMad_ListOf_Initialize_TitlePrinters:
}
```

`\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:n {
    \prop_item:Nn     \l__UWMad_ListOf_Hooks_prop {PrePush}
    \seq_put_right:Nn \l__UWMad_ListOf_EntryQueue_seq {#1}
    \prop_item:Nn     \l__UWMad_ListOf_Hooks_prop {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: {
    \prop_item:Nn      \l__UWMad_ListOf_Hooks_prop {PrePrint}
    \seq_map_inline:Nn \l__UWMad_ListOf_EntryQueue_seq {##1}
    \seq_clear:N       \l__UWMad_ListOf_EntryQueue_seq
    \prop_item:Nn      \l__UWMad_ListOf_Hooks_prop {PostPrint}
}
```

6.1 Nomenclature

Dimensions that are calculated and private.

```
\dim_new:N \l__UWMad_Nomenclature_WidestSymbol_dim
\dim_new:N \l__UWMad_Nomenclature_WidestUnit_dim
\dim_new:N \l__UWMad_Nomenclature_Entry_Symbol_Width_dim
\dim_new:N \l__UWMad_Nomenclature_Entry_Units_Width_dim
\dim_new:N \l__UWMad_Nomenclature_Entry_Description_Width_dim
```

User-adjustable dimensions that are public.

```
\dim_new:N \l_UWMad_Nomenclature_Skip_EntryPrint_dim
\dim_new:N \l_UWMad_Nomenclature_Entry_Margin_Left_dim
\dim_new:N \l_UWMad_Nomenclature_Entry_Margin_Bottom_dim
\dim_new:N \l_UWMad_Nomenclature_Entry_Margin_Right_dim
\dim_new:N \l_UWMad_Nomenclature_Entry_Margin_Top_dim
\dim_new:N \l_UWMad_Nomenclature_Entry_Pad_Column_dim
```

Coffins used in typesetting an entry's contents.

```
\coffin_new:N \l__UWMad_Nomenclature_Entry_coffin
\coffin_new:N \l__UWMad_Nomenclature_Symbol_coffin
\coffin_new:N \l__UWMad_Nomenclature_Description_coffin
\coffin_new:N \l__UWMad_Nomenclature_Units_coffin
```

Options for the units column.

```
\bool_new:N \l_UWMad_Nomenclature_Units_IncludeColumn_bool
\bool_new:N \l_UWMad_Nomenclature_Units_UseSIUnitx_bool
\bool_new:N \l_UWMad_Nomenclature_Units_UseDelimiter_bool
\tl_new:N   \l_UWMad_Nomenclature_Units_Delimiter_Left_tl
\tl_new:N   \l_UWMad_Nomenclature_Units_Delimiter_Right_tl
```

Miscellaneous token lists.

```
\tl_new:N   \l__UWMad_Nomenclature_Entry_LineStretch_tl
\tl_new:N   \l__UWMad_Nomenclature_Title_Main_tl

\cs_new:Nn \UWMad_Nomenclature_SetUnitsBox:n {
    \bool_if:nTF {
        \l_UWMad_Nomenclature_Units_UseDelimiter_bool &&
        \l_UWMad_Nomenclature_Units_UseSIUnitx_bool
    } {
        $
        \left\l_UWMad_Nomenclature_Units_Delimiter_Left_tl
            \si{#1}
        \right\l_UWMad_Nomenclature_Units_Delimiter_Right_tl
```

```
            $
    } {
        \bool_if:nTF {
            \l_UWMad_Nomenclature_Units_UseDelimiter_bool &&
            !\l_UWMad_Nomenclature_Units_UseSIUnitx_bool
        } {
            $
            \left\l_UWMad_Nomenclature_Units_Delimiter_Left_tl
                #1
            \right\l_UWMad_Nomenclature_Units_Delimiter_Right_tl
            $
        }{
            \si{#1}
        }
    }
}
```

`\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_eq: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
        {
            \UWMad_Nomenclature_SetUnitsBox:n{#1}
        }
}
```

And the defaults for all keys are now set.

`\UWMad_Nomenclature_ZeroWidest_Symbol:`	`\UWMad_Nomenclature_ZeroWidest_Symbol:`
`\UWMad_Nomenclature_ZeroWidest_Unit:`	`\UWMad_Nomenclature_ZeroWidest_Symbol:`

These commands set the widest symbol and unit lengths to 0pt.

```
\cs_new:Nn \UWMad_Nomenclature_ZeroWidest_Symbol: {
    \dim_set:Nn \l__UWMad_Nomenclature_WidestSymbol_dim {0pt}
}
\cs_new:Nn \UWMad_Nomenclature_ZeroWidest_Units: {
    \dim_set:Nn \l__UWMad_Nomenclature_WidestUnit_dim {0pt}
}
```

And the defaults for all keys are now set.

`\UWMad_Nomenclature_SetEntryWidths_NoUnits:` `\UWMad_Nomenclature_SetEntryWidths_NoUnits:`
`\UWMad_Nomenclature_SetEntryWidths_Units:` `\UWMad_Nomenclature_SetEntryWidths_Units:`

These commands sets the widths of the description, symbol, and (if present) unit boxes for a particular entry.

```
\cs_new:Nn \UWMad_Nomenclature_SetEntryWidths_NoUnits: {
    \dim_set:Nn \l__UWMad_Nomenclature_Entry_Symbol_Width_dim {
        1.01\l__UWMad_Nomenclature_WidestSymbol_dim
    }
    \dim_set:Nn \l__UWMad_Nomenclature_Entry_Description_Width_dim {
        \columnwidth -
        \l_UWMad_Nomenclature_Entry_Margin_Left_dim   -
        \l__UWMad_Nomenclature_Entry_Symbol_Width_dim -
        \l_UWMad_Nomenclature_Entry_Pad_Column_dim    -
        \l_UWMad_Nomenclature_Entry_Margin_Right_dim
    }
}
\cs_new:Nn \UWMad_Nomenclature_SetEntryWidths_Units: {
    \dim_set:Nn \l__UWMad_Nomenclature_Entry_Symbol_Width_dim {
        1.05\l__UWMad_Nomenclature_WidestSymbol_dim
    }
    \dim_set:Nn \l__UWMad_Nomenclature_Entry_Units_Width_dim {
        1.05\l__UWMad_Nomenclature_WidestUnit_dim
    }
    \dim_set:Nn \l__UWMad_Nomenclature_Entry_Description_Width_dim {
        0.995\columnwidth -
         \l_UWMad_Nomenclature_Entry_Margin_Left_dim   -
         \l__UWMad_Nomenclature_Entry_Symbol_Width_dim -
         \l__UWMad_Nomenclature_Entry_Units_Width_dim  -
        2\l_UWMad_Nomenclature_Entry_Pad_Column_dim    -
         \l_UWMad_Nomenclature_Entry_Margin_Right_dim
    }
}
```

And the defaults for all keys are now set.

\UWMad_Nomenclature_SetEntryWidths: \UWMad_Nomenclature_SetEntryWidths:

This function calls one of the appropriate above setters.

```
\cs_new:Nn \UWMad_Nomenclature_SetEntryWidths: {
    \bool_if:NTF \l_UWMad_Nomenclature_Units_IncludeColumn_bool {
        \UWMad_Nomenclature_SetEntryWidths_Units:
    } {
        \UWMad_Nomenclature_SetEntryWidths_NoUnits:
    }
}
```

And the defaults for all keys are now set.

\UWMad_Nomenclature_SetEntry_NoUnits:nn
\UWMad_Nomenclature_SetEntry_Units:nnn

\UWMad_Nomenclature_SetEntry_NoUnits: {⟨*symbol*⟩}{⟨*description*⟩}
\UWMad_Nomenclature_SetEntry_Units: {⟨*symbol*⟩}{⟨*units*⟩}{⟨*description*⟩}

These functions typeset the contents passed into them.

```
\cs_new:Nn \UWMad_Nomenclature_SetEntry_NoUnits:nn {
    \coffin_clear:N \l__UWMad_Nomenclature_Entry_coffin
    \coffin_clear:N \l__UWMad_Nomenclature_Symbol_coffin
    \coffin_clear:N \l__UWMad_Nomenclature_Description_coffin
    \vcoffin_set:Nnn
        \l__UWMad_Nomenclature_Entry_coffin
        {\l__UWMad_Nomenclature_Entry_Symbol_Width_dim} {#1}
    \vcoffin_set:Nnn
        \l__UWMad_Nomenclature_Description_coffin
        {\l__UWMad_Nomenclature_Entry_Description_Width_dim} {#2}
    \coffin_join:NnnNnnnn
        \l__UWMad_Nomenclature_Entry_coffin       {l}{T}
        \l__UWMad_Nomenclature_Symbol_coffin      {l}{T}
        {\l_UWMad_Nomenclature_Entry_Margin_Left_dim}{0pt}
    \coffin_join:NnnNnnnn
        \l__UWMad_Nomenclature_Entry_coffin        {l}{T}
        \l__UWMad_Nomenclature_Description_coffin  {l}{T}
        {
            \l_UWMad_Nomenclature_Entry_Margin_Left_dim   +
            \l__UWMad_Nomenclature_Entry_Symbol_Width_dim +
            \l_UWMad_Nomenclature_Entry_Pad_Column_dim
        } {0pt}
    \setstretch{\l__UWMad_Nomenclature_Entry_LineStretch_tl}
    \skip_vertical:n{\l_UWMad_Nomenclature_Entry_Margin_Top_dim}
    \coffin_typeset:Nnnnn
        \l__UWMad_Nomenclature_Entry_coffin {l}{t}{0pt}{0pt}
    \tex_hfill:D
    \skip_vertical:n{\l_UWMad_Nomenclature_Entry_Margin_Bottom_dim}
}
\cs_new:Nn \UWMad_Nomenclature_SetEntry_Units:nnn {
    \coffin_clear:N \l__UWMad_Nomenclature_Entry_coffin
%
%   Set the information into their coffins
    \vcoffin_set:Nnn
        \l__UWMad_Nomenclature_Symbol_coffin
        {\l__UWMad_Nomenclature_Entry_Symbol_Width_dim} {#1}
    \vcoffin_set:Nnn
        \l__UWMad_Nomenclature_Description_coffin
        {\l__UWMad_Nomenclature_Entry_Description_Width_dim} {#3}
%
%   Units setting: center using hfil and then handle bracing and siunitx
%   embeding options.
    \hcoffin_set:Nn
        \l__UWMad_Nomenclature_Units_coffin
```

```
\DeclareDocumentEnvironment {Nomenclature} { o o } {
%
%   Initialization
    \UWMad_ListOf_Initialize:
    \setlength{\parskip}{0pt}
    \setlength{\parindent}{0pt}
%
%
%   Check for an optional section declaration and
%   set Main section token list.
    \IfValueTF {#1} {
        \tl_set:Nx \l__UWMad_ListOf_Title_Main_tl { #1 }
    } { }
%
%
%
%   Set some hooks in the Nomenclature ListOf instance
    \UWMad_ListOf_SetHook:nn {PrePrint} {
        \UWMad_Nomenclature_SetEntryWidths:
    }
    \UWMad_ListOf_SetHook:nn {PostPrint} {
        \UWMad_Nomenclature_ZeroWidest_Symbol:
        \UWMad_Nomenclature_ZeroWidest_Units:
    }
%
%
%   User front-end for creating a Group
    \DeclareDocumentCommand \Group { o m } {
        \UWMad_ListOf_PrintEntries:
        \IfNoValueTF {##1} {
            \UWMad_ListOf_PrintTitle_Group:nn{##2}{##2}
        } {
            \UWMad_ListOf_PrintTitle_Group:nn{##1}{##2}
        }
    }
%
%   User front-end for creating a Subgroup
    \DeclareDocumentCommand \Subgroup { o m } {
        \UWMad_ListOf_PrintEntries:
        \IfNoValueTF {##1} {
            \UWMad_ListOf_PrintTitle_Subgroup:nn{##2}{##2}
        } {
            \UWMad_ListOf_PrintTitle_Subgroup:nn{##1}{##2}
        }
    }
%
%   User front-end for creating an entry
    \cs_undefine:N \Entry
    \bool_if:NTF \l_UWMad_Nomenclature_Units_IncludeColumn_bool {
        \DeclareDocumentCommand \Entry { m m m } {
            \UWMad_ListOf_PushEntry:n {
                \UWMad_Nomenclature_SetEntry_Units:nnn
                    {##1} {##2} {##3}
            }
            \UWMad_Nomenclature_UpdateWidest_Symbol:n{##1}
            \UWMad_Nomenclature_UpdateWidest_Units:n{##2}
        }
    } {
        \DeclareDocumentCommand \Entry { m m } {
            \UWMad_ListOf_PushEntry:n {
                \UWMad_Nomenclature_SetEntry_NoUnits:nn
                    {##1} {##2}
            }
            \UWMad_Nomenclature_UpdateWidest_Symbol:n{##1}
        }
    }
%
%   User front-end for reseting the column width
    \DeclareDocumentCommand  \PrintEntries { } {
        \UWMad_ListOf_PrintEntries:
    }
%
%
    \IfNoValueTF {#2} {
        \IfNoValueTF {#1} {
            \UWMad_ListOf_PrintTitle_Main:nn
                {\l__UWMad_Nomenclature_Title_Main_tl}
                {\l__UWMad_Nomenclature_Title_Main_tl}
        } {
            \UWMad_ListOf_PrintTitle_Main:nn{#1}{#1}
        }
    } {
        \UWMad_ListOf_PrintTitle_Main:nn{#1}{#2}
    }
%
} {
%   Flush the remaining entries from the ListOf queue.
    \UWMad_ListOf_PrintEntries:
}
%
%
%
```

```
%
%
%
%
%
\clist_new:N   \g__UWMad_Nomenclature_KeyValuePairs_clist
\clist_gset:Nn \g__UWMad_Nomenclature_KeyValuePairs_clist {
    main-title .tl_set:N  = \l__UWMad_Nomenclature_Title_Main_tl,
    main-title .default:n = Nomenclature,
    main-section     .tl_set:N = \l__UWMad_ListOf_Section_Main_tl,
    group-section    .tl_set:N = \l__UWMad_ListOf_Section_Group_tl,
    subgroup-section .tl_set:N = \l__UWMad_ListOf_Section_Subgroup_tl,
    main-section     .default:n = chapter,
    group-section    .default:n = section,
    subgroup-section .default:n = subsection,
    make-main-numbered     .bool_set:N =
        \l__UWMad_ListOf_MakeNumbered_Main_bool,
    make-group-numbered    .bool_set:N =
        \l__UWMad_ListOf_MakeNumbered_Group_bool,
    make-subgroup-numbered .bool_set:N =
        \l__UWMad_ListOf_MakeNumbered_Subgroup_bool,
    make-numbered .meta:n = {
        make-main-numbered     = #1,
        make-group-numbered    = #1,
        make-subgroup-numbered = #1
    },
    make-numbered .default:n = false,
    include-main-in-toc     .bool_set:N =
        \l__UWMad_ListOf_IncludeInTOC_Main_bool,
    include-group-in-toc    .bool_set:N =
        \l__UWMad_ListOf_IncludeInTOC_Group_bool,
    include-subgroup-in-toc .bool_set:N =
        \l__UWMad_ListOf_IncludeInTOC_Subgroup_bool,
    include-in-toc .meta:n = {
        include-main-in-toc     = #1,
        include-group-in-toc    = #1,
        include-subgroup-in-toc = #1
    },
    include-in-toc .default:n = true,
    print-skip          .dim_set:N =
        \l_UWMad_Nomenclature_Skip_EntryPrint_dim,
    entry-margin-top    .dim_set:N =
        \l_UWMad_Nomenclature_Entry_Margin_Top_dim,
    entry-margin-left   .dim_set:N =
        \l_UWMad_Nomenclature_Entry_Margin_Left_dim,
    entry-margin-right  .dim_set:N =
        \l_UWMad_Nomenclature_Entry_Margin_Right_dim,
    entry-margin-bottom  .dim_set:N =
        \l_UWMad_Nomenclature_Entry_Margin_Bottom_dim,
    entry-column-padding .dim_set:N =
        \l_UWMad_Nomenclature_Entry_Pad_Column_dim,
    print-skip           .default:n = 1em,
    entry-margin-top     .default:n = 0pt,
    entry-margin-left    .default:n = 1.1em,
    entry-margin-right   .default:n = 0pt,
    entry-margin-bottom  .default:n = 0.80em,
    entry-column-padding .default:n = 0.80em,
    entry-stretch .tl_set:N =
        \l__UWMad_Nomenclature_Entry_LineStretch_tl,
    entry-stretch .default:n = 1.1,
    include-units-column .bool_set:N =
        \l_UWMad_Nomenclature_Units_IncludeColumn_bool,
    include-units-column .default:n = false,
    units-embed-siunitx .bool_set:N =
        \l_UWMad_Nomenclature_Units_UseSIUnitx_bool,
    units-embed-siunitx .default:n = false,
    units-left-delimiter .code:n = {
        \tl_set:Nn \l_UWMad_Nomenclature_Units_Delimiter_Left_tl {#1}
        \bool_set_true:N \l_UWMad_Nomenclature_Units_UseDelimiter_bool
    },
    units-right-delimiter .code:n = {
        \tl_set:Nn \l_UWMad_Nomenclature_Units_Delimiter_Right_tl {#1}
        \bool_set_true:N \l_UWMad_Nomenclature_Units_UseDelimiter_bool
    }
}
%
%
\exp_args:Nnf
    \keys_define:nn
    { UWMadThesis / Nomenclature }
    {
        \clist_use:Nn \g__UWMad_Nomenclature_KeyValuePairs_clist {,}
    }
%
%
%
\keys_set:nn { UWMadThesis / Nomenclature } {
    main-title = Nomenclature ,
    main-section = chapter,
    make-numbered = true ,
    include-in-toc = true ,
    include-units-column = false,
    units-embed-siunitx = false ,
    print-skip           ,
    entry-margin-top     ,
    entry-margin-left    ,
    entry-margin-right   ,
    entry-margin-bottom  ,
    entry-column-padding ,
    entry-stretch
}
%
%
%
%
%
%
%
%
%
\tl_new:N \l__UWMad_Acronym_Title_Main_tl
%
\DeclareDocumentEnvironment {Acronym} { o o } {

    \IfNoValueTF {#2} {
        \IfNoValueTF {#1} {
            \begin{Nomenclature}
                [\l__UWMad_Acronym_Title_Main_tl]
                [\l__UWMad_Acronym_Title_Main_tl]
        } {
            \begin{Nomenclature}[#1][#1]
        }
    } {
        \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:n {
                    \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}

}
%
%
%
\cs_new:Nn \UWMad_Acronym_CreateLink:n {
    \hyperlink{Acronym:#1}{
        \color{\g__UWMad_Acronym_LinkColor_tl}
        \UWMad_Hash_Get:nn{Acronyms}{#1}
    }
}
%
%
\DeclareDocumentCommand \Acro { m } {
    \UWMad_Hash_IfKeySet:nnTF {Acronyms} {#1} {
        \bool_if:cTF {g__UWMad_Acronym_WasSet_#1_bool} {
            \bool_if:NTF \g__UWMad_Acronym_UseLinks_bool {
                \UWMad_Acronym_CreateLink:n{#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}
        }
    } { }
}
%
%
```

Define the keys for the Acronym system by expanding the `clist` created for the Nomenclature system.

```
\exp_args:Nnf
    \keys_define:nn
    { UWMadThesis / Acronym }
    {
        \clist_use:Nn \g__UWMad_Nomenclature_KeyValuePairs_clist {,}
    }
\keys_define:nn { UWMadThesis / Acronym } {
    main-title .tl_set:N  = \l__UWMad_Acronym_Title_Main_tl,
    main-title .default:n = Acronyms,
    use-links .bool_gset:N = \g__UWMad_Acronym_UseLinks_bool,
    use-links .default:n = true,
    link-color .tl_gset:N = \g__UWMad_Acronym_LinkColor_tl,
    link-color .default:n = blue
}
```

And the defaults for all keys are now set.

```
\keys_set:nn { UWMadThesis / Acronym } {
    main-title           ,
    main-section         ,
    group-section        ,
    subgroup-section     ,
    make-numbered        ,
    include-in-toc       ,
    include-units-column ,
    print-skip           ,
    entry-margin-top     ,
    entry-margin-left    ,
    entry-margin-right   ,
    entry-margin-bottom  ,
    entry-column-padding ,
    entry-stretch        ,
    use-links            ,
    link-color
}
```

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_sys_jobname_str.UWMad.PDFMetaData.aux} {
    \file_input:n {\c_sys_jobname_str.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_sys_jobname_str.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
}
```

Semantic names for the `\Degree` function.

```
\DeclareDocumentCommand \Doctorate { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_Degree_tl {Doctor~of~Philosophy}
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
}
\DeclareDocumentCommand \Masters { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_Degree_tl {Master's}
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
}
\DeclareDocumentCommand \Bachelors { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_Degree_tl {Bachelor's}
    \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}
}
```

Semantic names for the `\DocumentType` function.

```
\DeclareDocumentCommand \Dissertation { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
        dissertation
    }
}
\DeclareDocumentCommand \DoctoralThesis { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
        doctoral~thesis
    }
}
\DeclareDocumentCommand \MastersThesis { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
        master's~thesis
    }
}
\DeclareDocumentCommand \Thesis { } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
        thesis
    }
}
\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 alias.

```
\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
```

Define the optional user interface.

```
\DeclareDocumentCommand \Department { m } {
    \tl_gset:Nn \g__UWMad_ThesisInfo_Department_tl {#1}
}
```

Define an author interface for determingin if required information has been set.

```
\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}
    }
}
```

Define user accessors for thesis info.

```
\DeclareExpandableDocumentCommand \TheTitle { } {
    \g__UWMad_ThesisInfo_Title_tl
}
\DeclareExpandableDocumentCommand \TheAuthor { } {
    \g__UWMad_ThesisInfo_Author_tl
}
\DeclareExpandableDocumentCommand \TheProgram { } {
    \g__UWMad_ThesisInfo_Program_tl
}
\DeclareExpandableDocumentCommand \TheDegree { } {
    \g__UWMad_ThesisInfo_Degree_tl
}
\DeclareExpandableDocumentCommand \TheDocumentType { } {
    \g__UWMad_ThesisInfo_DocumentType_tl
}
\DeclareExpandableDocumentCommand \TheDefenseDate { } {
    \g__UWMad_ThesisInfo_DefenseDate_tl
}
\cs_gset_eq:NN \TheDefenceDate \TheDefenseDate
\DeclareExpandableDocumentCommand \TheInstitution { } {
    \g__UWMad_ThesisInfo_Institution_tl
}
\cs_set_eq:NN \TheUniversity \TheInstitution
%
\DeclareExpandableDocumentCommand \TheDepartment { } {
    \g__UWMad_ThesisInfo_Department_tl
}
\DeclareExpandableDocumentCommand \TheAdvisor { } {
    \g__UWMad_ThesisInfo_AdvisorName_tl
}
```

7.3 Committee Member List

Define internals for the Committee member list: a separator, a count, a coffin, and a sequence.

```
\tl_new:N   \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl
\tl_gset:Nn \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl {,}
\int_new:N \g__UWMad_ThesisInfo_CommitteeCount_int
\coffin_new:N  \g__UWMad_ThesisInfo_Committee_coffin
\vcoffin_set:Nnn \g__UWMad_ThesisInfo_Committee_coffin {\textwidth}{}
\seq_new:N \g__UWMad_ThesisInfo_Committee_CoffinExpanders_seq
```

```
\cs_new:Nn \__UWMad_ThesisInfo_Committee_AddMember:nnn {
    \seq_gput_right:Nn \g__UWMad_ThesisInfo_Committee_CoffinExpanders_seq {
        \vcoffin_set:Nnn \l_tmpa_coffin {\textwidth-1.01em} {
            #1
            \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl{}
            \
            \textsl{#2}
            \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl{}
            \
            \textsl{#3}
        }
        \coffin_join:NnnNnnnn
            \g__UWMad_ThesisInfo_Committee_coffin {l} {b}
            \l_tmpa_coffin                        {l} {t}
            {0pt}{-0.75em}
    }
}
\cs_new:Nn \__UWMad_ThesisInfo_Committee_AddAdvisor:nnn {
    \seq_gput_left:Nn \g__UWMad_ThesisInfo_Committee_CoffinExpanders_seq {
        \vcoffin_set:Nnn \l_tmpa_coffin {\textwidth-1.01em} {
            #1
            \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl{}
            \
            \textsl{#2}
            \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl{}
            \
            \textsl{#3}
            \bool_if:NTF \g__UWMad_ThesisInfo_PrintAdvisorMarker_bool {
                \
                (\g__UWMad_ThesisInfo_AdvisorMarker_tl{})
            } { }
        }
        \coffin_join:NnnNnnnn
            \g__UWMad_ThesisInfo_Committee_coffin {l} {b}
            \l_tmpa_coffin                        {l} {t}
            {0pt}{-0.75em}
    }
}
```

Define the Advisor and Adviser user interface.

```
\bool_new:N        \g__UWMad_ThesisInfo_PrintAdvisorMarker_bool
\bool_gset_false:N \g__UWMad_ThesisInfo_PrintAdvisorMarker_bool
\cs_new:Nn \UWMad_ThesisInfo_AdvisorInfo:nnn {
    \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}
    \__UWMad_ThesisInfo_Committee_AddAdvisor:nnn{#1}{#2}{#3}
}
\DeclareDocumentCommand \Advisor { m m m } {
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool
    \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorMarker_tl {Advisor}
    \UWMad_ThesisInfo_AdvisorInfo:nnn{#1}{#2}{#3}
 }
\DeclareDocumentCommand \Adviser { m m m } {
    \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool
    \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorMarker_tl {Adviser}
    \UWMad_ThesisInfo_AdvisorInfo:nnn{#1}{#2}{#3}
}
```

Define user interface for adding a person to the committee list.

```
\DeclareDocumentCommand \CommitteeMember { m m m } {
    \int_gincr:N \g__UWMad_ThesisInfo_CommitteeCount_int
    \__UWMad_ThesisInfo_Committee_AddMember:nnn{#1}{#2}{#3}
}
```

Define an author interface for printing the Committee member list.

```
\DeclareDocumentCommand \PrintCommitteeMemberList { } {
    {
        \seq_map_inline:Nn \g__UWMad_ThesisInfo_Committee_CoffinExpanders_seq {
            ##1
        }
        \coffin_typeset:Nnnnn \g__UWMad_ThesisInfo_Committee_coffin
        {l}{t}{1em}{0pt}
    }
}
```

7.4 PDF Metadata

Define metadata internals.

```
\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
```

Define user interface for setting metadata.

```
\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{pdfkeywords} {#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{pdfproducer}  {#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{pdfcreator} {#1}
    \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
        \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
    } { }
}
```

Define user interface for accessing metadata.

```
\DeclareExpandableDocumentCommand \TheSubject { } {
    \g__UWMad_PDFMetaData_Subject_tl
}
\DeclareExpandableDocumentCommand \TheKeywords { } {
    \g__UWMad_PDFMetaData_Keywords_tl
}
\DeclareExpandableDocumentCommand \TheProducer { } {
    \g__UWMad_PDFMetaData_Producer_tl
}
```

```
\DeclareExpandableDocumentCommand \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} {
        at~the                      \\[0.50em]
        \textsc{\TheInstitution{}}  \\[0.50em]
         \the\year
    }
}

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

\DeclareDocumentCommand \TitlePageCommitteLeadIn { } {
    The~dissertation~is~approved~by~the~following~members~of~the~Final~Oral~Com
}


\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{}\\
    \TitlePageCommitteLeadIn{} \\[-1.5em]
    \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
}
\cs_new:Nn \__UWMad_LicensePage_FinishPage: {
    \clearpage
}
%
```

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}
}
%
%
} {

    \bool_if:nTF {
        \l__UWMad_CCLicense_UseCreativeCommons_bool &&
        \l__UWMad_Copyright_UseCopyright_bool
    } {
        \msg_new:nnn { UWMadThesis } { SpecialPages / MultipleLicenses } {
            Both~Creative~Commons~and~Copyright~have~been~declared.~
            Please,~pick~one.
        }
        \msg_error:nn { UWMadThesis } { SpecialPages / MultipleLicenses }
    } { }



    \bool_if:NTF \l__UWMad_CCLicense_UseCreativeCommons_bool {

        \__UWMad_CCLicense_CreateType:
        \__UWMad_CCLicense_CheckTypeValidity:
        \bool_if:NTF \l__UWMad_CCLicense_IsValid_bool {
            \cs_new_eq:NN
                \__UWMad_LicensePage_LicenseText:
                \__UWMad_CCLicense_LicenseText:
        } { }

    } { }



    \bool_if:NTF \l__UWMad_Copyright_UseCopyright_bool {
        \cs_new_eq:NN
            \__UWMad_LicensePage_LicenseText:
            \__UWMad_Copyright_LicenseText:
    } { }



    \cs_if_exist:NTF \__UWMad_LicensePage_LicenseText: {
        \__UWMad_LicensePage_StartPage:
        \vbox_to_ht:nn {0.3333\textheight} {
            \__UWMad_LicensePage_LicenseText:
        }
        \__UWMad_LicensePage_FinishPage:
    } { }


}
%
```

Module 9

Relative Directory Input

9.1 Declarations and Initializations

Variable declarations and default initializations for Chapter directories.

```
\int_new:N  \g__UWMad_RelativeDirectory_Chapter_Count_int
\tl_new:N   \g__UWMad_RelativeDirectory_Chapter_Prefix_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Chapter_Suffix_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Chapter_CurrentPath_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Chapter_CurrentName_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Chapter_ParentPath_tl
\tl_gset:Nn \g__UWMad_RelativeDirectory_Chapter_ParentPath_tl {}
```

Variable declarations and default initializations for Section directories.

```
\int_new:N  \g__UWMad_RelativeDirectory_Section_Count_int
\tl_new:N   \g__UWMad_RelativeDirectory_Section_Prefix_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Section_Suffix_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Section_CurrentPath_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Section_CurrentName_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Section_ParentPath_tl
\tl_gset:Nn \g__UWMad_RelativeDirectory_Section_ParentPath_tl {
    \g__UWMad_RelativeDirectory_Chapter_CurrentPath_tl/
}
```

Variable declarations and default initializations for Subsection directories.

```
\int_new:N  \g__UWMad_RelativeDirectory_Subsection_Count_int
\tl_new:N   \g__UWMad_RelativeDirectory_Subsection_Prefix_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Subsection_Suffix_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Subsection_CurrentPath_tl
```

```
\tl_new:N   \g__UWMad_RelativeDirectory_Subsection_CurrentName_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Subsection_ParentPath_tl
\tl_gset:Nn \g__UWMad_RelativeDirectory_Subsection_ParentPath_tl {
    \g__UWMad_RelativeDirectory_Section_CurrentPath_tl/
}
```

Variable declaration for graphics inclusion

```
\tl_new:N   \g__UWMad_RelativeDirectory_Graphics_DirectoryName_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Graphics_Extension_tl
\tl_new:N   \g__UWMad_RelativeDirectory_Graphics_BaseName_tl
```

Variable declarations for search options.

```
\bool_new:N \g__UWMad_RelativeDirectory_CycleThrough_Graphics_bool
\bool_new:N \g__UWMad_RelativeDirectory_CycleThrough_Files_bool
```

Miscellaneous variable initializations for the system

```
\tl_new:N    \g__UWMad_RelativeDirectory_File_CurrentName_tl
\tl_new:N    \g__UWMad_RelativeDirectory_OptionalPath_tl
\seq_new:N   \g__UWMad_RelativeDirectory_PathStack_Files_seq
\seq_new:N   \g__UWMad_RelativeDirectory_PathStack_Graphics_seq
\bool_new:N  \g__UWMad_RelativeDirectory_IsFileFound_bool
```

Miscellaneous control sequence initializations for the system.

```
\cs_new:Nn \UWMad_RelativeDirectory_Chapter_SetName:    {}
\cs_new:Nn \UWMad_RelativeDirectory_Section_SetName:    {}
\cs_new:Nn \UWMad_RelativeDirectory_Subsection_SetName: {}
```

9.2 Back End Code

All of the underlying `expl3` code for this module is in this section.

9.2.1 File Inclusion

Special hooks for the automatic naming function below.

```
\cs_new:Nn \UWMad_RelativeDirectory_SetName_Increment_Hook_Chapter: {
    \int_gset:cn {g__UWMad_RelativeDirectory_Section_Count_int}{0}
    \int_gset:cn {g__UWMad_RelativeDirectory_Subsection_Count_int}{0}
}
\cs_new:Nn \UWMad_RelativeDirectory_SetName_Increment_Hook_Section: {
    \int_gset:cn {g__UWMad_RelativeDirectory_Subsection_Count_int}{0}
}
\cs_new:Nn \UWMad_RelativeDirectory_SetName_Increment_Hook_Subsection: {}
```

Directory name-setting functions.

```
\cs_new:Nn \UWMad_RelativeDirectory_SetName_None:n {
    \tl_gset:cx   {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl} {
        \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Prefix_tl}
        \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Suffix_tl}
    }
}
\cs_new:Nn \UWMad_RelativeDirectory_SetName_Increment:n {
    \use:c{UWMad_RelativeDirectory_SetName_Increment_Hook_ #1 :}
    \int_gincr:c  {g__UWMad_RelativeDirectory_ #1 _Count_int}
    \tl_gset:cx   {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl} {
        \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Prefix_tl}
        \int_to_arabic:n{
            \int_use:c{g__UWMad_RelativeDirectory_ #1 _Count_int}
        }
        \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Suffix_tl}
    }
}
\cs_new:Nn \UWMad_RelativeDirectory_SetName_Same:n {
    \tl_gset:cx   {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl} {
        \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Prefix_tl}
        \g__UWMad_RelativeDirectory_File_CurrentName_tl
        \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Suffix_tl}
    }
}
```

Name and path setter.

```
\cs_new:Nn \UWMad_RelativeDirectory_SetNameAndPath:n {

    \tl_gclear:c {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
    \tl_gclear:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}

    \tl_if_blank:VTF {\g__UWMad_RelativeDirectory_OptionalPath_tl} {
        \use:c {UWMad_RelativeDirectory_ #1 _SetName:}
    } {
        \tl_gset_eq:cN
            {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
            \g__UWMad_RelativeDirectory_OptionalPath_tl
    }
    \tl_gset:cx   {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl} {
        \tl_use:c {g__UWMad_RelativeDirectory_ #1 _ParentPath_tl}
        \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
    }
}
```

The default push function pushes to both the file and graphics stacks. However, if the user defines a single (the only) graphics folder, a files-only push function is also defined that will be used when that option is set.

```
\cs_new:Nn \__UWMad_RelativeDirectory_StackPush_Default:n {
    \tl_gset_eq:Nc
        \g_tmpa_tl
        {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
    \tl_if_blank:VTF {\g_tmpa_tl} { } {
        \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Files_seq {
            \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
        }
        \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Graphics_seq {
            \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
        }
    }
}
\cs_new:Nn \__UWMad_RelativeDirectory_StackPush_Files:n {
    \tl_gset_eq:Nc
        \g_tmpa_tl
        {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
    \tl_if_blank:VTF {\g_tmpa_tl} { } {
        \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Files_seq {
            \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
        }
    }
}
```

The default push function uses the default function above. If the user sets a graphics directory name (in which there may be multiple graphics directories in all subdirectories), this will be re-defined.

```
\cs_new:Nn \__UWMad_RelativeDirectory_StackPush:n {
    \__UWMad_RelativeDirectory_StackPush_Default:n{#1}
}
```

Pre-stack update functions for the supported sections.

```
\cs_new:Nn \UWMad_RelativeDirectory_UpdateStack_Chapter_PreHook: {
    \seq_gclear:N \g__UWMad_RelativeDirectory_PathStack_Files_seq
    \seq_gclear:N \g__UWMad_RelativeDirectory_PathStack_Graphics_seq
}
\cs_new:Nn \UWMad_RelativeDirectory_UpdateStack_Section_PreHook: {}
\cs_new:Nn \UWMad_RelativeDirectory_UpdateStack_Subsection_PreHook: {}
```

This function updates the current name, path, and stack(s). Chapters inclusions always clear the stacks.

```
\cs_new:Nn \UWMad_RelativeDirectory_UpdateStack:n {
    \use:c {UWMad_RelativeDirectory_UpdateStack_ #1 _PreHook:}
    \UWMad_RelativeDirectory_SetNameAndPath:n{#1}
    \__UWMad_RelativeDirectory_StackPush:n{#1}
}
```

Two file inputers: one cycles through the current path stack searching for the file from deepest to highest and the other only searches the deepest (i.e., current) directory.

```
\cs_new:Nn \UWMad_RelativeDirectory_IncludeFile_CycleThrough: {
    \seq_map_inline:Nn \g__UWMad_RelativeDirectory_PathStack_Files_seq {
        \tl_gset:Nx \g_tmpa_tl {
            ./##1/
            \g__UWMad_RelativeDirectory_File_CurrentName_tl
        }
        \bool_if:NTF \g__UWMad_RelativeDirectory_IsFileFound_bool { } {
            \file_if_exist:nTF { \g_tmpa_tl } {
                \file_input:n{ \g_tmpa_tl }
                \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
                \seq_map_break:
            } { }
        }
    }
}
\cs_new:Nn \UWMad_RelativeDirectory_IncludeFile_CheckDeepest: {

    \seq_get:NN
        \g__UWMad_RelativeDirectory_PathStack_Files_seq
        \g_tmpa_tl
    \tl_gset:Nx \g_tmpa_tl {
            ./\g_tmpa_tl/
            \g__UWMad_RelativeDirectory_File_CurrentName_tl
    }
    \file_if_exist:nTF {\g_tmpa_tl} {
        \file_input:n{\g_tmpa_tl}
        \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
    } { }
}
```

Driver functions used in the user-front end.

```
\cs_new:Nn \UWMad_RelativeDirectory_IncludeDriver:nn {
    \tl_gset:Nn \g__UWMad_RelativeDirectory_File_CurrentName_tl {#2}
    \UWMad_RelativeDirectory_UpdateStack:n{#1}
    \UWMad_RelativeDirectory_IncludeFile:
}
\cs_new:Nn \UWMad_RelativeDirectory_IncludeDriver:nnn {

    \tl_gset:Nn \g__UWMad_RelativeDirectory_OptionalPath_tl {#3}
    \tl_gset:Nn \g__UWMad_RelativeDirectory_File_CurrentName_tl {#2}
    \UWMad_RelativeDirectory_UpdateStack:n{#1}
    \UWMad_RelativeDirectory_IncludeFile:
    \tl_gclear:N \g__UWMad_RelativeDirectory_OptionalPath_tl
}
```

This is a wrapper function for the above two functions with two additional behaviors: if the file is not found from the search stack, it will check the topmost TEX directory for the file and issue a warning if it is not found.

```
\msg_new:nnn { UWMadThesis }{ RelativeDirectory / FileNotFound } {
    The~requested~file~'#1'~was~not~found~in~the~current~search~stack~nor~the~
    main~LaTeX~directory~for~the~job~'\c_sys_jobname_str'.
}
\cs_new:Nn \UWMad_RelativeDirectory_IncludeFile: {
    \bool_gset_false:N \g__UWMad_RelativeDirectory_IsFileFound_bool

    \bool_if:NTF \g__UWMad_RelativeDirectory_CycleThrough_Files_bool {
        \UWMad_RelativeDirectory_IncludeFile_CycleThrough:
    } {
        \UWMad_RelativeDirectory_IncludeFile_CheckDeepest:
    }
    \bool_if:NTF \g__UWMad_RelativeDirectory_IsFileFound_bool { } {
        \file_if_exist:nTF {\g__UWMad_RelativeDirectory_File_CurrentName_tl} {
            \file_input:n{ \g__UWMad_RelativeDirectory_File_CurrentName_tl }
            \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
        } {
            \msg_warning:nnx
                { UWMadThesis }
                { RelativeDirectory / FileNotFound }
                { \g__UWMad_RelativeDirectory_File_CurrentName_tl }
        }
    }
}
```

9.2.2 Graphics Inclusion

This code copies the existing `\includegraphics` command such that it can be used in a compatible way with the LaTeX 2$_\varepsilon$ system. This technically breaks the `expl3` naming convention since an |n| argument specifier is not a for double square braces, but it is deemed good enough.

```
\cs_new_eq:NN
    \__UWMad_RelativeDirectory_IncludeGraphics_Original:nn
    \includegraphics
\cs_undefine:N
    \includegraphics
```

This function defines the push procedure when a graphics directory name is given. This function will replace the default stack push if the user defines a graphics directory.

```
\cs_new:Nn \__UWMad_RelativeDirectory_StackPush_FilesAndGraphics:n {
    \tl_gset_eq:Nc
        \g_tmpa_tl
        {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
    \tl_if_blank:VTF {\g_tmpa_tl} { } {
        \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Files_seq {
            \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
        }
        \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Graphics_seq {
            \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
        }
        \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Graphics_seq {
            \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}/
            \g__UWMad_RelativeDirectory_Graphics_DirectoryName_tl
        }
    }
}
```

Two graphics includers: one cycles through the current path stack searching for the file from deepest to highest and the other only searches the deepest (i.e., current graphic's) directory.

```
\cs_new:Nn \UWMad_RelativeDirectory_IncludeGraphics_CycleThrough:n {

    \UWMad_File_PathFileName:NNNx
        \l_tmpa_tl
        \g__UWMad_RelativeDirectory_Graphics_BaseName_tl
        \g__UWMad_RelativeDirectory_Graphics_Extension_tl
        {\g__UWMad_RelativeDirectory_File_CurrentName_tl}

    \seq_map_inline:Nn \g__UWMad_RelativeDirectory_PathStack_Graphics_seq {

        \tl_gset:Nx \g_tmpa_tl {
            ./##1/
            \g__UWMad_RelativeDirectory_File_CurrentName_tl
        }

        \bool_if:NTF \g__UWMad_RelativeDirectory_IsFileFound_bool { } {
            \file_if_exist:nTF { \g_tmpa_tl } {
                \tl_gset:Nx \g_tmpa_tl {
                    ./##1/
                    \g__UWMad_RelativeDirectory_Graphics_BaseName_tl
                }
                \__UWMad_RelativeDirectory_IncludeGraphics_Original:nn
                    [ #1 ] {\g_tmpa_tl}
                \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
                \seq_map_break:
            } { }
        }
    }
}
\cs_new:Nn \UWMad_RelativeDirectory_IncludeGraphics_CheckDeepest:n {

    \seq_get:NN
        \g__UWMad_RelativeDirectory_PathStack_Graphics_seq
        \g_tmpa_tl

    \tl_gset:Nx \g_tmpb_tl {
            ./\g_tmpa_tl/
            \g__UWMad_RelativeDirectory_File_CurrentName_tl
    }

    \UWMad_File_PathFileName:NNNx
        \l_tmpa_tl
        \g__UWMad_RelativeDirectory_Graphics_BaseName_tl
        \g__UWMad_RelativeDirectory_Graphics_Extension_tl
        {\g_tmpb_tl}

    \file_if_exist:nTF { \g_tmpb_tl } {
        \__UWMad_RelativeDirectory_IncludeGraphics_Original:nn
            [ #1 ]
            { \g__UWMad_RelativeDirectory_Graphics_BaseName_tl }
        \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
    } { }
}
```

This is a wrapper function for the above two functions with two additional behaviors: if the graphic is not found from the search stack, it will check the topmost TEX directory and issue a warning if it is still not found.

```
\msg_new:nnn { UWMadThesis }{ RelativeDirectory / GraphicNotFound } {
    The~requested~graphic~'#1'~was~not~found~in~the~current~search~stack~nor~
    the~main~LaTeX~directory~for~the~job~'\c_sys_jobname_str'.
}
\cs_new:Nn \UWMad_RelativeDirectory_IncludeGraphics:n {
    \bool_gset_false:N \g__UWMad_RelativeDirectory_IsFileFound_bool
    \bool_if:NTF \g__UWMad_RelativeDirectory_CycleThrough_Graphics_bool {
        \UWMad_RelativeDirectory_IncludeGraphics_CycleThrough:n{#1}
    } {
        \UWMad_RelativeDirectory_IncludeGraphics_CheckDeepest:n{#1}
    }
    \bool_if:NTF \g__UWMad_RelativeDirectory_IsFileFound_bool { } {
        \file_if_exist:nTF {\g__UWMad_RelativeDirectory_File_CurrentName_tl} {
            \__UWMad_RelativeDirectory_IncludeGraphics_Original:nn
            [ #1 ]
            { \g__UWMad_RelativeDirectory_File_CurrentName_tl }
            \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
        } {
            \msg_warning:nnx
                { UWMadThesis }
                { RelativeDirectory / GraphicNotFound }
                { \g__UWMad_RelativeDirectory_File_CurrentName_tl }
        }
    }
}
```

9.2.3 Key-Value Option Definitions

Being the key definitions

```
\keys_define:nn { UWMadThesis / RelativeDirectory } {
```

Chapter prefix and suffix keys.

```
    chapter-directory-prefix    .tl_gset:N =
        \g__UWMad_RelativeDirectory_Chapter_Prefix_tl,
    chapter-directory-prefix    .default:n =,
    chapter-directory-suffix    .tl_gset:N =
        \g__UWMad_RelativeDirectory_Chapter_Suffix_tl,
    chapter-directory-suffix    .default:n =,
```

Chapter naming conventions

```
    chapter-directory-name      .choice:,
    chapter-directory-name / none .code:n = {
        \cs_gset:Nn \UWMad_RelativeDirectory_Chapter_SetName: {
            \UWMad_RelativeDirectory_SetName_None:n{Chapter}
        }
    },
    chapter-directory-name / same .code:n = {
        \cs_gset:Nn \UWMad_RelativeDirectory_Chapter_SetName: {
            \UWMad_RelativeDirectory_SetName_Same:n{Chapter}
        }
    },
    chapter-directory-name / increment .code:n = {
        \cs_gset:Nn \UWMad_RelativeDirectory_Chapter_SetName: {
            \UWMad_RelativeDirectory_SetName_Increment:n{Chapter}
        }
    },
    chapter-directory-name      .default:n = none,
```

Section prefix and suffix keys.

```
    section-directory-prefix    .tl_gset:N =
        \g__UWMad_RelativeDirectory_Section_Prefix_tl,
    section-directory-prefix    .default:n =,
    section-directory-suffix    .tl_gset:N =
        \g__UWMad_RelativeDirectory_Section_Suffix_tl,
    section-directory-suffix    .default:n =,
```

Section naming conventions

```
    section-directory-name      .choice:,
    section-directory-name / none .code:n = {
        \cs_gset:Nn \UWMad_RelativeDirectory_Section_SetName: {
            \UWMad_RelativeDirectory_SetName_None:n{Section}
        }
    },
    section-directory-name / same .code:n = {
        \cs_gset:Nn \UWMad_RelativeDirectory_Section_SetName: {
            \UWMad_RelativeDirectory_SetName_Same:n{Section}
        }
    },
    section-directory-name / increment .code:n = {
        \cs_gset:Nn \UWMad_RelativeDirectory_Section_SetName: {
            \UWMad_RelativeDirectory_SetName_Increment:n{Section}
        }
    },
    section-directory-name      .default:n = none,
```

Subsection prefix and suffix keys.

```
    subsection-directory-prefix    .tl_gset:N =
        \g__UWMad_RelativeDirectory_Subsection_Prefix_tl,
    subsection-directory-prefix    .default:n =,
    subsection-directory-suffix    .tl_gset:N =
        \g__UWMad_RelativeDirectory_Subsection_Suffix_tl,
    subsection-directory-suffix    .default:n =,
```

Subsection naming conventions

```
    subsection-directory-name      .choice:,
    subsection-directory-name / none .code:n = {
        \cs_gset:Nn \UWMad_RelativeDirectory_Subsection_SetName: {
            \UWMad_RelativeDirectory_SetName_None:n{Subsection}
        }
    },
    subsection-directory-name / same .code:n = {
        \cs_gset:Nn \UWMad_RelativeDirectory_Subsection_SetName: {
            \UWMad_RelativeDirectory_SetName_Same:n{Subsection}
        }
    },
    subsection-directory-name / increment .code:n = {
        \cs_gset:Nn \UWMad_RelativeDirectory_Subsection_SetName: {
            \UWMad_RelativeDirectory_SetName_Increment:n{Subsection}
        }
    },
    subsection-directory-name      .default:n = none,
```

Graphics directory keys.

```
    graphics-directory-name .code:n = {
        \tl_gset:Nn \g__UWMad_RelativeDirectory_Graphics_DirectoryName_tl {
            #1
        }
        \tl_if_blank:nTF { #1 } {
            \cs_gset:Nn \__UWMad_RelativeDirectory_StackPush:n {
                \__UWMad_RelativeDirectory_StackPush_Default:n{##1}
            }
        } {
            \cs_gset:Nn \__UWMad_RelativeDirectory_StackPush:n {
                \__UWMad_RelativeDirectory_StackPush_FilesAndGraphics:n{##1}
            }
        }
    },
    the-only-graphics-directory .code:n = {
        \bool_set_false:N
            \g__UWMad_RelativeDirectory_CycleThrough_Graphics_bool
        \seq_gclear:N \g__UWMad_RelativeDirectory_PathStack_Graphics_seq
        \seq_gpush:Nn \g__UWMad_RelativeDirectory_PathStack_Graphics_seq {
            #1
        }
        \cs_gset:Nn \UWMad_RelativeDirectory_UpdateStack_Chapter_PreHook: {
            \seq_gclear:N \g__UWMad_RelativeDirectory_PathStack_Files_seq
        }
        \cs_gset:Nn \__UWMad_RelativeDirectory_StackPush:n {
            \__UWMad_RelativeDirectory_StackPush_Files:n{##1}
        }
    },
```

Path search keys.

```
    cycle-file-paths .bool_gset:N =
        \g__UWMad_RelativeDirectory_CycleThrough_Files_bool,
    cycle-file-paths .default:n = false,
    cycle-graphic-paths .bool_gset:N =
        \g__UWMad_RelativeDirectory_CycleThrough_Graphics_bool,
    cycle-graphic-paths .default:n = true
}
```

Set the default values for the keys.

```
\keys_set:nn { UWMadThesis / RelativeDirectory } {
    chapter-directory-prefix,
    chapter-directory-suffix,
    section-directory-prefix,
    section-directory-suffix,
    subsection-directory-prefix,
    subsection-directory-suffix,
    chapter-directory-name,
    section-directory-name,
    subsection-directory-name,
    cycle-file-paths,
    cycle-graphic-paths
}
```

9.3 User Front Ends

```
\DeclareDocumentCommand \Include { m o m } {
    \IfNoValueTF { #2 } {
        \UWMad_RelativeDirectory_IncludeDriver:nnn{#1}{#3}{}
    } {
        \UWMad_RelativeDirectory_IncludeDriver:nnn{#1}{#3}{#2}
    }
}
\DeclareDocumentCommand \IncludeChapter { o m } {
    \IfNoValueTF { #1 } {
        \UWMad_RelativeDirectory_IncludeDriver:nnn{Chapter}{#2}{}
    } {
        \UWMad_RelativeDirectory_IncludeDriver:nnn{Chapter}{#2}{#1}
    }
}
\DeclareDocumentCommand \IncludeSection { o m } {
    \IfNoValueTF { #1 } {
        \UWMad_RelativeDirectory_IncludeDriver:nnn{Section}{#2}{}
    } {
        \UWMad_RelativeDirectory_IncludeDriver:nnn{Section}{#2}{#1}
    }
}
\DeclareDocumentCommand \IncludeSubsection { o m } {
    \IfNoValueTF { #1 } {
        \UWMad_RelativeDirectory_IncludeDriver:nnn{Subsection}{#2}{}
    } {
        \UWMad_RelativeDirectory_IncludeDriver:nnn{Subsection}{#2}{#1}
    }
}
\DeclareDocumentCommand \IncludeGraphics { o m } {
    \tl_gset:Nn \g__UWMad_RelativeDirectory_File_CurrentName_tl {#2}
    \IfValueTF { #1 } {
        \UWMad_RelativeDirectory_IncludeGraphics:n{#1}
    } {
        \UWMad_RelativeDirectory_IncludeGraphics:n{}
    }
}
\cs_new_eq:NN
    \includegraphics
    \IncludeGraphics

\ExplSyntaxOff
```

Change History

1.0

General: Hello 1