

# UWMadThesis Class Manual

Troy Christopher Haskin

# Table of Contents

<b>I</b>	<b>User Guide</b>	<b>1</b>
<b>1</b>	<b>Thesis and PDF Information</b>	<b>2</b>
1.1	Required . . . . .	2
1.2	Optional . . . . .	4
1.3	Accessors . . . . .	6
<b>2</b>	<b>Special Pages</b>	<b>8</b>
2.1	Title Page . . . . .	8
2.2	License Page . . . . .	8
2.2.1	Copyright . . . . .	9
2.2.2	Creative Commons . . . . .	11
<b>3</b>	<b>Layout And Style</b>	<b>15</b>
3.1	Captions . . . . .	15
3.2	Links . . . . .	15
3.2.1	Link Colors . . . . .	16
3.2.2	References . . . . .	16
3.3	Paragraph Spacing . . . . .	17
<b>4</b>	<b>Getting Started</b>	<b>18</b>
4.1	Options On-Load . . . . .	18
4.2	Feature Options . . . . .	18

4.3	Identification Commands . . . . .	18
<b>5</b>	<b>Sectioning</b>	<b>19</b>
5.1	Front Matter . . . . .	19
5.2	Appendix . . . . .	19
5.3	Table of Contents Tweaks . . . . .	20
<b>6</b>	<b>List Environments</b>	<b>21</b>
6.1	Nomenclature . . . . .	21
6.1.1	Command Descriptions . . . . .	22
6.1.2	Examples . . . . .	23
6.1.3	Customization . . . . .	24
6.2	Acronym . . . . .	26
6.2.1	Description . . . . .	26
6.2.2	Example . . . . .	26
6.2.3	Acronym Customization . . . . .	27
<b>7</b>	<b>Math</b>	<b>29</b>
7.1	Derivative Commands . . . . .	29
7.2	Operators . . . . .	33
7.3	Miscellaneous Commands . . . . .	37
<b>8</b>	<b>Programming</b>	<b>40</b>
<b>9</b>	<b>Relative Directory Includes</b>	<b>41</b>
9.1	File Inclusion . . . . .	42

	iii
9.2 Naming Conventions . . . . .	42
9.2.1 Increment . . . . .	43
9.2.2 Same . . . . .	44
9.2.3 None . . . . .	44
9.3 Including Graphics . . . . .	45
9.4 Search Controls . . . . .	46
9.5 User Options . . . . .	46
<b>II Implementation</b>	<b>48</b>
<b>1 Front Matter</b>	<b>49</b>
1.1 expl3 Package and Identification . . . . .	49
1.2 Identification and Defaults . . . . .	50
1.3 Options . . . . .	50
1.4 Package Loads . . . . .	52
1.4.1 Hyperref Prior . . . . .	52
1.4.2 Hyperref Now . . . . .	53
1.4.3 Hyperref Forever . . . . .	53
1.5 Key-Value Interface . . . . .	54
<b>2 Programming</b>	<b>56</b>
2.1 Utility Commands . . . . .	56
2.2 Collections . . . . .	61
2.2.1 Stacks . . . . .	61

	iv
2.2.2	Queues . . . . . 63
2.2.3	Dequeues . . . . . 66
2.2.4	Hashes . . . . . 69
2.3	User-Level Abstractions . . . . . 72
2.3.1	Utility Commands . . . . . 72
2.3.2	Command Creators . . . . . 74
2.3.3	Types . . . . . 76
<b>3</b>	<b>Layout And Styles 79</b>
3.1	Float Styles . . . . . 79
3.2	Links . . . . . 80
3.3	Page Layout . . . . . 83
<b>4</b>	<b>Sectioning 85</b>
4.1	Appendix . . . . . 85
4.2	Front Matter . . . . . 86
4.3	TOC Tweaks . . . . . 88
4.4	Section-Level Commands . . . . . 91
<b>5</b>	<b>Math 94</b>
5.1	Derivative Commands . . . . . 94
5.2	Operators and Functions . . . . . 97
5.3	Miscellaneous Functions . . . . . 98
<b>6</b>	<b>ListOf 101</b>

6.1	Nomenclature . . . . .	104
<b>7</b>	<b>Thesis and PDF Information</b>	<b>119</b>
7.1	Metadata clist and Aux Write . . . . .	119
7.2	Thesis Information . . . . .	120
7.3	Committee Member List . . . . .	125
7.4	PDF Metadata . . . . .	127
<b>8</b>	<b>Special Pages</b>	<b>130</b>
8.1	MakeTitlePage . . . . .	130
8.2	LicensePage . . . . .	132
8.2.1	Copyright . . . . .	132
8.2.2	Creative Commons . . . . .	133
8.2.3	LicensePage Proper . . . . .	136
<b>9</b>	<b>Relative Directory Input</b>	<b>140</b>
9.1	Declarations and Initializations . . . . .	140
9.2	Back End Code . . . . .	141
9.2.1	File Inclusion . . . . .	142
9.2.2	Graphics Inclusion . . . . .	146
9.2.3	Key-Value Option Definitions . . . . .	148
9.3	User Front Ends . . . . .	152
	<b>Change History</b>	<b>154</b>

# Part I

## User Guide

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

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  $\text{\LaTeX}$  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.

---

<code>\Title</code>	<code>\Title    {\textit{title}}</code>
<code>\Author</code>	<code>\Author   {\textit{author name}}</code>
<code>\Program</code>	<code>\Program {\textit{program}}</code>

---

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



---

<code>\Degree</code>	<code>\Degree {<math>\langle degree \rangle</math>}</code>
<code>\Doctorate</code>	<code>\Doctorate</code>
<code>\Masters</code>	<code>\Masters</code>
<code>\Bachelors</code>	<code>\Bachelors</code>

---

Only one of these commands is required to define the  $\{\langle degree \rangle\}$  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  $\{\langle degree \rangle\}$  to “Doctor of Philosophy”
- `\Masters` sets  $\{\langle degree \rangle\}$  to “Master’s”
- `\Bachelors` sets  $\{\langle degree \rangle\}$  to “Bachelor’s”

---

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

---

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

Since  $\{\langle defense date \rangle\}$  has no parsing performed on it, any valid text or expandable argument may be entered and will be typeset as-entered.

---

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

---

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

---

<code>\CommitteeMember</code>	<code>\CommitteeMember {<i>&lt;member name&gt;</i>} {<i>&lt;member position&gt;</i>} {<i>&lt;member program&gt;</i>}</code>
<code>\Advisor</code>	<code>\Advisor {<i>&lt;advisor name&gt;</i>} {<i>&lt;advisor position&gt;</i>} {<i>&lt;advisor program&gt;</i>}</code>
<code>\Adviser</code>	<code>\Adviser {<i>&lt;advisor name&gt;</i>} {<i>&lt;advisor position&gt;</i>} {<i>&lt;advisor program&gt;</i>}</code>

---

`\CommitteeMember` can be used as many times as required. However, if the list of members becomes too large, formatting of the [Title Page](#) will suffer.

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.

---

<code>\DocumentType</code>	<code>\DocumentType {<math>\langle document type \rangle</math>}</code>
<code>\Dissertation</code>	<code>\Dissertation</code>
<code>\DoctoralThesis</code>	<code>\DoctoralThesis</code>
<code>\MastersThesis</code>	<code>\MastersThesis</code>
<code>\Thesis</code>	<code>\Thesis</code>
<code>\Prelim</code>	<code>\Prelim</code>

---

By default, the `\MakeTitlePage` command prints the phrase “A  $\langle document type \rangle$  submitted in partial fulfillment of the requirements for the degree of” on the title page”. The default  $\langle document type \rangle$  is “report”. This command sets the value to any valid text.

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  $\langle document type \rangle$  to something similar to their command name:

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

---

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

---

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

---

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

---

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

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

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

These are very thin distinctions and complicated by the typical workflow of a  $\text{\LaTeX}$  document: installing a  $\text{\TeX}$  distribution, editing a text file in  $\text{\TeX}/\text{\LaTeX}$  editor, and running the document through a  $\text{\TeX}$  engine with the  $\text{\LaTeX}$  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  $\text{\TeX}$  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 L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> 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:

---

<code>\LicenseOwner</code>	<code>\LicenseOwner {<i>&lt;owner name&gt;</i>}</code>
<code>\LicenseYear</code>	<code>\LicenseYear {<i>&lt;year&gt;</i>}</code>

---

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

### 2.2.1 Copyright

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

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

picture or other audiovisual work, to display the copyrighted work publicly

6. in the case of sound recordings, to perform the copyrighted work publicly by means of a digital audio transmission

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

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

Section 408 of the Copyright Act states: for any work produced after 1978, “the owner of copyright or of any exclusive right in the work may obtain registration of the copyright claim by delivering to the Copyright Office the deposit specified by this section, together with the application and fee”. In others words, a copy of the work can be submitted to the Copyright Office and subsequently placed in the Library of Congress for official recognition of copyright. However, registration is not compulsory since “[s]uch registration is not a condition of copyright protection”.

---

**\Copyright**    \Copyright

---

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

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

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



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

Copyright © 3001 by Theodore Huxton

### 2.2.2 Creative Commons

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

---

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

---

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

---

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

---

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

An example of usage would be

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

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

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

Troy Christopher Haskin, 2014

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

---

<code>\CCVersion</code>	<code>\CCVersion{<i>&lt;CC version&gt;</i>}</code>
<code>\CCPorting</code>	<code>\CCPorting{<i>&lt;CC porting&gt;</i>}</code>
<code>\CCURL</code>	<code>\CCURL {<i>&lt;CC link&gt;</i>}</code>
<code>\CCURLText</code>	<code>\CCURLText{<i>&lt;CC link text&gt;</i>}</code>

---

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

An example of usage would be

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

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

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

## Feature Set 3

# Layout And Style

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

---

<code>\MakeLinksTheseColors</code>	<code>\MakeLinksTheseColors{\link color}{\cite color}{\url color}</code>
------------------------------------	--

---

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.

---

<code>\MakeLinksThisColor</code>	<code>\MakeLinksThisColor{\color}</code>
----------------------------------	--

---

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.

---

<code>\MakeLinksBlack</code>	<code>\MakeLinksBlack</code>
<code>\MakeLinksBlue</code>	<code>\MakeLinksBlue</code>
<code>\MakeLinksRed</code>	<code>\MakeLinksRed</code>

---

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 commands have been created.

---

<code>\PadParagraphs</code>	<code>\PadParagraphs</code>
-----------------------------	-----------------------------

---

This command adds 1em of vertical space between paragraphs with no indentation. This is the default style of this class.

---

<code>\IndentParagraphs</code>	<code>\IndentParagraphs</code>
--------------------------------	--------------------------------

---

This command adds 1.5em of indent at the beginning of paragraphs (save those that follow section heads) with no extra vertical space.

## 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  $\text{\LaTeX} 2_{\epsilon}$  are `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, and `subparagraph`. The `UWMadThesis` class defines some new section commands and makes some other adjustments to the default commands.

## 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.

---

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

---

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

## 5.2 Appendix

The standard method of including appendices in  $\text{\LaTeX}$  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  $\text{\LaTeX}$  documents and, therefore, redefines `\appendix`.

---

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

---

The appendix commands now acts 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.

---

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

---

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

---

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

---

Camel-cased versions of the standard  $\text{\LaTeX}$  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.

---

<code>\Entry</code>	<code>\Entry{<i>&lt;symbol&gt;</i>}{<i>&lt;description&gt;</i>}</code>
	<code>\Entry{<i>&lt;symbol&gt;</i>}{<i>&lt;units&gt;</i>}{<i>&lt;description&gt;</i>}</code>

---

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

---

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

---

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

### 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.
$\rho$	Density
$\mu$	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

$\rho$	Density
$\mu$	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`.

---

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

---

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

### 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 <code>\Group</code> command	section	se
subgroup-section	Section level for <code>\Subgroup</code> command	subsection	se
main-title	Title for the nomenclature	Nomenclature	t
group-title	Title for the <code>\Group</code> command	—	t
subgroup-title	Title for the <code>\Subgroup</code> 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.

---

<code>\deriv</code>	<code>\deriv {\langle function \rangle} {\langle variable \rangle} {\langle order \rangle}</code>
<code>\pderiv</code>	<code>\pderiv {\langle function \rangle} {\langle variable \rangle} {\langle order \rangle}</code>
<code>\tderiv</code>	<code>\tderiv {\langle function \rangle} {\langle variable \rangle} {\langle order \rangle}</code>

---

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

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

```

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

```

and is typeset as

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

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

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

---

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

---

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

```

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

```

and is typeset as

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

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

---

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

---

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

---

 $\backslash\mathrm{derivSymbol}$      $\backslash\mathrm{derivSymbol}$ 
 $\backslash\mathrm{pderivSymbol}$ 


---

 $\backslash\mathrm{tderivSymbol}$ 


---

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,  $\backslash\mathrm{pderivSymbol}$  will be appear as  $\partial$ .

---

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


---

 $\backslash\mathrm{tderivSymbolChange}$ 


---

These commands will TEMPORARILY change the symbol used by the associated `deriv` commands. The symbol will revert back to the original, default value after leaving the  $\mathrm{T\!E\!X}$  group where the switch was made (more often than not for  $\mathrm{L\!A\!T\!E\!X}$  users, this means “upon exiting an environment”). For example:

```

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

```

typesets as

$$\frac{\mathrm{d}U}{\mathrm{d}t} = \frac{\delta Q}{\delta t} - \frac{\delta W}{\delta t} \quad (6)$$

and now, after the environment, the  $\backslash\mathrm{derivSymbol}$  is once again ‘d’.

---

<code>\derivSymbolChangeDefault</code>	<code>\derivSymbolChangeDefault {⟨symbol⟩}</code>
<code>\pderivSymbolChangeDefault</code>	<code>\pderivSymbolChangeDefault {⟨symbol⟩}</code>
<code>\tderivSymbolChangeDefault</code>	<code>\tderivSymbolChangeDefault {⟨symbol⟩}</code>

---

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

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

and now, after the environment, the `\derivSymbol` is ‘ $\delta$ ’.

---

<code>\DelimiterChangeDefault</code>	<code>\DelimiterChangeDefault {⟨left delim⟩} {⟨right delim⟩}</code>
--------------------------------------	---

---

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

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

and is typeset as

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

and notice that the `\derivSymbol` is still  $\delta$ .

## 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.

---

$\backslash\text{Sup}$   
 $\backslash\text{Inf}$

---

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

```
\begin{align}
  \backslash\text{Inf}_{x \in \mathbb{R}} \{0 < x < 1\} \&= 0 \\\[0.50em]
  \backslash\text{Sup}_{x \in \mathbb{R}} \{0 < x < 1\} \&= 1
\end{align}
```

is typeset as

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

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

---

$\backslash\text{Lim}$

---

The limit operator:

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

is typeset as

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



---

 $\backslash\text{Min}$ 
 $\backslash\text{Max}$ 


---

The maximum and minimum value operators

```

\begin{equation}
\begin{align}
\text{\Min}_{x \in \mathbb{R}} \text{\Sin}(x) &= -1 \quad \ll[0.50em] \\
\text{\Max}_{x \in \mathbb{R}} \text{\Sin}(x) &= +1
\end{align}
\end{equation}

```

is typeset as

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

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

---

 $\backslash\text{ArgMin}$ 
 $\backslash\text{ArgMax}$ 


---

The maximum and minimum argument operators

```

\begin{equation}
\begin{align}
\text{\ArgMin}_{x \in \mathbb{R}} \text{\Sin}(x) &= \frac{3\pi}{2} + 2\pi n \\
\text{\ArgMax}_{x \in \mathbb{R}} \text{\Sin}(x) &= \frac{\pi}{2} + 2\pi n
\end{align}
\end{equation}

```

is typeset as

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

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

---

 $\backslash\text{Abs}$ 
 $\backslash\text{Ln}$  $\backslash\text{Log}$  $\backslash\text{Exp}$ 

Common set of operators in uppercase form.

---

 $\backslash\text{Cos}$ 
 $\backslash\text{Sin}$  $\backslash\text{Tan}$  $\backslash\text{Sec}$  $\backslash\text{Csc}$  $\backslash\text{Cot}$ 

Standard trigonometric functions and their reciprocals.

---

 $\backslash\text{Cosh}$ 
 $\backslash\text{Sinh}$  $\backslash\text{Tanh}$  $\backslash\text{Sech}$  $\backslash\text{Csch}$  $\backslash\text{Coth}$ 

Hyperbolic trigonometric functions and their reciprocals.

---

 $\backslash\text{ArcCos}$ 
 $\backslash\text{ArcSin}$  $\backslash\text{ArcTan}$  $\backslash\text{ArcSec}$  $\backslash\text{ArcCsc}$  $\backslash\text{ArcCot}$ 

Standard inverse trigonometric functions and their reciprocals.

---

 $\backslash\text{ArcCosh}$ 
 $\backslash\text{ArcSinh}$  $\backslash\text{ArcTanh}$  $\backslash\text{ArcSech}$  $\backslash\text{ArcCsch}$  $\backslash\text{ArcCoth}$ 

Hyperbolic inverse trigonometric functions and their reciprocals.

## 7.3 Miscellaneous Commands

---

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

---

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

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

---

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

---

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

---

<code>\subs</code>	<code>\subs</code> [ <i><math>\langle space \rangle</math></i> ] { <i><math>\langle text subscript \rangle</math></i> }
<code>\sup</code> s	<code>\sup</code> s [ <i><math>\langle space \rangle</math></i> ] { <i><math>\langle text superscript \rangle</math></i> }
<code>\subsup</code> s	<code>\subsup</code> s [ <i><math>\langle subscript space \rangle</math></i> ] { <i><math>\langle text subscript \rangle</math></i> } [ <i><math>\langle superscript space \rangle</math></i> ] { <i><math>\langle text superscript \rangle</math></i> }

---

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 [ *$\langle space \rangle$* ] 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 ‘ $T_P$ ,  $T_P$ ’.  $T^P$

---

`\OneOver`   `\OneOver`   `{\langle denominator \rangle}`

---

`\oneo`

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

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

is typeset as

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

---

`\dd`   `\dd`   `{\langle variable \rangle}`

---

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

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

is typeset as

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

---

```
\dprime \dprime
```

```
\tprime
```

---

These commands take no arguments and simply mean ‘double prime’ and ‘triple prime’.

For example, the input

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

is typeset as

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

## Feature Set 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

L<sup>A</sup>T<sub>E</sub>X 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 be:

```
\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 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 L<sup>A</sup>T<sub>E</sub>X 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.

---

<code>\IncludeChapter</code>	<code>\IncludeChapter</code> <code>[[<i>path</i>]]{<i>filename</i>}</code>
<code>\IncludeSection</code>	<code>\IncludeSection</code> <code>[[<i>path</i>]]{<i>filename</i>}</code>
<code>\IncludeSubsection</code>	<code>\IncludeSubsection</code> <code>[[<i>path</i>]]{<i>filename</i>}</code>

---

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 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X 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  $\text{\LaTeX}$  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 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  $\langle search\ stack \rangle$  will handle either.

### 9.2.2 Same

Suppose a user has a  $\text{\LaTeX}$  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 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  $\langle search\ stack \rangle$  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) <i>Increment</i>	(B) <i>Same</i>
/Main/	/Main/
-- Main.tex	-- Main.tex
-- Chapter-1	-- Introduction-Chapter
-- Chapter.tex	-- Introduction.tex
-- Section-1.tex	-- Motivation.tex
-- Section-2.tex	-- LiteratureReview.tex
-- Chapter-2	-- Theory-Chapter
-- Chapter.tex	-- Theory.tex
-- Section-1	-- LinearSystems-Section
-- Section.tex	-- LinearSystems.tex
-- Section-2	-- NewtonsMethod-Section
-- Section.tex	-- NewtonsMethods.tex
----	----

conventions described above, the system will require the optional argument for dynamic file searching to be possible. The `UWMadThesis` class  $\langle search\ stack \rangle$  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 by 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.

---

<code>\IncludeGraphics</code>	<code>\IncludeGraphics[<i>&lt;options&gt;</i>]{<i>&lt;graphic name&gt;</i>}</code>
<code>\includegraphics</code>	<code>\includegraphics[<i>&lt;options&gt;</i>]{<i>&lt;graphic name&gt;</i>}</code>

---

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
chapter-directory-prefix	Directory prefix used for <code>\IncludeChapter</code>	—
chapter-directory-suffix	Directory suffix used for <code>\IncludeChapter</code>	—
chapter-directory-name	Naming convention used for <code>\IncludeChapter</code>	none
section-directory-prefix	Directory prefix used for <code>\IncludeSection</code>	—
section-directory-suffix	Directory suffix used for <code>\IncludeSection</code>	—
section-directory-name	Naming convention used for <code>\IncludeSection</code>	none
subsection-directory-prefix	Directory prefix used for <code>\IncludeSubsection</code>	—
subsection-directory-suffix	Directory suffix used for <code>\IncludeSubsection</code>	—
subsection-directory-name	Naming convention used for <code>\IncludeSubsection</code>	none
graphics-directory-name	Graphics directory name for multiple directories	—
the-only-graphics-directory	Graphics directory name for a single directory	—
cycle-file-paths	Search the entire file stack or only the lowest level	false
cycle-graphics-paths	Search the entire graphic stack or only the lowest level	true

<sup>†</sup> Valid choices: none, same, increment

## Part II

# Implementation

## Module 1

# Front Matter

Much of this class is written using the L<sup>A</sup>T<sub>E</sub>X3 Programming Layer; this will be denoted as `expl3`. The `expl3` is the first piece of a new system designed to succeed L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> in the future. However, while the programming layer is solid and remarkable, a lot of presentation work still needs to be done. Therefore, this class uses L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> code where necessary and will hopefully be slowly pulled out as needed. The good news is that since everything is more-or-less an abstraction of T<sub>E</sub>X, it should work together well.

## 1.1 `expl3` Package and Identification

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

```

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

10 \ExplSyntaxOn
```

## 1.2 Identification and Defaults

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

```

11 \tl_const:Nn \c__UWMad_Class_Name_tl      {UWMadThesis}
12 \tl_const:Nn \c__UWMad_Class_Version_tl   {1.0}
13 \tl_const:Nn \c__UWMad_Class_Date_tl      {2014/04/01}
14 \tl_const:Nn \c__UWMad_Class_Description_tl {
15     LaTeX2e+~Thesis~Class~for~UW~Madison
16 }
17 \tl_const:Nn \c__UWMad_UniversityLong_tl   {University~of~Wisconsin--Madison}
18 \tl_const:Nn \c__UWMad_UniversityShort_tl  {UW--Madison}

```

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

```

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

```

In an effort to allow the thesis class to adapt to new underlying classes, the class that the `UWMadThesis` class loads is declared as a mutable token list. The default is the  $\text{\LaTeX}$  base class `report`.

```

22 \tl_new:N      \g_UWMad_ParentClass_tl
23 \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.

```

24 \msg_new:nnn{ UWMadThesis }{ Options / StyleViolation }{
25     Option~'#1'~violates~\c_UWMadUniversityShort_tl{ }~
26     Dissertation~Guidelines;~consider~omission
27 }
28 \cs_new:Nn \__UWMad_FrontMatter_StyleWarning:n {
29     \msg_warning:nnn { UWMadThesis }{ Options / StyleViolation } { #1 }

```



```

30 \PassOptionsToClass{#1}{\g_UWMad_ParentClass_tl}
31 }

```

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

```

32 \bool_new:N \g__UWMad_MathTweaks_bool
33 \bool_gset_true:N \g__UWMad_MathTweaks_bool

```

Declare the options.

```

34 \DeclareOption{NoMath} {
35   \bool_gset_false:N \g__UWMad_MathTweaks_bool
36 }
37 \DeclareOption{Quiet} {
38   \msg_redirect_module:nnn { UWMadThesis } { warning } { none }
39 }

```

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

```

40 \DeclareOption{article} {
41   \tl_gset:Nn \g_UWMad_ParentClass_tl {article}
42 }

```

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

```

43 \DeclareOption{a4paper} {
44   \__UWMad_FrontMatter_StyleWarning:n {\CurrentOption}
45 }
46 \DeclareOption{twoside} {
47   \__UWMad_FrontMatter_StyleWarning:n {\CurrentOption}
48 }

```

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.

```

49 \DeclareOption{l3doc} {
50   \tl_gset:Nn \g_UWMad_ParentClass_tl {l3doc}
51   \tl_const:cn {ver@thumbpdf.sty} {}
52 }

```

Pass all remaining options to the base class.

```

53 \DeclareOption*{
54     \PassOptionsToClass {
55         \CurrentOption
56     } {
57         \g_UWMad_ParentClass_t1
58     }
59 }

```

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

```

60 \ProcessOptions\relax
61 \LoadClass[oneside,12pt]{\g_UWMad_ParentClass_t1}[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`.

```

62 \RequirePackage{xparse}
63 \RequirePackage{fixltx2e}
64 \RequirePackage{microtype}
65 \RequirePackage{array}
66 \RequirePackage{float}
67 \RequirePackage{graphicx}
68 \RequirePackage{setspace}
69 \RequirePackage{geometry}

```

Load the  $\mathcal{AMS}$  suite.

```

70 \RequirePackage{amsmath}
71 \RequirePackage{amsfonts}
72 \RequirePackage{amssymb}
73 \RequirePackage{mathtools}

```

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

```

74 \bool_if:nTF {\xetex_if_engine_p: || \luatex_if_engine_p:} {
75   \RequirePackage{fontspec}
76   \defaultfontfeatures{Ligatures={TeX}}
77   \setmainfont
78     [SmallCapsFont = {Latin~Modern~Roman~Caps}]
79     {Latin~Modern~Roman}
80   %
81   \RequirePackage{polyglossia}
82   \setmainlanguage[variant = usmax]{english}
83 } {
84   \RequirePackage{lmodern}
85   \RequirePackage[T1]{fontenc}
86   %
87   \RequirePackage[english]{babel}
88 }

```

### 1.4.2 Hyperref Now

Load `hyperref` and `bookmark`.

```

89 \RequirePackage{hyperref}
90 \RequirePackage{bookmark}

```

### 1.4.3 Hyperref Forever

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

```

91 \RequirePackage[noabbrev,nameinlink]{cleveref}
92 \RequirePackage[usenames,dvipsnames,svgnames,table,hyperref]{xcolor}
93 \RequirePackage{caption}
94 \RequirePackage{subcaption}

```

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

```

95 \DeclareDocumentCommand \UWMadClass { } {
96   \texttt{\c__UWMad_Class_Name_tl}~class
97 }
98 \DeclareDocumentCommand \UWMadClassVersion { } {
99   \c__UWMad_Class_Version_tl
100 }
101 \DeclareDocumentCommand \UWMadClassDate { } {
102   \c__UWMad_Class_Date_tl
103 }
104 \DeclareDocumentCommand \UWMadLong { } {
105   \c__UWMad_UniversityLong_tl
106 }
107 \DeclareDocumentCommand \UWMadShort { } {
108   \c__UWMad_UniversityShort_tl
109 }
110 %

```

## 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.

```

111 \cs_generate_variant:Nn \keys_set:nn {nf}
112 \tl_new:N \l__UWMad_Setup_ModuleName_tl
113 \clist_new:N \l__UWMad_Setup_OptionList_clist
114 \cs_new:Nn \__UWMad_Setup_ProcessInput:n {
115     \seq_set_split:Nnn \l_tmpa_seq {,} {#1}
116     \seq_map_inline:Nn \l_tmpa_seq {
117         \seq_set_split:Nnn \l_tmpb_seq {/} {##1}
118         \seq_pop:NN \l_tmpb_seq \l__UWMad_Setup_ModuleName_tl
119         \seq_pop:NN \l_tmpb_seq \l__UWMad_Setup_OptionList_clist
120         \clist_map_inline:Nn \l__UWMad_Setup_OptionList_clist {
121             \tl_set:Nx \l_tmpa_tl {
122                 \l__UWMad_Setup_ModuleName_tl / \tl_trim_spaces:n{####1}
123             }
124             \exp_args:Nnf
125                 \keys_set:nn { UWMadThesis } { \l_tmpa_tl }
126         }
127     }
128 }
129 \DeclareDocumentCommand \UWMadSetup { m } {
130     \__UWMad_Setup_ProcessInput:n{#1}
131 }

```

## Module 2

# Programming

This section outlines the Programming module for the `UWMadThesis` class. It encompasses thin abstractions from the standard `expl3`'s type and collection systems and provides  $\text{\LaTeX 2}_{\varepsilon}$  abstractions for several other features.

## 2.1 Utility Commands

Define some messages for the rest of the module.

```

132 \msg_new:nnn {UWMadThesis} {Programming/UnregisteredVariable} {
133     `#1'~is~not~a~registered~#2.~~The~#2~must~be~defined~
134     before~usage~by~the~function~\string\UWMad_#2_DefineLocal:n~or~
135     \string\UWMad_#2_DefineGlobal:n.
136 }
137 \msg_new:nnn {UWMadThesis} {Programming/Undefined} {
138     The~#2~`#1'~is~undefined.~~The~#2~must~be~defined~
139     before~usage~by~the~function~\string\UWMad_#2_Define:n.
140 }
141 \msg_new:nnn {UWMadThesis} {Programming/Defined} {
142     The~#2~`#1'~is~already~defined~and~will~not~altered.
143 }
```

---

<code>\UWMad_Hook_Prepended:cn</code>	<code>\UWMad_Hook_Prepended:cn {&lt;command name&gt;} {&lt;prepend code&gt;}</code>
<code>\UWMad_Hook_Prepended:Nn</code>	<code>\UWMad_Hook_Prepended:Nn &lt;command&gt; {&lt;prepend code&gt;}</code>

---

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

```

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

---

<code>\UWMad_Hook_Append:cn</code>	<code>\UWMad_Hook_Append:cn {&lt;command name&gt;} {&lt;append code&gt;}</code>
<code>\UWMad_Hook_Append:Nn</code>	<code>\UWMad_Hook_Append:Nn &lt;command&gt; {&lt;append code&gt;}</code>

---

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

```

156 \cs_new:Nn \UWMad_Hook_Append:cn {
157     \cs_new_eq:cc {#1-Default:} {#1}
158     \cs_gset:cn {#1:} {\cs:w #1-Default:\cs_end: #2}
159     \cs_undefine:c {#1}
160     \cs_new_eq:cc {#1} {#1:}
161 }
162 \cs_new:Nn \UWMad_Hook_Append:Nn {
163     \cs_new_eq:cN {\string#1-Default:} #1
164     \cs_gset:cn {\string#1:} {\cs:w\string#1-Default:\cs_end: #2}
165     \cs_undefine:N #1
166     \cs_new_eq:Nc #1 {\string#1:}
167 }
```

---

<code>\UWMad_Definition_Swap:Nn</code>	<code>\UWMad_Definition_Swap:Nn &lt;command&gt; {\replacement code}</code>
<code>\UWMad_Definition_Swap:cn</code>	<code>\UWMad_Definition_Reset:N &lt;command&gt;</code>
<code>\UWMad_Definition_Reset:N</code>	<code>\UWMad_Definition_Swap:cn {\command name} {\replacement code}</code>
<code>\UWMad_Definition_Reset:c</code>	<code>\UWMad_Definition_Reset:c {\command name}</code>

---

These commands “swap” in a new definition of a command and, when called, reset it to its default definition.

```

168 \cs_new:Nn \UWMad_Definition_Swap:Nn {
169     \cs_if_exist:NTF #1 {
170         \cs_new_eq:cN {\string#1-Default:} #1
171         \cs_gset_eq:Nc #1 {#2}
172     } {
173         \cs_new:Nn #1 {#2}
174     }
175 }
176 \cs_new:Nn \UWMad_Definition_Reset:N {
177     \cs_if_exist:cTF {\string#1-Default:} {
178         \cs_gset_eq:Nc #1 {\string#1-Default:}
179         \cs_undefine:c {\string#1-Default:}
180     } { }
181 }
182 \cs_generate_variant:Nn \UWMad_Definition_Swap:Nn {cn}
183 \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

```
184 \tl_new:N \g__UWMad_File_Path_tl
185 \tl_new:N \g__UWMad_File_Extension_tl
186 \tl_new:N \g__UWMad_File_Marker_Extension_tl
187 \tl_new:N \g__UWMad_File_Marker_Directory_tl
188 \tl_gset:Nn \g__UWMad_File_Marker_Extension_tl {.}
189 \tl_gset:Nn \g__UWMad_File_Marker_Directory_tl {/}
190 \bool_new:N \g__UWMad_File_IsExtensionFound_bool
191 \bool_new:N \g__UWMad_File_IsDirectoryFound_bool
```

Define the body of the function.

```
192 \cs_new:Nn \UWMad_File_GetExtension:nNN {
193
194     \tl_gclear:N \g__UWMad_File_Path_tl
195     \tl_gclear:N \g__UWMad_File_Extension_tl
196     \bool_set_false:N \g__UWMad_File_IsExtensionFound_bool
197     \bool_set_false:N \g__UWMad_File_IsDirectoryFound_bool
198
199     \tl_set:Nx \l_tmpa_tl {
200         \tl_reverse:V {#1}
201     }
202
203     \tl_map_inline:Nn \l_tmpa_tl {
204
205         \tl_set:Nn {\l_tmpb_tl}{##1}
206
207         \bool_if:NTF \g__UWMad_File_IsExtensionFound_bool {
208             \tl_gput_left:Nn \g__UWMad_File_Path_tl {##1}
209         } {
210             \bool_if:NTF \g__UWMad_File_IsDirectoryFound_bool {
211                 \tl_gput_left:Nn \g__UWMad_File_Path_tl {##1}
212             } {
213                 \tl_if_eq:NNTF \l_tmpb_tl \g__UWMad_File_Marker_Extension_tl {
214                     \bool_set_true:N \g__UWMad_File_IsExtensionFound_bool
215                 } {
216                     \tl_if_eq:NNTF \l_tmpb_tl \g__UWMad_File_Marker_Directory_tl {
217                         \bool_gset_true:N \g__UWMad_File_IsDirectoryFound_bool
218                         \tl_gput_left:Nn \g__UWMad_File_Path_tl {##1}
219                     } {
220                         \tl_gput_left:Nn \g__UWMad_File_Extension_tl {##1}
221                     }
222                 }
223             }
224         }
225     }
226 }
```

---

<code>\_UWMad_IfDefined:nnnnT</code>	<code>\_UWMad_IfDefined:nnnnT{&lt;Prefix&gt;}{&lt;ID&gt;}{&lt;Suffix&gt;}{&lt;Type&gt;}{&lt;Code&gt;}</code>
<code>\_UWMad_IfUndefined:nnnnT</code>	<code>\_UWMad_IfUndefined:nnnnT{&lt;Prefix&gt;}{&lt;ID&gt;}{&lt;Suffix&gt;}{&lt;Type&gt;}{&lt;Code&gt;}</code>

---

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

```

241 \cs_new:Nn \_UWMad_IfDefined:nnnnT{
242   \cs_if_exist:cTF {#1#2#3} {
243     #5
244   }{
245       \msg_error:nnnn
246       {UWMadThesis}
247       {Programming/Undefined}
248       {#2}
249       {#4}
250   }
251 }
252 \cs_new:Nn \_UWMad_IfUndefined:nnnnT{
253   \cs_if_free:cTF {#1#2#3} {
254     #5
255   }{
256       \msg_warning:nnnn
257       {UWMadThesis}
258       {Programming/Defined}
259       {#2}
260       {#4}
261   }
262 }
```

---

<code>\_UWMad_IfDefined:nT</code>	<code>\_UWMad_IfDefined:nT{&lt;CommandName&gt;}{&lt;TrueCode&gt;}</code>
<code>\_UWMad_IfUndefined:nT</code>	<code>\_UWMad_IfUndefined:nT{&lt;CommandName&gt;}{&lt;TrueCode&gt;}</code>

---

These commands are simplifications of the above commands and that only take a `{<CommandName>}` and `{<TrueCode>}`.

```

263 \cs_new:Nn \_UWMad_IfDefined:nT{
264   \_UWMad_IfDefined:nnnnT{}{#1}{}{command}{#2}
265 }
266 \cs_new:Nn \_UWMad_IfUndefined:nT{
267   \_UWMad_IfUndefined:nnnnT{}{#1}{}{command}{#2}
268 }
```

## 2.2 Collections

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

All of the collection systems are thin abstractions of `expl3`'s `l3tl`, `l3seq`, and `l3prop` modules to avoid developing one-shot systems while allowing more endeavoring authors access to the features without learning `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.

```

269 \cs_new:Nn \__UWMad_Stack_IfDefined:nT {
270     \__UWMad_IfDefined:nnnnT{g__UWMad_Stack_}{#1}{}{Stack}{#2}
271 }
272 \cs_new:Nn \__UWMad_Stack_IfUndefined:nT{
273     \__UWMad_IfUndefined:nnnnT{g__UWMad_Stack_}{#1}{}{Stack}{#2}
274 }

```

---

\UWMad\_Stack\_Define:n

---

Define a new Stack.

```

275 \cs_new:Nn \UWMad_Stack_Define:n {
276     \__UWMad_Stack_IfUndefined:nT {#1} {
277         \tl_new:c {g__UWMad_Stack_#1}
278     }
279 }

```

---

\UWMad\_Stack\_Clear:n

---

Clear but do not undefine a defined Stack.

```

280 \cs_new:Nn \UWMad_Stack_Clear:n {
281     \__UWMad_Stack_IfDefined:nT {#1} {
282         \tl_gclear:c {g__UWMad_Stack_#1}
283     }
284 }

```

---

\UWMad\_Stack\_Delete:n

---

Clear and undefine a defined Stack.

```

285 \cs_new:Nn \UWMad_Stack_Delete:n {
286     \__UWMad_Stack_IfDefined:nT {#1} {
287         \tl_gclear:c {g__UWMad_Stack_#1}
288         \cs_undefine:c {g__UWMad_Stack_#1}
289     }
290 }

```

---

**\UWMad\_Stack\_Push:nn**

---

Push a value on to a defined Stack.

```

291 \cs_new:Nn \UWMad_Stack_Push:nn {
292   \__UWMad_Stack_IfDefined:nT {#1} {
293     \tl_gput_left:cn {g__UWMad_Stack_#1} {#2}
294   }
295 }
296 %
297 %
298 \cs_generate_variant:Nn \tl_head:N { c }
299 \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.

```

300 \cs_new:Nn \UWMad_Stack_Pop:n {
301   \__UWMad_Stack_IfDefined:nT {#1} {
302     \tl_set:Nf \l_tmpa_tl {\tl_head:c {g__UWMad_Stack_#1}}
303     \tl_set:cf {g__UWMad_Stack_#1} {\tl_tail:c {g__UWMad_Stack_#1}}
304     \tl_use:N \l_tmpa_tl
305   }
306 }

```

---

**\UWMad\_Stack\_Walk:nn**

---

Iterate of the elements of a defined Stack in a FILO sense with supplied code.

```

307 \cs_new:Nn \UWMad_Stack_Walk:nn {
308   \tl_map_inline:cn {g__UWMad_Stack_#1} {#2}
309 }

```

### 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.

```
310 \cs_new:Nn \__UWMad_Queue_IfDefined:nT {
311     \__UWMad_IfDefined:nnnnT{g__UWMad_Queue_}{#1}{}{Queue}{#2}
312 }
313 \cs_new:Nn \__UWMad_Queue_IfUndefined:nT{
314     \__UWMad_IfUndefined:nnnnT{g__UWMad_Queue_}{#1}{}{Queue}{#2}
315 }
```

---

```
\UWMad_Queue_Define:n
```

---

Define a new Queue.

```
316 \cs_new:Nn \UWMad_Queue_Define:n {
317     \__UWMad_Queue_IfUndefined:nT {#1} {
318         \tl_new:c {g__UWMad_Queue_#1}
319     }
320 }
```

---

```
\UWMad_Queue_Clear:n
```

---

Clear but do not undefine a defined Queue.

```
321 \cs_new:Nn \UWMad_Queue_Clear:n {
322     \__UWMad_Queue_IfDefined:nT {#1} {
323         \tl_gclear:c {g__UWMad_Queue_#1}
324     }
325 }
```

---

```
\UWMad_Queue_Delete:n
```

---

Clear and undefine a defined Queue.

```
326 \cs_new:Nn \UWMad_Queue_Delete:n {
327     \__UWMad_Queue_IfDefined:nT {#1} {
328         \tl_gclear:c {g__UWMad_Queue_#1}
329         \cs_undefine:c {g__UWMad_Queue_#1}
330     }
331 }
```

---

**\UWMad\_Queue\_Pop:nn**


---

Push an item on to the start of a defined Queue.

```

332 \cs_new:Nn \UWMad_Queue_Push:nn {
333     \__UWMad_Queue_IfDefined:nT {#1} {
334         \tl_gput_left:cn {g__UWMad_Queue_#1} {{#2}}
335     }
336 }
337 %
338 %
339 \cs_generate_variant:Nn \tl_head:N { c }
340 \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.

```

341 \cs_new:Nn \UWMad_Queue_Pop:n {
342     \__UWMad_Queue_IfDefined:nT {#1} {
343         \tl_reverse:c {g__UWMad_Queue_#1}
344         \tl_set:Nf \l_tmpa_tl
345             {\tl_head:c {g__UWMad_Queue_#1}}
346         \tl_set:cf {g__UWMad_Queue_#1}
347             {\tl_tail:c {g__UWMad_Queue_#1}}
348         \tl_reverse:c {g__UWMad_Queue_#1}
349         \tl_use:N \l_tmpa_tl
350     }
351 }

```

---

**\UWMad\_Queue\_Walk:nn**


---

Iterate of the elements of a defined Queue in a FIFO sense with supplied code.

```

352 \cs_new:Nn \UWMad_Queue_Walk:nn {
353     \__UWMad_Queue_IfDefined:nT {#1} {
354         \group_begin:
355             \tl_reverse:c {g__UWMad_Queue_#1}
356             \tl_map_inline:cn {g__UWMad_Queue_#1} {#2}
357         \group_end:
358     }
359 }

```

---

`\UWMad_Queue_IfEmpty:nTF`

---

Execute true/false code depending on the emptiness of a defined Queue.

```

360 \cs_new:Nn \UWMad_Queue_IfEmpty:nTF {
361   \__UWMad_Queue_IfDefined:nT {#1} {
362     \tl_if_empty:cTF {g__UWMad_Queue_#1}{
363       #2
364     }{
365       #3
366     }
367   }
368 }
```

### 2.2.3 Deques

This set of commands is a simple system for creating and working with double-ended queues (dequeues, 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.

```

369 \cs_new:Nn \__UWMad_Deque_IfDefined:nT {
370   \__UWMad_IfDefined:nnnnT{g__UWMad_Deque_}{#1}{}{Deque}{#2}
371 }
372 \cs_new:Nn \__UWMad_Deque_IfUndefined:nT{
373   \__UWMad_IfUndefined:nnnnT{g__UWMad_Deque_}{#1}{}{Deque}{#2}
374 }
```



---

**\UWMad\_Deque\_Define:n**


---

Define a new Deque.

```

375 \cs_new:Nn \UWMad_Deque_Define:n {
376     \__UWMad_Deque_IfUndefined:nT {#1} {
377         \seq_new:c {g__UWMad_Deque_#1}
378     }
379 }
```

---

**\UWMad\_Deque\_Clear:n**


---

Clear but do not undefine a defined Deque.

```

380 \cs_new:Nn \UWMad_Deque_Clear:n {
381     \__UWMad_Deque_IfDefined:nT {#1} {
382         \seq_gclear:c {g__UWMad_Deque_#1}
383     }
384 }
```

---

**\UWMad\_Deque\_Delete:n**


---

Clear and undefine a defined Deque.

```

385 \cs_new:Nn \UWMad_Deque_Delete:n {
386     \__UWMad_Deque_IfDefined:nT {#1} {
387         \seq_gclear:c {g__UWMad_Deque_#1}
388         \cs_undefine:c {g__UWMad_Deque_#1}
389     }
390 }
```

---

\UWMad\_Deque\_PushLeft:nn  
 \UWMad\_Deque\_PushRight:nn

---

Push an element on to the left or right of a defined Deque.

```

391 \cs_new:Nn \UWMad_Deque_PushLeft:nn {
392   \__UWMad_Deque_IfDefined:nT {#1} {
393     \seq_gput_left:cn {g__UWMad_Deque_#1} {#2}
394   }
395 }
396 \cs_new:Nn \UWMad_Deque_PushRight:nn {
397   \__UWMad_Deque_IfDefined:nT {#1} {
398     \seq_gput_right:cn {g__UWMad_Deque_#1} {#2}
399   }
400 }
```

---

\UWMad\_Deque\_PopLeft:nn  
 \UWMad\_Deque\_PopRight:nn

---

Pop an element from the left or right of a defined Deque and place it into the input stream.

```

401 \cs_new:Nn \UWMad_Deque_PopLeft:n {
402   \__UWMad_Deque_IfDefined:nT {#1} {
403     \seq_gpop_left:cN {g__UWMad_Deque_#1} \l_tmpa_tl
404     \tl_use:N \l_tmpa_tl
405   }
406 }
407 \cs_new:Nn \UWMad_Deque_PopRight:n {
408   \__UWMad_Deque_IfDefined:nT {#1} {
409     \seq_gpop_right:cN {g__UWMad_Deque_#1} \l_tmpa_tl
410     \tl_use:N \l_tmpa_tl
411   }
412 }
```

---

```
\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.

```

413 \cs_new:Nn \UWMad_Deque_WalkLeftToRight:nn {
414     \__UWMad_Deque_IfDefined:nT {#1} {
415         \seq_map_inline:cn {g__UWMad_Deque_#1} {#2}
416     }
417 }
418 \cs_generate_variant:Nn \seq_reverse:N {c}
419 \cs_new:Nn \UWMad_Deque_WalkRightToLeft:nn {
420     \__UWMad_Deque_IfDefined:nT {#1} {
421         \group_begin:
422             \seq_reverse:c      {g__UWMad_Deque_#1}
423             \seq_map_inline:cn {g__UWMad_Deque_#1} {#2}
424         \group_end:
425     }
426 }
```

## 2.2.4 Hashes

This set of commands is a simple system for creating and working with hashes (more often called associative arrays or dictionaries, but erring on the side of usability, Ruby's jargon will be used). Hashes are a type of array that indexes values by (at least in  $\text{\LaTeX}$ ) 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 guaranteed to be the set order**. If walk order is needed to be guaranteed, 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  $\text{\LaTeX}$ 3 programming.

```

427 \cs_generate_variant:Nn \prop_gput:Nnn { c x n }
428 \cs_generate_variant:Nn \prop_if_in:NnTF { c x TF }
429 \cs_generate_variant:Nn \prop_if_in:NnTF { c f TF }
430 \cs_generate_variant:Nn \prop_get:Nn { c x }
431 \cs_generate_variant:Nn \prop_get:Nn { c f }
432 \cs_generate_variant:Nn \prop_get:NnNTF { c x N TF }
433 \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.

```

434 \cs_new:Nn \__UWMad_Hash_IfDefined:nT {
435     \__UWMad_IfDefined:nnnnT{g__UWMad_Hash_}{#1}{#2}{Hash}{#2}
436 }
437 \cs_new:Nn \__UWMad_Hash_IfUndefined:nT{
438     \__UWMad_IfUndefined:nnnnT{g__UWMad_Hash_}{#1}{#2}{Hash}{#2}
439 }

```

---

```
\UWMad_Hash_Define:n
```

---

Define a new Hash.

```

440 \cs_new:Nn \UWMad_Hash_Define:n {
441     \__UWMad_Hash_IfUndefined:nT {#1} {
442         \prop_new:c {g__UWMad_Hash_#1}
443     }
444 }

```

---

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

---

Set the value of a key of a defined Hash.

```

445 \cs_new:Nn \UWMad_Hash_Set:nnn {
446     \__UWMad_Hash_IfDefined:nT {#1} {
447         \prop_gput:cxn {g__UWMad_Hash_#1}{#2}{#3}
448     }
449 }

```

---

\UWMad\_Hash\_Get:nn


---

Get the value of a key of a defined Hash and place it into the input stream.

```

450 \cs_generate_variant:Nn \prop_get:cn {cf}
451 \cs_new:Nn \UWMad_Hash_Get:nn {
452     \__UWMad_Hash_IfDefined:nT {#1} {
453         \prop_get:cf {g__UWMad_Hash_#1}{#2}
454     }
455 }
```

---

\UWMad\_Hash\_Unset:nn


---

Undefine a key-value pair in a defined Hash.

```

456 \cs_new:Nn \UWMad_Hash_Unset:nn {
457     \__UWMad_Hash_IfDefined:nT {#1} {
458         \prop_gremove:cx {g__UWMad_Hash_#1} {#2}
459     }
460 }
```

---

\UWMad\_Hash\_IfKeySet:nnTF


---

Execute true/false code depending on if a key is set in a defined Hash.

```

461 \cs_generate_variant:Nn \tl_to_lowercase:n {f}
462 \cs_new:Nn \UWMad_Hash_IfKeySet:nnTF {
463     \__UWMad_Hash_IfDefined:nT {#1} {
464         \prop_if_in:cfTF {g__UWMad_Hash_#1} {\tl_trim_spaces:n{#2}} {
465             #3
466         }{
467             #4
468         }
469     }
470 }
```

---

`\UWMad_Hash_Walk:nn`


---

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

```

471 \cs_new:Nn \UWMad_Hash_Walk:nn {
472     \__UWMad_Hash_IfDefined:nT {#1} {
473         \prop_map_inline:cn {g__UWMad_Hash_#1} {#2}
474     }
475 }
```

---

`\UWMad_Hash_Delete:n`


---

Clear and undefine a defined Hash.

```

476 \cs_new:Nn \UWMad_Hash_Delete:n {
477     \__UWMad_Hash_IfDefined:nT {#1} {
478         \prop_gclear:c {g__UWMad_Hash_#1}
479         \cs_undefine:c {g__UWMad_Hash_#1}
480     }
481 }
```

## 2.3 User-Level Abstractions

The commands that follow are  $\text{\LaTeX 2}_{\epsilon}$ -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

---

<code>\IfCommandExists</code>	<code>\IfCommandExists{&lt;Command Name&gt;}{&lt;True&gt;}{&lt;False&gt;}</code>
<code>\IfCommandDoesNotExist</code>	<code>\IfCommandDoesNotExist{&lt;Command Name&gt;}{&lt;True&gt;}{&lt;False&gt;}</code>

---

This command pair is used instead of L<sup>A</sup>T<sub>E</sub>X's `\@ifundefined`. Since it is  $\varepsilon$ -T<sub>E</sub>X, this command will allow for a switch to `\@ifundefined` if problems arise from non- $\varepsilon$ -T<sub>E</sub>X users in the future.

```

482 \DeclareDocumentCommand \IfCommandExistsTF { m +m +m } {
483   \cs_if_exist:cTF {#1}{
484     #2
485   }{
486     #3
487   }
488 }
489 \DeclareDocumentCommand \IfCommandDoesNotExistTF { m +m +m } {
490   \cs_if_free:cTF {#1}{
491     #2
492   }{
493     #3
494   }
495 }
```

---

<code>\IfStringEmpty</code>	<code>\IfStringEmpty{&lt;String&gt;}{&lt;True&gt;}{&lt;False&gt;}</code>
-----------------------------	--

---

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

```

496 \cs_generate_variant:Nn \tl_if_blank:nTF {fTF}
497 \DeclareDocumentCommand \IfEmptyTF { m +m +m } {
498   \tl_if_blank:fTF {#1}{
499     #2
500   }{
501     #3
502   }
503 }
```

---

<code>\IfCommandEmpty</code>	<code>\IfCommandEmpty{&lt;Command&gt;}{&lt;True&gt;}{&lt;False&gt;}</code>
------------------------------	--

---

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

```

504 \DeclareDocumentCommand \IfCommandEmptyTF { m +m +m }{
505     \tl_if_blank:oTF{#1}{
506         #2
507     }{
508         #3
509     }
510 }
```

## 2.3.2 Command Creators

---

<code>\MakeCommand</code>	<code>\MakeCommand{&lt;Command Name&gt;}{&lt;Code&gt;}</code>
<code>\ReMakeCommand</code>	<code>\ReMakeCommand{&lt;Command Name&gt;}{&lt;Code&gt;}</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.

```

511 \DeclareDocumentCommand \MakeCommand { 0{ } m +m } {
512     \cs_if_free:cTF {#2} {
513         \cs_set:cpn {#2} #1 {#3}
514     }{
515         \msg_warning:nnnn
516             {UWMadThesis}{Programming/Defined}{#2}{command}
517     }
518 }
519 \DeclareDocumentCommand \ReMakeCommand { 0{ } m +m }{
520     \cs_if_exist:cTF {#2} {
521         \cs_set:cpn {#2} #1 {#3}
522     }{
523         \msg_error:nnnn
524             {UWMadThesis}{Programming/Undefined}{#2}{command}
525     }
526 }
```



---

**\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.

```

527 \DeclareDocumentCommand \MakeGlobalCommand { 0{} +m m } {
528     \cs_gset:cpn {#2} #1 {#3}
529 }
```

---

**\MakeCommandUndefined**    \MakeCommandUndefined{<Command Name>}

---

Globally undefines the command specified by {<Command Name>}.

```

530 \DeclareDocumentCommand \MakeCommandUndefined { m } {
531     \cs_undefine:c {#1}
532 }
```

---

**\CopyCommand**    \CopyCommand{<Command Name 1>}{<Command Name 2>}

---

Copies the definition 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.

```

533 \DeclareDocumentCommand \CopyCommand { m m } {
534     \cs_if_free:cTF {#1} {
535         \cs_if_free:cTF {#2} {
536             \cs_gset_eq:cc {#2}{#1}
537         }{
538             \msg_warning:nnnn
539                 {UWMadThesis}{Programming/Defined}{#2}{command}
540         }
541     }{
542         \msg_warning:nnnn
543             {UWMadThesis}{Programming/Defined}{#1}{command}
544     }
545 }
```

### 2.3.3 Types

---

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

---

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> version of the Boolean Type system above.

```

546 \DeclareDocumentCommand \CreateBoolean { m } {
547     \bool_new:c {g__UWMad_Programming_API_#1_bool}
548 }
549 \DeclareDocumentCommand \CreateBooleanTrue { m } {
550     \bool_new:c {g__UWMad_Programming_API_#1_bool}
551     \bool_gset_true:c {g__UWMad_Programming_API_#1_bool}
552 }
553 \DeclareDocumentCommand \CreateBooleanFalse { m } {
554     \bool_new:c {g__UWMad_Programming_API_#1_bool}
555 }
556 \DeclareDocumentCommand \SetBooleanTrue { m } {
557     \bool_gset_true:c {g__UWMad_Programming_API_#1_bool}
558 }
559 \DeclareDocumentCommand \SetBooleanFalse { m } {
560     \bool_gset_false:c {g__UWMad_Programming_API_#1_bool}
561 }
562 \DeclareDocumentCommand \IfBooleanTrueTF { m +m +m } {
563     \bool_if:cTF {g__UWMad_Programming_API_#1_bool} {
564         #2
565     } {
566         #3
567     }
568 }
569 \DeclareDocumentCommand \IfBooleanFalseTF { m +m +m } {
570     \bool_if:cTF {g__UWMad_Programming_API_#1_bool} {
571         #3
572     } {
573         #2
574     }
575 }
```

---

`\CreateLength`  
`\AddToLength`  
`\SetLength`  
`\ValueOfLength`  
`\IfLengthTF`

---

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> version of the Boolean Type system above.

```

576 \DeclareDocumentCommand \CreateLength { m m } {
577     \dim_new:c {g__UWMad_Programming_API_#1_dim}
578     \dim_gset:cn {g__UWMad_Programming_API_#1_dim} {#2}
579 }
580 \DeclareDocumentCommand \AddToLength { m m } {
581     \dim_gadd:cn {g__UWMad_Programming_API_#1_dim} {#2}
582 }
583 \DeclareDocumentCommand \SetLength { m m } {
584     \dim_gset:cn {g__UWMad_Programming_API_#1_dim} {#2}
585 }
586 \DeclareDocumentCommand \ValueOfLength { m } {
587     \dim_use:c {g__UWMad_Programming_API_#1_dim}
588 }
589 \DeclareDocumentCommand \IfLengthTF { m m m +m +m } {
590     \dim_compare:nNnTF {#1} #2 {#3} {
591         #4
592     } {
593         #5
594     }
595 }
```

---

 $\backslash$ CreateCounter

 $\backslash$ AddToCounter

 $\backslash$ StepCounter

 $\backslash$ SetCounter

 $\backslash$ ValueOfCounter

 $\backslash$ IfCounterTF

 $\backslash$ CounterToArabic

 $\backslash$ CounterToALPHA

 $\backslash$ CounterToAlpha

 $\backslash$ CounterToROMAN

 $\backslash$ CounterToRoman

---

L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> version of the Counter Type system above.

```

596 \DeclareDocumentCommand \CreateCounter { m m } {
597     \int_new:c {g__UWMad_Programming_API_#1_int}
598     \int_gset:cn {g__UWMad_Programming_API_#1_int} {#2}
599 }
600 \DeclareDocumentCommand \AddToCounter { m m } {
601     \int_gadd:cn {g__UWMad_Programming_API_#1_int} {#2}
602 }
603 \DeclareDocumentCommand \StepCounter { m m } {
604     \int_gincr:cn {g__UWMad_Programming_API_#1_int} {#2}
605 }
606 \DeclareDocumentCommand \SetCounter { m m } {
607     \int_gset:cn {g__UWMad_Programming_API_#1_int} {#2}
608 }
609 \DeclareDocumentCommand \ValueOfCounter { m m } {
610     \int_use:c {g__UWMad_Programming_API_#1_int}
611 }
612 \DeclareDocumentCommand \IfCounterTF { m m m +m +m } {
613     \int_compare:nNnTF {#1} {#2} {#3} {
614         #4
615     } {
616         #5
617     }
618 }
619 \DeclareDocumentCommand \CounterToArabic { m } {
620     \int_to_arabic:c {g__UWMad_Programming_API_#1_int}
621 }
622 \DeclareDocumentCommand \CounterToALPHA { m } {
623     \int_to_Alph:c {g__UWMad_Programming_API_#1_int}
624 }
625 \DeclareDocumentCommand \CounterToAlpha { m } {
626     \int_to_alph:c {g__UWMad_Programming_API_#1_int}
627 }
628 \DeclareDocumentCommand \CounterToROMAN { m } {
629     \int_to_Roman:c {g__UWMad_Programming_API_#1_int}
630 }
631 \DeclareDocumentCommand \CounterToRoman { m } {
632     \int_to_roman:c {g__UWMad_Programming_API_#1_int}
633 }

```

## Module 3

# Layout And Styles

### 3.1 Float Styles

Make equation references of the form (#).

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

Default table style

```
635 \captionsetup [table] {
636     format          = hang                ,
637     labelsep        = colon                ,
638     justification    = justified           ,
639     labelfont        = sc                  ,
640     textfont         = sl                  ,
641     font             = {normal,stretch=1.1},
642     width            = 0.9\textwidth      ,
643     position         = above               ,
644     skip             = 0.50em
645 }
```

Default figure style.

```
646 \captionsetup [figure] {
647     format          = hang                ,
648     labelsep        = colon                ,
649     justification    = justified           ,
650     labelfont        = sl                  ,
651     textfont         = sl                  ,
652     font             = {normal,stretch=1.1},
653     width            = 0.9\textwidth      ,
```

```

654     position      = above           ,
655     skip           = 0.5em
656 }

```

## 3.2 Links

Define a darker green than |green|.

```

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

```

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

```

658 \tl_new:N \l_UWMad_LayoutStyle_Color_Link_tl
659 \tl_set:Nn \l_UWMad_LayoutStyle_Color_Link_tl {blue}
660 \tl_new:N \l_UWMad_LayoutStyle_Color_Cite_tl
661 \tl_set:Nn \l_UWMad_LayoutStyle_Color_Cite_tl {UWMadGreen}
662 \tl_new:N \l_UWMad_LayoutStyle_Color_URL_tl
663 \tl_set:Nn \l_UWMad_LayoutStyle_Color_URL_tl {violet}

```

Define a new color and hyperlink defaults

```

664 \hypersetup {
665     colorlinks      = true,
666     linkcolor       = \l_UWMad_LayoutStyle_Color_Link_tl,
667     citecolor       = \l_UWMad_LayoutStyle_Color_Cite_tl,
668     urlcolor        = \l_UWMad_LayoutStyle_Color_URL_tl,
669     pdfdisplaydoctitle = true,
670     pdfview         = {FitH},
671     pdfstartview    = {FitH},
672     pdfpagelayout    = OneColumn,
673     plainpages      = false,
674     hypertexnames    = true,
675     bookmarksopenlevel = 1,
676     bookmarksopen    = true,
677     unicode          = true
678 }

```

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

```

679 \cs_new:Nn \UWMad_LayoutStyle_ResetLinkColors: {
680     \hypersetup {
681         colorlinks = true,
682         linkcolor   = \l_UWMad_LayoutStyle_Color_Link_tl,
683         citecolor   = \l_UWMad_LayoutStyle_Color_Cite_tl,
684         urlcolor    = \l_UWMad_LayoutStyle_Color_URL_tl
685     }
686 }

```

Define user interfaces to setting link colors.

```

687 \DeclareDocumentCommand \MakeLinksTheseColors { m m m } {
688     \tl_set:Nn \l_UWMad_LayoutStyle_Color_Link_tl {#1}
689     \tl_set:Nn \l_UWMad_LayoutStyle_Color_Cite_tl {#2}
690     \tl_set:Nn \l_UWMad_LayoutStyle_Color_URL_tl  {#3}
691     \UWMad_LayoutStyle_ResetLinkColors:
692 }
693 \DeclareDocumentCommand \MakeLinksThisColor { m } {
694     \tl_set:Nn \l_UWMad_LayoutStyle_Color_Link_tl {#1}
695     \tl_set:Nn \l_UWMad_LayoutStyle_Color_Cite_tl {#1}
696     \tl_set:Nn \l_UWMad_LayoutStyle_Color_URL_tl  {#1}
697     \UWMad_LayoutStyle_ResetLinkColors:
698 }

```

Define user interfaces to specialized color commands.

```

699 \keys_define:nn { UWMadThesis / LayoutStyle } {
700     link-color .code:n = {
701         \hypersetup {
702             colorlinks = true,
703             linkcolor   = #1,
704         }
705     },
706     cite-color .code:n = {
707         \hypersetup {
708             colorlinks = true,
709             citecolor   = #1,
710         }
711     },
712     url-color .code:n = {
713         \hypersetup {
714             colorlinks = true,
715             urlcolor    = #1,
716         }
717     },

```

```

718 link-color .default:n = blue,
719 cite-color .default:n = UWMadGreen,
720 url-color .default:n = blue,
721 all-link-color .code:n = {
722     \hypersetup {
723         colorlinks = true,
724         linkcolor   = #1,
725         citecolor   = #1,
726         urlcolor    = #1,
727     }
728 },
729 make-links-blue .meta:n = {
730     all-link-color = blue
731 },
732 make-links-black .meta:n = {
733     all-link-color = black
734 },
735 make-links-red .meta:n = {
736     all-link-color = red
737 }
738 }
739 \keys_set:nn { UWMadThesis / LayoutStyle } {
740     link-color,
741     cite-color,
742     url-color
743 }
744 \DeclareDocumentCommand \MakeLinksBlack { } {
745     \keys_set:nn { UWMadThesis / LayoutStyle } {
746         make-links-black = true
747     }
748 }
749 \DeclareDocumentCommand \MakeLinksBlue { } {
750     \keys_set:nn { UWMadThesis / LayoutStyle } {
751         make-links-blue = true
752     }
753 }
754 \DeclareDocumentCommand \MakeLinksRed { } {
755     \keys_set:nn { UWMadThesis / LayoutStyle } {
756         make-links-red = true
757     }
758 }

```



### 3.3 Page Layout

One inch margins and letter (paper size) are set.

```

759 \geometry{
760     includehead = true,
761     margin       = 1.0in,
762     paper        = letterpaper,
763 }
```

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

```

764 \doublespacing
765 \UWMad_Hook_Prepended:Nn \singlespacing {
766     \_UWMad_FrontMatter_StyleWarning:n {
767         University~guidelines~require~double~spacing.~
768         If~this~is~for~temporary~use,~please~use~the~spacing~environment.
769     }
770 }
771 \UWMad_Hook_Prepended:Nn \onehalfspacing {
772     \_UWMad_FrontMatter_StyleWarning:n {
773         University~guidelines~require~double~spacing.~
774         If~this~is~for~temporary~use,~please~use~the~spacing~environment.
775     }
776 }
```

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

```

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

Define user interface for defining different indentation styles.

```

779 \keys_define:nn { UWMadThesis / LayoutStyle } {
780     indent-length .code:n = {
781         \setlength{\parindent}{#1}
782     },
783     skip-length .code:n = {
784         \setlength{\parskip}{#1}
785     },
786     indent-length .default:n = 0pt,
787     skip-length .default:n    = 1em,
```

```

788 paragraph-style .choice:,
789 paragraph-style / indent .code:n = {
790     \setlength{\parindent}{1.50em}
791     \setlength{\parskip}{0pt}
792 },
793 paragraph-style / pad .code:n = {
794     \setlength{\parindent}{0pt}
795     \setlength{\parskip}{1em}
796 }
797 }
798 \keys_set:nn { UWMadThesis / LayoutStyle } {
799     paragraph-style = pad
800 }

```

## 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.

```

801 \UWMad_Hook_Prepended:Nn \@chapter {
802   \thispagestyle{myheadings}
803   \int_compare:nNnTF {\value{chapter}} = {0} {
804     \pagenumbering{arabic}
805   } { }
806 }
807 \UWMad_Hook_Prepended:Nn \@schapter {
808   \thispagestyle{myheadings}
809 }

```

## 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.

```

810 \int_new:N \g__UWMad_Appendix_Counter_int
811 \int_set:Nn \g__UWMad_Appendix_Counter_int {0}

```

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

```

812 \cs_new:Nn \__UWMad_Appendix_Initialize:{
813   \par
814   \setcounter{section}{0}

```

```

815 \cs_gset_eq:NN \@chapapp \appendixname
816 \cs_gset:Npn \thechapter {
817   \int_to_Alph:n {
818     \g__UWMad_Appendix_Counter_int
819   }
820 }
821 }

```

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

```

822 \cs_undefine:N \appendix
823 \DeclareDocumentCommand \appendix { s o m } {
824
825   \int_compare:nNnTF {\g__UWMad_Appendix_Counter_int} = {0} {
826     \__UWMad_Appendix_Initialize:
827   } { }
828   \int_gincr:N \g__UWMad_Appendix_Counter_int
829
830   \IfBooleanTF { #1 } {
831     \chapter*{#3}
832   } {
833     \IfNoValueTF { #2 } {
834       \chapter[#3]{#3}
835     } {
836       \chapter[#2]{#3}
837     }
838   }
839 }

```

## 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.

```

840 %
841 \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.

```

842 \cs_new:Nn \__UWMad_FrontMatter_Register:nn {
843
844     \int_compare:nNnTF {\g__UWMad_FrontMatter_Counter_int} = {0} {
845         \pagenumbering{roman}
846     } { }
847
848     \int_gincr:N \g__UWMad_FrontMatter_Counter_int
849     \addcontentsline
850         {toc}
851         {#1}
852         {#2}
853 }
```

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

```

854 \tl_new:N \g__UWMad_FrontMatter_Title_Dedications_tl
855 \tl_new:N \g__UWMad_FrontMatter_Title_Acknowledgments_tl
856 \tl_new:N \g__UWMad_FrontMatter_Title_Abstract_tl
857 \tl_new:N \g__UWMad_FrontMatter_Title_UMIAbstract_tl
858 \tl_new:N \g__UWMad_FrontMatter_Title_Preface_tl
859 %
860 \tl_gset:Nn \g__UWMad_FrontMatter_Title_Dedications_tl
861     {Dedications}
862 \tl_gset:Nn \g__UWMad_FrontMatter_Title_Acknowledgments_tl
863     {Acknowledgments}
864 \tl_gset:Nn \g__UWMad_FrontMatter_Title_Abstract_tl
865     {Abstract}
866 \tl_gset:Nn \g__UWMad_FrontMatter_Title_UMIAbstract_tl
867     {Abstract}
868 \tl_gset:Nn \g__UWMad_FrontMatter_Title_Preface_tl
869     {Preface}
```

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

```

870 \cs_undefine:N \abstract
871 \cs_undefine:N \endabstract
872
873 \DeclareDocumentCommand \FrontMatterSetSection { m m } {
```

```

874
875 \tl_set_eq:Nc
876 \l_tmpa_tl
877 {g__UWMad_FrontMatter_Title_#2_tl}
878
879 \IfNoValueTF { #1 } { } {
880 \IfEmptyTF { #1 } { } {
881 \tl_set:Nn \l_tmpa_tl {#1}
882 }
883 }
884
885 \chapter*{\l_tmpa_tl}
886 \__UWMad_FrontMatter_Register:nn {chapter} {
887 \l_tmpa_tl
888 }
889
890 }
891 \DeclareDocumentCommand \dedications { g } {
892 \FrontMatterSetSection{#1}{Dedications}
893 }
894 \DeclareDocumentCommand \acknowledgments { g } {
895 \FrontMatterSetSection{#1}{Acknowledgments}
896 }
897 \DeclareDocumentCommand \abstract { g } {
898 \FrontMatterSetSection{#1}{Abstract}
899 }
900 \DeclareDocumentCommand \umiabstract { g } {
901 \FrontMatterSetSection{#1}{Abstract}
902 }
903 \DeclareDocumentCommand \preface { g } {
904 \FrontMatterSetSection{#1}{Preface}
905 }

```

## 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.

```

906 \cs_gset_eq:NN \TableOfContentsDefault \tableofcontents
907 \cs_gset_eq:NN \ListOfTablesDefault \listoftables
908 \cs_gset_eq:NN \ListOfFiguresDefault \listoffigures
909 \cs_undefine:N \tableofcontents
910 \cs_undefine:N \listoftables
911 \cs_undefine:N \listoffigures

```

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

```

912 \tl_new:N \g__UWMad_TOC_Name_TOC_tl
913 \tl_new:N \g__UWMad_TOC_Name_LOT_tl
914 \tl_new:N \g__UWMad_TOC_Name_LOF_tl
915 \tl_gset:Nn \g__UWMad_TOC_Name_TOC_tl {Table~of~Contents}
916 \tl_gset:Nn \g__UWMad_TOC_Name_LOT_tl {List~of~Tables}
917 \tl_gset:Nn \g__UWMad_TOC_Name_LOF_tl {List~of~Figures}

```

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

```

918 \DeclareDocumentCommand \tableofcontents { } {
919
920   \tl_gset_eq:NN \contentsname \g__UWMad_TOC_Name_TOC_tl
921
922   \group_begin:
923     \setstretch{1.05}
924     \phantomsection
925     \ExplSyntaxOff
926     \TableOfContentsDefault
927     \ExplSyntaxOn
928     \__UWMad_FrontMatter_Register:nn
929       {chapter}
930       {\contentsname}
931   \clearpage
932   \group_end:
933 }
934 \DeclareDocumentCommand \listoftables { } {
935
936   \cs_set_eq:NN \listtablename \g__UWMad_TOC_Name_LOT_tl
937
938   \group_begin:
939     \setstretch{1.05}
940     \ExplSyntaxOff
941     \ListOfTablesDefault

```

```

942         \ExplSyntaxOn
943         \__UWMad_FrontMatter_Register:nn
944             {chapter}
945             {\listtablename}
946         \clearpage
947     \group_end:
948 }
949 \DeclareDocumentCommand \listoffigures { } {
950
951     \cs_set_eq:NN \listfigurename \g__UWMad_TOC_Name_LOF_tl
952
953     \group_begin:
954         \setstretch{1.05}
955         \ExplSyntaxOff
956         \ListOfFiguresDefault
957         \ExplSyntaxOn
958         \__UWMad_FrontMatter_Register:nn
959             {chapter}
960             {\listfigurename}
961     \clearpage
962     \group_end:
963 }

```

Camel-cased aliases.

```

964 \cs_set_eq:NN \TableOfContents \tableofcontents
965 \cs_set_eq:NN \ListOfTables     \listoftables
966 \cs_set_eq:NN \ListOfFigures   \listoffigures

```

User-level commands to change the default names.

```

967 \DeclareDocumentCommand \TableOfContentsName { m } {
968     \tl_gset:Nn \g__UWMad_TOC_Name_TOC_tl {#1}
969 }
970 \DeclareDocumentCommand \ListOfTablesName { m } {
971     \tl_gset:Nn \g__UWMad_TOC_Name_LOT_tl {#1}
972 }
973 \DeclareDocumentCommand \ListOfFiguresName { m } {
974     \tl_gset:Nn \g__UWMad_TOC_Name_LOF_tl {#1}
975 }

```



## 4.4 Section-Level Commands

These commands are used internally when needing to check if a user-supplied `section` is a  $\text{\LaTeX} 2_{\epsilon}$ -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.

```

976 \tl_const:Nn \c__UWMad_SectionsLevel_part_tl      {-1}
977 \tl_const:Nn \c__UWMad_SectionsLevel_chapter_tl   {0}
978 \tl_const:Nn \c__UWMad_SectionsLevel_section_tl   {1}
979 \tl_const:Nn \c__UWMad_SectionsLevel_subsection_tl {2}
980 \tl_const:Nn \c__UWMad_SectionsLevel_subsubsection_tl {3}
981 \tl_const:Nn \c__UWMad_SectionsLevel_paragraph_tl  {4}
982 \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.

```

983 \msg_new:nnn { UWMadThesis } { Sectioning / UndefinedSection } {
984     Undefined~section~'#1'~used.
985 }
986 \cs_new:Nn \UWMad_IfSectionExists:nT {
987     \tl_if_exist:cTF {c__UWMad_SectionsLevel_ #1 _tl} {
988         #2
989     } {
990         \msg_error:nnn
991             { UWMadThesis }
992             { Sectioning / UndefinedSection }
993             {#1}
994     }
995 }

```

Variables that map a level number to a section.

```

996 \tl_const:cn {c__UWMad_LevelsSection_-1_tl} {part}
997 \tl_const:cn {c__UWMad_LevelsSection_ 0_tl} {chapter}
998 \tl_const:cn {c__UWMad_LevelsSection_ 1_tl} {section}
999 \tl_const:cn {c__UWMad_LevelsSection_ 2_tl} {subsection}
1000 \tl_const:cn {c__UWMad_LevelsSection_ 3_tl} {subsubsection}
1001 \tl_const:cn {c__UWMad_LevelsSection_ 4_tl} {paragraph}
1002 \tl_const:cn {c__UWMad_LevelsSection_ 5_tl} {subparagraph}

```

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

```

1003 \tl_const:Nn \c__UWMad_NextSection_part_tl      {chapter}
1004 \tl_const:Nn \c__UWMad_NextSection_chapter_tl   {section}
1005 \tl_const:Nn \c__UWMad_NextSection_section_tl    {subsection}
1006 \tl_const:Nn \c__UWMad_NextSection_subsection_tl {subsubsection}
1007 \tl_const:Nn \c__UWMad_NextSection_subsubsection_tl {paragraph}
1008 \tl_const:Nn \c__UWMad_NextSection_paragraph_tl  {subparagraph}

```

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

```

1009 \tl_const:Nn \c__UWMad_PreviousSection_chapter_tl {part}
1010 \tl_const:Nn \c__UWMad_PreviousSection_section_tl {chapter}
1011 \tl_const:Nn \c__UWMad_PreviousSection_subsection_tl {section}
1012 \tl_const:Nn \c__UWMad_PreviousSection_subsubsection_tl {subsection}
1013 \tl_const:Nn \c__UWMad_PreviousSection_paragraph_tl {subsubsection}
1014 \tl_const:Nn \c__UWMad_PreviousSection_subparagraph_tl {paragraph}

```

Given a section, acquire its level number.

```

1015 \cs_new:Nn \UWMad_SectionToLevel:n {
1016     \UWMad_IfSectionExists:nT {#1} {
1017         \tl_use:c {c__UWMad_SectionsLevel_ #1 _tl}
1018     }
1019 }

```

Given a level number, acquire its section.

```

1020 \cs_new:Nn \UWMad_LevelToSection:n {
1021     \UWMad_IfSectionExists:nT {#1} {
1022         \tl_use:c {c__UWMad_LevelsSection_ #1 _tl}
1023     }
1024 }

```

Given a section, acquire its next lower one.

```

1025 \cs_new:Nn \UWMad_NextSection:n {
1026     \UWMad_IfSectionExists:nT {#1} {
1027         \tl_use:c {c__UWMad_NextSection_ #1 _tl}
1028     }
1029 }

```

Given a section, acquire its next higher one.

```
1030 \cs_new:Nn \UWMad_PreviousSection:n {  
1031     \UWMad_IfSectionExists:nT {#1} {  
1032         \tl_use:c {c__UWMad_PreviousSection_ #1 _tl}  
1033     }  
1034 }
```

## Module 5

### Math

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

```

1035 \tex_everydisplay:D \exp_after:wN {
1036     \tex_the:D \tex_everydisplay:D
1037     \cs_set_eq:NN \frac \dfrac
1038 }

```

#### 5.1 Derivative Commands

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

```

1039 \tl_new:N \g_UWMad_Math_derivSymbol_tl
1040 \tl_gset:Nn \g_UWMad_Math_derivSymbol_tl {\mathrm{d}}
1041 \tl_new:N \g_UWMad_Math_pderivSymbol_tl
1042 \tl_gset:Nn \g_UWMad_Math_pderivSymbol_tl {\partial}
1043 \tl_new:N \g_UWMad_Math_tderivSymbol_tl
1044 \tl_gset:Nn \g_UWMad_Math_tderivSymbol_tl {\mathrm{D}}
1045 \tl_new:N \g_UWMad_Math_DelimiterDefaultLeft_tl
1046 \tl_gset:Nn \g_UWMad_Math_DelimiterDefaultLeft_tl {[ ]
1047 \tl_new:N \g_UWMad_Math_DelimiterDefaultRight_tl
1048 \tl_gset:Nn \g_UWMad_Math_DelimiterDefaultRight_tl {[ ]
1049 \tl_new:N \l_UWMad_Math_DelimiterLeft_tl
1050 \tl_new:N \l_UWMad_Math_DelimiterRight_tl

```

Define the user interface accessors.

```

1051 \DeclareDocumentCommand \derivSymbol { } {
1052     \g_UWMad_Math_derivSymbol_tl
1053 }
1054 \DeclareDocumentCommand \pderivSymbol { } {

```

```

1055     \g_UWMad_Math_pderivSymbol_tl
1056   }
1057   \DeclareDocumentCommand \tderivSymbol { } {
1058     \g_UWMad_Math_tderivSymbol_tl
1059   }

```

Define the user interface local mutators.

```

1060   \DeclareDocumentCommand \derivSymbolChange { m } {
1061     \tl_set:Nn \g_UWMad_Math_derivSymbol_tl {#1}
1062   }
1063   \DeclareDocumentCommand \pderivSymbolChange { m } {
1064     \tl_set:Nn \g_UWMad_Math_pderivSymbol_tl {#1}
1065   }
1066   \DeclareDocumentCommand \tderivSymbolChange { m } {
1067     \tl_set:Nn \g_UWMad_Math_tderivSymbol_tl {#1}
1068   }

```

Define the user interface global mutators.

```

1069   \DeclareDocumentCommand \derivSymbolChangeDefault { m } {
1070     \tl_gset:Nn \g_UWMad_Math_derivSymbol_tl {#1}
1071   }
1072   \DeclareDocumentCommand \pderivSymbolChangeDefault { m } {
1073     \tl_gset:Nn \g_UWMad_Math_pderivSymbol_tl {#1}
1074   }
1075   \DeclareDocumentCommand \tderivSymbolChangeDefault { m } {
1076     \tl_gset:Nn \g_UWMad_Math_tderivSymbol_tl {#1}
1077   }

```

Define the \left and \right delimiter global mutators.

```

1078   \DeclareDocumentCommand \DelimiterChangeDefault { m m } {
1079     \tl_gset:Nn \g_UWMad_Math_DelimiterDefaultLeft_tl {#1}
1080     \tl_gset:Nn \g_UWMad_Math_DelimiterDefaultRight_tl {#2}
1081   }

```

Define the generic regular and big derivative functions.

```

1082   \DeclareDocumentCommand \DerivativeGeneral { +m +m m m } {
1083     \frac{ #4^{#3} #1 }
1084           { #4      #2^{#3} }
1085   }
1086   \DeclareDocumentCommand \DerivativeGeneralBig { +m +m m m m m } {

```

```

1087
1088 \IfNoValueTF {#5} {
1089     \tl_set_eq:NN
1090         \l_UWMad_Math_DelimiterLeft_tl
1091         \g_UWMad_Math_DelimiterDefaultLeft_tl
1092 } {
1093     \tl_set:Nn \l_UWMad_Math_DelimiterLeft_tl {#5}
1094 }
1095
1096 \IfNoValueTF {#6} {
1097     \tl_set_eq:NN
1098         \l_UWMad_Math_DelimiterRight_tl
1099         \g_UWMad_Math_DelimiterDefaultRight_tl
1100 } {
1101     \tl_set:Nn \l_UWMad_Math_DelimiterRight_tl {#6}
1102 }
1103
1104 \frac{ #4^{#3}      }
1105       { #4 #2^{#3} }
1106 \!\!
1107 \left\l_UWMad_Math_DelimiterLeft_tl
1108     #1
1109 \right\l_UWMad_Math_DelimiterRight_tl
1110 }

```

Define the three supported derivative types' small forms.

```

1111 \DeclareDocumentCommand \deriv { +m +m G{} } {
1112     \DerivativeGeneral
1113     {#1}{#2}{#3}{\derivSymbol}
1114 }
1115 \DeclareDocumentCommand \pderiv { +m +m G{} } {
1116     \DerivativeGeneral
1117     {#1}{#2}{#3}{\pderivSymbol}
1118 }
1119 \DeclareDocumentCommand \tderiv { +m +m G{} } {
1120     \DerivativeGeneral
1121     {#1}{#2}{#3}{\tderivSymbol}
1122 }

```

Define the three supported derivative types' big forms.

```

1123 \DeclareDocumentCommand \derivbig { o +m o +m G{} } {
1124     \DerivativeGeneralBig
1125     {#2}{#4}{#5}{\derivSymbol}{#1}{#3}

```

```

1126 }
1127 \DeclareDocumentCommand \pderivbig { o +m o +m G{} } {
1128   \DerivativeGeneralBig
1129     {#2}{#4}{#5}{\pderivSymbol}{#1}{#3}
1130 }
1131 \DeclareDocumentCommand \tderivbig { o +m o +m G{} } {
1132   \DerivativeGeneralBig
1133     {#2}{#4}{#5}{\tderivSymbol}{#1}{#3}
1134 }

```

## 5.2 Operators and Functions

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

```

1135 \DeclareMathOperator*\Sup      {Sup}
1136 \DeclareMathOperator*\Inf      {Inf}
1137 \DeclareMathOperator*\Lim      {Lim}
1138 \DeclareMathOperator*\Min      {Min}
1139 \DeclareMathOperator*\Max      {Max}
1140 \DeclareMathOperator*\ArgMin   {ArgMin}
1141 \DeclareMathOperator*\ArgMax   {ArgMax}
1142 \DeclareMathOperator*\Abs      {Abs}
1143 \DeclareMathOperator*\Ln       {Ln}
1144 \DeclareMathOperator*\Log      {Log}
1145 \DeclareMathOperator*\Exp      {Exp}
1146 \DeclareMathOperator*\Cos      {Cos}
1147 \DeclareMathOperator*\Sin      {Sin}
1148 \DeclareMathOperator*\Tan      {Tan}
1149 \DeclareMathOperator*\Sec      {Sec}
1150 \DeclareMathOperator*\Csc      {Csc}
1151 \DeclareMathOperator*\Cot      {Cot}
1152 \DeclareMathOperator*\Cosh     {Cosh}
1153 \DeclareMathOperator*\Sinh     {Sinh}
1154 \DeclareMathOperator*\Tanh     {Tanh}
1155 \DeclareMathOperator*\Sech     {Sech}
1156 \DeclareMathOperator*\Csch     {Csch}
1157 \DeclareMathOperator*\Coth     {Coth}
1158 \DeclareMathOperator*\ArcCos   {ArcCos}
1159 \DeclareMathOperator*\ArcSin   {ArcSin}
1160 \DeclareMathOperator*\ArcTan   {ArcTan}
1161 \DeclareMathOperator*\ArcSec   {ArcSec}

```

```

1162 \DeclareMathOperator{\ArcCsc}  {\ArcCsc}
1163 \DeclareMathOperator{\ArcCot}   {\ArcCot}
1164 \DeclareMathOperator{\ArcCosh}  {\ArcCosh}
1165 \DeclareMathOperator{\ArcSinh}  {\ArcSinh}
1166 \DeclareMathOperator{\ArcTanh}  {\ArcTanh}
1167 \DeclareMathOperator{\ArcSech}  {\ArcSech}
1168 \DeclareMathOperator{\ArcCsch}  {\ArcCsch}
1169 \DeclareMathOperator{\ArcCoth}  {\ArcCoth}

```

## 5.3 Miscellaneous Functions

Define the root function that has a tail.

```

1170 \cs_new:Nn \UWMad_Math_RootWithTail:nn {
1171
1172   \hbox_set:Nn \l_tmpa_box {
1173     $
1174     \mathchoice
1175       {\root #1 \of {#2\:\!}}
1176       {\root #1 \of {#2\:\!}}
1177       {\root #1 \of {#2\:\!}}
1178       {\root #1 \of {#2\:\!}}
1179     $
1180   }
1181   %
1182   \dim_set:Nn \l_tmpa_dim {\box_ht:N \l_tmpa_box}
1183   \dim_set:Nn \l_tmpb_dim {0.8\l_tmpa_dim}
1184   %
1185   \hbox_set:Nn \l_tmpb_box {
1186     \tex_vrule:D height \l_tmpa_dim depth -\l_tmpb_dim
1187   }
1188   %
1189   \box_use:N \l_tmpa_box
1190   \box_move_down:nn {0.40pt}{\box_use:N \l_tmpb_box}
1191 }
1192 \DeclareDocumentCommand \Sqrt { O{} m } {
1193   \UWMad_Math_RootWithTail:nn{#1}{#2}
1194 }

```

User interface math mode check.



```

1195 \DeclareExpandableDocumentCommand \IfMathModeTF { +m +m } {
1196     \mode_if_math:TF {
1197         #1
1198     }{
1199         $#2$
1200     }
1201 }

```

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

```

1202 \cs_gset_eq:NN \supsipa \sups
1203 \cs_undefine:N \sups

```

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

```

1204 \ExplSyntaxOff
1205 \DeclareDocumentCommand \subs { 0{} +m } {%
1206     \IfMathModeTF{%
1207         _{\!\!\!:#1\text{\scriptsize #2}}}%
1208     }{%
1209         _{\!#1\text{\scriptsize #2}}}%
1210     }%
1211 }%
1212 \DeclareDocumentCommand \sups { 0{} +m } {%
1213     \IfMathModeTF{%
1214         ^{#1\text{\scriptsize #2}}}%
1215     }{%
1216         ^{#1\text{\scriptsize #2}}}%
1217     }%
1218 }%
1219 \DeclareDocumentCommand \subsup { 0{} +m 0{} +m } {%
1220     \IfMathModeTF{%
1221         _{#1\text{\scriptsize #2}}^{!\!\!:#3\text{\scriptsize #4}}}%
1222     }{%
1223         _{#1\text{\scriptsize #2}}^{!\!\!:#3\text{\scriptsize #4}}}%
1224     }%
1225 }%
1226 \ExplSyntaxOn
1227 \cs_gset_eq:NN \subsup \subsup

```

The one-over functions discussed in the manual.

```

1228 \DeclareDocumentCommand \OneOver { +m } {
1229     \frac{1}{#1}

```

```

1230 }
1231 \DeclareDocumentCommand \oneo { +m } {
1232     \OneOver{#1}
1233 }

```

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

```

1234 \DeclareDocumentCommand \dd { m } {
1235     \mathrm{d}{#1}
1236 }

```

The prime commands discussed in the manual.

```

1237 \DeclareDocumentCommand \dprime { } {
1238     {\prime\prime}
1239 }
1240 \DeclareDocumentCommand \tprime { } {
1241     {\prime\prime\prime}
1242 }

```

Two commands that were necessary for proper typesetting.

```

1243 \DeclareDocumentCommand \LessThan { } {<}
1244 \DeclareDocumentCommand \GreaterThan { } {>}
1245 %

```

## Module 6

# ListOf

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

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

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

ListOf variable declarations for section levels.

```
1246 \tl_new:N \l__UWMad_ListOf_Section_Main_tl
1247 \tl_new:N \l__UWMad_ListOf_Section_Group_tl
1248 \tl_new:N \l__UWMad_ListOf_Section_Subgroup_tl
```

Boolean declarations for numbering and Table of Contents-inclusions.

```
1249 \bool_new:N \l__UWMad_ListOf_MakeNumbered_Main_bool
1250 \bool_new:N \l__UWMad_ListOf_MakeNumbered_Group_bool
1251 \bool_new:N \l__UWMad_ListOf_MakeNumbered_Subgroup_bool
1252 \bool_new:N \l__UWMad_ListOf_IncludeInTOC_Main_bool
1253 \bool_new:N \l__UWMad_ListOf_IncludeInTOC_Group_bool
1254 \bool_new:N \l__UWMad_ListOf_IncludeInTOC_Subgroup_bool
```

Entry queue and and Hook hash initialization

```

1255 \UWMad_Queue_Define:n {l__ListOf_EntryQueue}
1256 \UWMad_Hash_Define:n {l__ListOf_Hook}
1257 \cs_new:Nn \__UWMad_Listof_SetHooks_Blank: {
1258     \UWMad_Hash_Set:nnn {l__ListOf_Hook}{PreTitle-Main} {}
1259     \UWMad_Hash_Set:nnn {l__ListOf_Hook}{PostTitle-Main} {}
1260     \UWMad_Hash_Set:nnn {l__ListOf_Hook}{PreTitle-Group} {}
1261     \UWMad_Hash_Set:nnn {l__ListOf_Hook}{PostTitle-Group} {}
1262     \UWMad_Hash_Set:nnn {l__ListOf_Hook}{PreTitle-Subgroup} {}
1263     \UWMad_Hash_Set:nnn {l__ListOf_Hook}{PostTitle-Subgroup} {}
1264     \UWMad_Hash_Set:nnn {l__ListOf_Hook}{PrePush} {}
1265     \UWMad_Hash_Set:nnn {l__ListOf_Hook}{PostPush} {}
1266     \UWMad_Hash_Set:nnn {l__ListOf_Hook}{PrePrint} {}
1267     \UWMad_Hash_Set:nnn {l__ListOf_Hook}{PostPrint} {}
1268 }

```

Function initializations for sectioning and title print commands.

```

1269 \cs_new:Nn \__UWMad_ListOf_SectioningCommand_Main: {}
1270 \cs_new:Nn \__UWMad_ListOf_SectioningCommand_Group: {}
1271 \cs_new:Nn \__UWMad_ListOf_SectioningCommand_Subgroup: {}
1272 \cs_new:Nn \UWMad_ListOf_PrintTitle_Main:nn {}
1273 \cs_new:Nn \UWMad_ListOf_PrintTitle_Group:nn {}
1274 \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-.\*

```

1275 \cs_new:Nn \UWMad_ListOf_SetHook:nn {
1276     \UWMad_Hash_Set:nnn{l__ListOf_Hook}{#1}{#2}
1277 }

```

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

```

1278 \cs_new:Nn \__UWMad_ListOf_Initialize_SectioningCommand:n {
1279     \UWMad_IfSectionExists:nT {
1280         \tl_use:c {l__UWMad_ListOf_Section_ #1 _tl}
1281     } {
1282         \cs_set_eq:cc
1283             {__UWMad_ListOf_SectioningCommand_ #1 :}
1284             {\tl_use:c {l__UWMad_ListOf_Section_ #1 _tl}}

```

```

1285     }
1286 }
1287 \cs_new:Nn \__UWMad_ListOf_Initialize_SectioningCommands: {
1288     \__UWMad_ListOf_Initialize_SectioningCommand:n {Main}
1289     \__UWMad_ListOf_Initialize_SectioningCommand:n {Group}
1290     \__UWMad_ListOf_Initialize_SectioningCommand:n {Subgroup}
1291 }

```

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

```

1292 \cs_new:Nn \__UWMad_ListOf_Initialize_TitlePrinter:n {
1293     \cs_set:cn {UWMad_ListOf_PrintTitle_ #1 :nn} {
1294         \UWMad_Hash_Get:nn{l__ListOf_Hook}{PreTitle-#1}
1295         \bool_if:cTF {l__UWMad_ListOf_MakeNumbered_ #1 _bool} {
1296             \bool_if:cTF {l__UWMad_ListOf_IncludeInTOC_ #1 _bool} {
1297                 \use:c{__UWMad_ListOf_SectioningCommand_ #1 :}{##1}{##2}
1298             } {
1299                 \int_gset_eq:NN \l_tmpa_int \c@tocdepth
1300                 \int_gset:Nn \c@tocdepth {-1}
1301                 \use:c{__UWMad_ListOf_SectioningCommand_ #1 :}{##2}
1302                 \int_gset:Nn \c@tocdepth {\l_tmpa_int}
1303             }
1304         } {
1305             \use:c{__UWMad_ListOf_SectioningCommand_ #1 :} * {##2}
1306             \bool_if:cTF {l__UWMad_ListOf_IncludeInTOC_ #1 _bool} {
1307                 \addcontentsline
1308                     {toc}
1309                     {\tl_use:c{l__UWMad_ListOf_Section_ #1 _tl}}
1310                     {##1}
1311             } { }
1312         }
1313         \UWMad_Hash_Get:nn{l__ListOf_Hook}{PostTitle-#1}
1314     }
1315 }
1316 \cs_new:Nn \__UWMad_ListOf_Initialize_TitlePrinters: {
1317     \__UWMad_ListOf_Initialize_TitlePrinter:n {Main}
1318     \__UWMad_ListOf_Initialize_TitlePrinter:n {Group}
1319     \__UWMad_ListOf_Initialize_TitlePrinter:n {Subgroup}
1320 }

```

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

```

1321 \cs_new:Nn \__UWMad_ListOf_Clear: {
1322     \UWMad_Queue_Clear:n {l__ListOf_EntryQueue}
1323     \__UWMad_ListOf_SetHooks_Blank:

```

```

1324 }
1325 \cs_new:Nn \UWMad_ListOf_Initialize: {
1326   \__UWMad_ListOf_Clear:
1327   \__UWMad_ListOf_Initialize_SectioningCommands:
1328   \__UWMad_ListOf_Initialize_TitlePrinters:
1329 }

```

---

```

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

```

---

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

```

1330 \cs_new:Nn \UWMad_ListOf_PushEntry:n {
1331   \UWMad_Hash_Get:nn {l__ListOf_Hook}{PrePush}
1332   \UWMad_Queue_Push:nn {l__ListOf_EntryQueue}{#1}
1333   \UWMad_Hash_Get:nn {l__ListOf_Hook}{PostPush}
1334 }

```

---

```

\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.

```

1335 \cs_new:Nn \UWMad_ListOf_PrintEntries: {
1336   \UWMad_Hash_Get:nn {l__ListOf_Hook}{PrePrint}
1337   \UWMad_Queue_Walk:nn {l__ListOf_EntryQueue}{##1}
1338   \UWMad_Queue_Clear:n {l__ListOf_EntryQueue}
1339   \UWMad_Hash_Get:nn {l__ListOf_Hook}{PostPrint}
1340 }

```

## 6.1 Nomenclature

Dimensions that are calculated and private.

```

1341 \dim_new:N \l__UWMad_Nomenclature_WidestSymbol_dim
1342 \dim_new:N \l__UWMad_Nomenclature_WidestUnit_dim
1343 \dim_new:N \l__UWMad_Nomenclature_Entry_Symbol_Width_dim
1344 \dim_new:N \l__UWMad_Nomenclature_Entry_Units_Width_dim
1345 \dim_new:N \l__UWMad_Nomenclature_Entry_Description_Width_dim

```

User-adjustable dimensions that are public.

```

1346 \dim_new:N \l_UWMad_Nomenclature_Skip_EntryPrint_dim
1347 \dim_new:N \l_UWMad_Nomenclature_Entry_Margin_Left_dim
1348 \dim_new:N \l_UWMad_Nomenclature_Entry_Margin_Bottom_dim
1349 \dim_new:N \l_UWMad_Nomenclature_Entry_Margin_Right_dim
1350 \dim_new:N \l_UWMad_Nomenclature_Entry_Margin_Top_dim
1351 \dim_new:N \l_UWMad_Nomenclature_Entry_Pad_Column_dim

```

Coffins used in typesetting an entry's contents.

```

1352 \coffin_new:N \l__UWMad_Nomenclature_Entry_coffin
1353 \coffin_new:N \l__UWMad_Nomenclature_Symbol_coffin
1354 \coffin_new:N \l__UWMad_Nomenclature_Description_coffin
1355 \coffin_new:N \l__UWMad_Nomenclature_Units_coffin

```

Options for the units column.

```

1356 \bool_new:N \l_UWMad_Nomenclature_Units_IncludeColumn_bool
1357 \bool_new:N \l_UWMad_Nomenclature_Units_UseSIUnitx_bool
1358 \bool_new:N \l_UWMad_Nomenclature_Units_UseDelimiter_bool
1359 \tl_new:N \l_UWMad_Nomenclature_Units_Delimiter_Left_tl
1360 \tl_new:N \l_UWMad_Nomenclature_Units_Delimiter_Right_tl

```

Miscellaneous token lists.

```

1361 \tl_new:N \l__UWMad_Nomenclature_Entry_LineStretch_tl
1362 \tl_new:N \l__UWMad_Nomenclature_Title_Main_tl

1363 \cs_new:Nn \UWMad_Nomenclature_SetUnitsBox:n {
1364   \bool_if:nTF {
1365     \l_UWMad_Nomenclature_Units_UseDelimiter_bool &&
1366     \l_UWMad_Nomenclature_Units_UseSIUnitx_bool
1367   } {
1368     $
1369     \left\l_UWMad_Nomenclature_Units_Delimiter_Left_tl
1370     \si{#1}
1371     \right\l_UWMad_Nomenclature_Units_Delimiter_Right_tl
1372     $
1373   } {
1374     \bool_if:nTF {
1375       \l_UWMad_Nomenclature_Units_UseDelimiter_bool &&

```

```

1376         !\l_UWMad_Nomenclature_Units_UseSIUnitx_bool
1377     } {
1378         $
1379         \left\l_UWMad_Nomenclature_Units_Delimiter_Left_tl
1380         #1
1381         \right\l_UWMad_Nomenclature_Units_Delimiter_Right_tl
1382         $
1383     }{
1384         \si{#1}
1385     }
1386 }
1387 }

```

---

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

---

These commands update the widest symbol and widest unit lengths.

```

1388 \cs_new:Nn \UWMad_Nomenclature_UpdateWidest:Nn {
1389     \hbox_set:Nn \l_tmpa_box {#2}
1390     \dim_set:Nn \l_tmpa_dim {\box_wd:N \l_tmpa_box}
1391     \dim_compare:nNnTF {#1} < {\l_tmpa_dim} {
1392         \dim_set_eq:NN #1 \l_tmpa_dim
1393     } { }
1394 }
1395 \cs_new:Nn \UWMad_Nomenclature_UpdateWidest_Symbol:n {
1396     \UWMad_Nomenclature_UpdateWidest:Nn
1397     \l__UWMad_Nomenclature_WidestSymbol_dim {#1}
1398 }
1399 %
1400 \cs_new:Nn \UWMad_Nomenclature_UpdateWidest_Units:n {
1401     \UWMad_Nomenclature_UpdateWidest:Nn
1402     \l__UWMad_Nomenclature_WidestUnit_dim
1403     {
1404         \UWMad_Nomenclature_SetUnitsBox:n{#1}
1405     }
1406 }

```

And the defaults for all keys are now set.



---

<code>\UWMad_Nomenclature_ZeroWidest_Symbol:</code>	<code>\UWMad_Nomenclature_ZeroWidest_Symbol:</code>
<code>\UWMad_Nomenclature_ZeroWidest_Unit:</code>	<code>\UWMad_Nomenclature_ZeroWidest_Symbol:</code>

---

These commands set the widest symbol and unit lengths to Opt.

```

1407 \cs_new:Nn \UWMad_Nomenclature_ZeroWidest_Symbol: {
1408     \dim_set:Nn \l__UWMad_Nomenclature_WidestSymbol_dim {Opt}
1409 }
1410 \cs_new:Nn \UWMad_Nomenclature_ZeroWidest_Units: {
1411     \dim_set:Nn \l__UWMad_Nomenclature_WidestUnit_dim {Opt}
1412 }
```

And the defaults for all keys are now set.

---

<code>\UWMad_Nomenclature_SetEntryWidths_NoUnits:</code>	<code>\UWMad_Nomenclature_SetEntryWidths_NoUnits:</code>
<code>\UWMad_Nomenclature_SetEntryWidths_Units:</code>	<code>\UWMad_Nomenclature_SetEntryWidths_Units:</code>

---

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

```

1413 \cs_new:Nn \UWMad_Nomenclature_SetEntryWidths_NoUnits: {
1414     \dim_set:Nn \l__UWMad_Nomenclature_Entry_Symbol_Width_dim {
1415         1.01\l__UWMad_Nomenclature_WidestSymbol_dim
1416     }
1417     \dim_set:Nn \l__UWMad_Nomenclature_Entry_Description_Width_dim {
1418         0.995\textwidth -
1419         \l_UWMad_Nomenclature_Entry_Margin_Left_dim -
1420         \l__UWMad_Nomenclature_Entry_Symbol_Width_dim -
1421         \l_UWMad_Nomenclature_Entry_Pad_Column_dim -
1422         \l_UWMad_Nomenclature_Entry_Margin_Right_dim
1423     }
1424 }
1425 \cs_new:Nn \UWMad_Nomenclature_SetEntryWidths_Units: {
1426     \dim_set:Nn \l__UWMad_Nomenclature_Entry_Symbol_Width_dim {
1427         1.05\l__UWMad_Nomenclature_WidestSymbol_dim
1428     }
1429     \dim_set:Nn \l__UWMad_Nomenclature_Entry_Units_Width_dim {
1430         1.05\l__UWMad_Nomenclature_WidestUnit_dim
1431     }
1432     \dim_set:Nn \l__UWMad_Nomenclature_Entry_Description_Width_dim {
1433         0.995\textwidth -
1434         \l_UWMad_Nomenclature_Entry_Margin_Left_dim -
1435         \l__UWMad_Nomenclature_Entry_Symbol_Width_dim -
1436         \l__UWMad_Nomenclature_Entry_Units_Width_dim -
1437         2\l_UWMad_Nomenclature_Entry_Pad_Column_dim -
1438         \l_UWMad_Nomenclature_Entry_Margin_Right_dim
1439     }
1440 }
```

And the defaults for all keys are now set.

---

`\UWMad_Nomenclature_SetEntryWidths: \UWMad_Nomenclature_SetEntryWidths:`

---

This function calls one of the appropriate above setters.

```
1441 \cs_new:Nn \UWMad_Nomenclature_SetEntryWidths: {  
1442     \bool_if:NTF \l_UWMad_Nomenclature_Units_IncludeColumn_bool {  
1443         \UWMad_Nomenclature_SetEntryWidths_Units:  
1444     } {  
1445         \UWMad_Nomenclature_SetEntryWidths_NoUnits:  
1446     }  
1447 }
```

And the defaults for all keys are now set.

---

<code>\UWMad_Nomenclature_SetEntry_NoUnits:nn</code>	<code>\UWMad_Nomenclature_SetEntry_NoUnits:</code>
<code>\UWMad_Nomenclature_SetEntry_Units:nnn</code>	<code>{\symbol}{\description}</code>

---

	<code>\UWMad_Nomenclature_SetEntry_Units:</code>
	<code>{\symbol}{\units}{\description}</code>

These functions typeset the contents passed into them.

```

1448 \cs_new:Nn \UWMad_Nomenclature_SetEntry_NoUnits:nn {
1449   \coffin_clear:N \l__UWMad_Nomenclature_Entry_coffin
1450   \coffin_clear:N \l_tmpa_coffin
1451   \vcoffin_set:Nnn
1452     \l__UWMad_Nomenclature_Symbol_coffin
1453     {\l__UWMad_Nomenclature_Entry_Symbol_Width_dim} {#1}
1454   \vcoffin_set:Nnn
1455     \l__UWMad_Nomenclature_Description_coffin
1456     {\l__UWMad_Nomenclature_Entry_Description_Width_dim} {#2}
1457   \coffin_join:NnnNnnnn
1458     \l__UWMad_Nomenclature_Entry_coffin      {l}{t}
1459     \l__UWMad_Nomenclature_Symbol_coffin      {l}{t}
1460     {\l__UWMad_Nomenclature_Entry_Margin_Left_dim}{Opt}
1461   \coffin_join:NnnNnnnn
1462     \l__UWMad_Nomenclature_Entry_coffin      {r}{t}
1463     \l__UWMad_Nomenclature_Description_coffin {l}{t}
1464     {\l__UWMad_Nomenclature_Entry_Pad_Column_dim}{Opt}
1465   \group_begin:
1466     \setstretch{\l__UWMad_Nomenclature_Entry_LineStretch_tl}
1467     \skip_vertical:n{\l__UWMad_Nomenclature_Entry_Margin_Top_dim}
1468     \coffin_typeset:Nnnnn
1469       \l__UWMad_Nomenclature_Entry_coffin {l}{t}{Opt}{Opt}
1470     \skip_vertical:n{\l__UWMad_Nomenclature_Entry_Margin_Bottom_dim}
1471   \group_end:
1472 }
1473 \cs_new:Nn \UWMad_Nomenclature_SetEntry_Units:nnn {
1474   \coffin_clear:N \l__UWMad_Nomenclature_Entry_coffin
1475   %
1476   % Set the information into their coffins
1477   \vcoffin_set:Nnn
1478     \l__UWMad_Nomenclature_Symbol_coffin
1479     {\l__UWMad_Nomenclature_Entry_Symbol_Width_dim} {#1}
1480   \vcoffin_set:Nnn
1481     \l__UWMad_Nomenclature_Description_coffin
1482     {\l__UWMad_Nomenclature_Entry_Description_Width_dim} {#3}
1483   %
1484   % Units setting: center using hfil and then handle bracing and siunitx
1485   % embedding options.
1486   \hcoffin_set:Nn
1487     \l__UWMad_Nomenclature_Units_coffin
1488     {
1489       \UWMad_Nomenclature_SetUnitsBox:n{#2}
1490     }
1491   %

```

```

1521 \DeclareDocumentEnvironment {Nomenclature} { o o } {
1522 %
1523 %   Initialization
1524 \UWMad_ListOf_Initialize:
1525 \setlength{\parskip}{0pt}
1526 %
1527 %
1528 %   Check for an optional section declaration and
1529 %   set Main section token list.
1530 \IfValueTF {#1} {
1531     \tl_set:Nx \l__UWMad_ListOf_Title_Main_tl { #1 }
1532 } { }
1533 %
1534 %
1535 %
1536 %   Set some hooks in the Nomenclature ListOf instance
1537 \UWMad_ListOf_SetHook:nn {PrePrint} {
1538     \UWMad_Nomenclature_SetEntryWidths:
1539 }
1540 \UWMad_ListOf_SetHook:nn {PostPrint} {
1541     \UWMad_Nomenclature_ZeroWidest_Symbol:
1542     \UWMad_Nomenclature_ZeroWidest_Units:
1543 }
1544 %
1545 %
1546 %   User front-end for creating a Group
1547 \DeclareDocumentCommand \Group { o m } {
1548     \UWMad_ListOf_PrintEntries:
1549     \IfNoValueTF {##1} {
1550         \UWMad_ListOf_PrintTitle_Group:nn{##2}{##2}
1551     } {
1552         \UWMad_ListOf_PrintTitle_Group:nn{##1}{##2}
1553     }
1554 }
1555 %
1556 %   User front-end for creating a Subgroup
1557 \DeclareDocumentCommand \Subgroup { o m } {
1558     \UWMad_ListOf_PrintEntries:
1559     \IfNoValueTF {##1} {
1560         \UWMad_ListOf_PrintTitle_Subgroup:nn{##2}{##2}
1561     } {
1562         \UWMad_ListOf_PrintTitle_Subgroup:nn{##1}{##2}
1563     }
1564 }
1565 %
1566 %   User front-end for creating an entry

```

```

1567 \cs_undefine:N \Entry
1568 \bool_if:NTF \l_UWMad_Nomenclature_Units_IncludeColumn_bool {
1569   \DeclareDocumentCommand \Entry { m m m } {
1570     \UWMad_ListOf_PushEntry:n {
1571       \UWMad_Nomenclature_SetEntry_Units:nnn
1572         {##1} {##2} {##3}
1573     }
1574     \UWMad_Nomenclature_UpdateWidest_Symbol:n{##1}
1575     \UWMad_Nomenclature_UpdateWidest_Units:n{##2}
1576   }
1577 } {
1578   \DeclareDocumentCommand \Entry { m m } {
1579     \UWMad_ListOf_PushEntry:n {
1580       \UWMad_Nomenclature_SetEntry_NoUnits:nn
1581         {##1} {##2}
1582     }
1583     \UWMad_Nomenclature_UpdateWidest_Symbol:n{##1}
1584   }
1585 }
1586 %
1587 % User front-end for resetting the column width
1588 \DeclareDocumentCommand \PrintEntries { } {
1589   \UWMad_ListOf_PrintEntries:
1590 }
1591 %
1592 %
1593 \IfNoValueTF {#2} {
1594   \IfNoValueTF {#1} {
1595     \UWMad_ListOf_PrintTitle_Main:nn
1596       {\l_UWMad_Nomenclature_Title_Main_tl}
1597       {\l_UWMad_Nomenclature_Title_Main_tl}
1598   } {
1599     \UWMad_ListOf_PrintTitle_Main:nn{#1}{#1}
1600   }
1601 } {
1602   \UWMad_ListOf_PrintTitle_Main:nn{#1}{#2}
1603 }
1604 %
1605 } {
1606 % Flush the remaining entries from the ListOf queue.
1607 \UWMad_ListOf_PrintEntries:
1608 }
1609 %
1610 %
1611 %
1612 %

```

```

1613 %
1614 %
1615 %
1616 %
1617 \clist_new:N    \g__UWMad_Nomenclature_KeyValuePairs_clist
1618 \clist_gset:Nn \g__UWMad_Nomenclature_KeyValuePairs_clist {
1619     main-title .tl_set:N = \l__UWMad_Nomenclature_Title_Main_tl,
1620     main-title .default:n = Nomenclature,
1621     main-section .tl_set:N = \l__UWMad_ListOf_Section_Main_tl,
1622     group-section .tl_set:N = \l__UWMad_ListOf_Section_Group_tl,
1623     subgroup-section .tl_set:N = \l__UWMad_ListOf_Section_Subgroup_tl,
1624     main-section .default:n = chapter,
1625     group-section .default:n = section,
1626     subgroup-section .default:n = subsection,
1627     make-main-numbered .bool_set:N =
1628         \l__UWMad_ListOf_MakeNumbered_Main_bool,
1629     make-group-numbered .bool_set:N =
1630         \l__UWMad_ListOf_MakeNumbered_Group_bool,
1631     make-subgroup-numbered .bool_set:N =
1632         \l__UWMad_ListOf_MakeNumbered_Subgroup_bool,
1633     make-numbered .meta:n = {
1634         make-main-numbered = #1,
1635         make-group-numbered = #1,
1636         make-subgroup-numbered = #1
1637     },
1638     make-numbered .default:n = false,
1639     include-main-in-toc .bool_set:N =
1640         \l__UWMad_ListOf_IncludeInTOC_Main_bool,
1641     include-group-in-toc .bool_set:N =
1642         \l__UWMad_ListOf_IncludeInTOC_Group_bool,
1643     include-subgroup-in-toc .bool_set:N =
1644         \l__UWMad_ListOf_IncludeInTOC_Subgroup_bool,
1645     include-in-toc .meta:n = {
1646         include-main-in-toc = #1,
1647         include-group-in-toc = #1,
1648         include-subgroup-in-toc = #1
1649     },
1650     include-in-toc .default:n = true,
1651     print-skip .dim_set:N =
1652         \l__UWMad_Nomenclature_Skip_EntryPrint_dim,
1653     entry-margin-top .dim_set:N =
1654         \l__UWMad_Nomenclature_Entry_Margin_Top_dim,
1655     entry-margin-left .dim_set:N =
1656         \l__UWMad_Nomenclature_Entry_Margin_Left_dim,
1657     entry-margin-right .dim_set:N =
1658         \l__UWMad_Nomenclature_Entry_Margin_Right_dim,

```

```

1659 entry-margin-bottom .dim_set:N =
1660     \l_UWMad_Nomenclature_Entry_Margin_Bottom_dim,
1661 entry-column-padding .dim_set:N =
1662     \l_UWMad_Nomenclature_Entry_Pad_Column_dim,
1663 print-skip .default:n = 1em,
1664 entry-margin-top .default:n = 0pt,
1665 entry-margin-left .default:n = 1.1em,
1666 entry-margin-right .default:n = 0pt,
1667 entry-margin-bottom .default:n = 0.80em,
1668 entry-column-padding .default:n = 0.80em,
1669 entry-stretch .tl_set:N =
1670     \l__UWMad_Nomenclature_Entry_LineStretch_tl,
1671 entry-stretch .default:n = 1.1,
1672 include-units-column .bool_set:N =
1673     \l_UWMad_Nomenclature_Units_IncludeColumn_bool,
1674 include-units-column .default:n = false,
1675 units-embed-siunitx .bool_set:N =
1676     \l_UWMad_Nomenclature_Units_UseSIUnitx_bool,
1677 units-embed-siunitx .default:n = false,
1678 units-left-delimiter .code:n = {
1679     \tl_set:Nn \l_UWMad_Nomenclature_Units_Delimiter_Left_tl {#1}
1680     \bool_set_true:N \l_UWMad_Nomenclature_Units_UseDelimiter_bool
1681 },
1682 units-right-delimiter .code:n = {
1683     \tl_set:Nn \l_UWMad_Nomenclature_Units_Delimiter_Right_tl {#1}
1684     \bool_set_true:N \l_UWMad_Nomenclature_Units_UseDelimiter_bool
1685 }
1686 }
1687 %
1688 %
1689 \exp_args:Nnf
1690     \keys_define:nn
1691     { UWMadThesis / Nomenclature }
1692     {
1693         \clist_use:Nn \g__UWMad_Nomenclature_KeyValuePairs_clist {,}
1694     }
1695 %
1696 %
1697 %
1698 \keys_set:nn { UWMadThesis / Nomenclature } {
1699     main-title ,
1700     main-section ,
1701     group-section ,
1702     subgroup-section ,
1703     make-numbered ,
1704     include-in-toc ,

```



```

1705     include-units-column ,
1706     units-embed-siunitx ,
1707     print-skip           ,
1708     entry-margin-top     ,
1709     entry-margin-left    ,
1710     entry-margin-right   ,
1711     entry-margin-bottom  ,
1712     entry-column-padding ,
1713     entry-stretch
1714 }
1715 %
1716 %
1717 %
1718 %
1719 %
1720 %
1721 %
1722 %
1723 %
1724 \tl_new:N \l__UWMad_Acronym_Title_Main_tl
1725 %
1726 \DeclareDocumentEnvironment {Acronym} { o o } {
1727
1728     \IfNoValueTF {#2} {
1729         \IfNoValueTF {#1} {
1730             \begin{Nomenclature}
1731                 [\l__UWMad_Acronym_Title_Main_tl]
1732                 [\l__UWMad_Acronym_Title_Main_tl]
1733             } {
1734                 \begin{Nomenclature}[#1][#1]
1735             }
1736         } {
1737             \begin{Nomenclature}[#1][#2]
1738         }
1739
1740 %
1741 %
1742 \UWMad_Hash_Define:n{Acronyms}
1743 \UWMad_Hash_Define:n{AcronymMeanings}
1744 %
1745 %
1746 \cs_undefine:N \Entry
1747 \DeclareDocumentCommand \Entry { o m m } {
1748     \IfNoValueTF {##1} {
1749
1750         \UWMad_Hash_Set:nnn{Acronyms}          {##2}{##2}

```

```

1751         \UWMad_Hash_Set:nnn{AcronymMeanings}{##2}{##3}
1752         \bool_new:c {g__UWMad_Acronym_WasSet_##2_bool}
1753         %
1754         \UWMad_ListOf_PushEntry:n {
1755             \hypertarget{Acronym:##2}{
1756                 \UWMad_Nomenclature_SetEntry_NoUnits:nn
1757                     {##2} {##3}
1758             }
1759
1760     } {
1761
1762         \UWMad_Hash_Set:nnn{Acronyms}          {##1}{##2}
1763         \UWMad_Hash_Set:nnn{AcronymMeanings}{##1}{##3}
1764         \bool_new:c {g__UWMad_Acronym_WasSet_##1_bool}
1765         %
1766         \UWMad_ListOf_PushEntry:nn {Nomenclature} {
1767             \hypertarget{Acronym:##1}{
1768                 \UWMad_Nomenclature_SetEntry_NoUnits:nn
1769                     {##2} {##3}
1770             }
1771
1772     }
1773     \UWMad_Nomenclature_UpdateWidest_Symbol:n{##2}
1774 }
1775 } {
1776
1777     \end{Nomenclature}
1778
1779 }
1780 %
1781 %
1782 %
1783 \cs_new:Nn \UWMad_Acronym_CreateLink:n {
1784     \hyperlink{Acronym:#1}{
1785         \color{\g__UWMad_Acronym_LinkColor_tl}
1786         \UWMad_Hash_Get:nn{Acronyms}{#1}
1787     }
1788 }
1789 %
1790 %
1791 \DeclareDocumentCommand \Acro { m } {
1792     \UWMad_Hash_IfKeySet:nnTF {Acronyms} {#1} {
1793         \bool_if:cTF {g__UWMad_Acronym_WasSet_#1_bool} {
1794             \bool_if:NTF \g__UWMad_Acronym_UseLinks_bool {
1795                 \UWMad_Acronym_CreateLink:n{#1}
1796             } {

```

```

1797         \UWMad_Hash_Get:nn{Acronyms}{#1}
1798     }
1799 } {
1800     \UWMad_Hash_Get:nn{AcronymMeanings}{#1}~
1801     (
1802         \UWMad_Hash_Get:nn{Acronyms}{#1}
1803     )
1804     \bool_gset_true:c {g__UWMad_Acronym_WasSet_#1_bool}
1805 }
1806 } { }
1807 }
1808 %
1809 %

```

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

```

1810 \exp_args:Nnf
1811     \keys_define:nn
1812     { UWMadThesis / Acronym }
1813     {
1814         \clist_use:Nn \g__UWMad_Nomenclature_KeyValuePairs_clist {,}
1815     }
1816 \keys_define:nn { UWMadThesis / Acronym } {
1817     main-title .tl_set:N = \l__UWMad_Acronym_Title_Main_tl,
1818     main-title .default:n = Acronyms,
1819     use-links .bool_gset:N = \g__UWMad_Acronym_UseLinks_bool,
1820     use-links .default:n = true,
1821     link-color .tl_gset:N = \g__UWMad_Acronym_LinkColor_tl,
1822     link-color .default:n = blue
1823 }

```

And the defaults for all keys are now set.

```

1824 \keys_set:nn { UWMadThesis / Acronym } {
1825     main-title ,
1826     main-section ,
1827     group-section ,
1828     subgroup-section ,
1829     make-numbered ,
1830     include-in-toc ,
1831     include-units-column ,
1832     print-skip ,
1833     entry-margin-top ,
1834     entry-margin-left ,

```

```
1835     entry-margin-right    ,
1836     entry-margin-bottom    ,
1837     entry-column-padding    ,
1838     entry-stretch           ,
1839     use-links                ,
1840     link-color
1841 }
```

## 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|`.

```
1842 \clist_new:N \g__UWMad_MetaDataList_clist
```

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

```
1843 \cs_new:Nn \UWMad_MetaData_PushToList:nn {
1844   \clist_gput_right:Nn \g__UWMad_MetaDataList_clist {
1845     #1={#2}
1846   }
1847 }
```

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

```
1848 \bool_new:N \g__UWMad_MetaData_GenerateAux_bool
1849 \bool_new:N \g__UWMad_MetaData_IsDocument_bool
```

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

```
1850 \file_if_exist:nTF{\c_job_name_tl.UWMad.PDFMetaData.aux} {
```

```

1851 \file_input:n {\c_job_name_tl.UWMad.PDFMetaData.aux}
1852 }{}

```

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.

```

1853 \AtBeginDocument{
1854   \clist_if_empty:NTF \g__UWMad_MetaDataList_clist { } {
1855     \exp_args:Nx \hypersetup {
1856       \clist_use:Nn\g__UWMad_MetaDataList_clist{,}
1857     }
1858   } { }
1859   \bool_gset_true:N \g__UWMad_MetaData_IsDocument_bool
1860 }

```

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

```

1861 \AtEndDocument{
1862   \bool_if:NTF \g__UWMad_MetaData_GenerateAux_bool {
1863     \clist_if_empty:NTF \g__UWMad_MetaDataList_clist { } {
1864       \iow_new:N \g__UWMad_PDFMetaData_HyperSetup_io
1865       \iow_open:Nn \g__UWMad_PDFMetaData_HyperSetup_io {
1866         \c_job_name_tl.UWMad.PDFMetaData.aux
1867       }
1868       \iow_now:Nx \g__UWMad_PDFMetaData_HyperSetup_io {
1869         \noexpand\ExplSyntaxOff
1870         \noexpand\hypersetup
1871         {\clist_use:Nn\g__UWMad_MetaDataList_clist{,}}
1872         \noexpand\ExplSyntaxOn
1873       }
1874       \iow_close:N \g__UWMad_PDFMetaData_HyperSetup_io
1875     } { }
1876   } { }
1877 }

```

## 7.2 Thesis Information

Declare the `|ThesisInfo|` token list variables.

```

1878 \tl_new:N \g__UWMad_ThesisInfo_Title_tl
1879 \tl_new:N \g__UWMad_ThesisInfo_Author_tl
1880 \tl_new:N \g__UWMad_ThesisInfo_DefenseDate_tl
1881 \tl_new:N \g__UWMad_ThesisInfo_Department_tl
1882 \tl_new:N \g__UWMad_ThesisInfo_Program_tl
1883 \tl_new:N \g__UWMad_ThesisInfo_Degree_tl
1884 \tl_new:N \g__UWMad_ThesisInfo_DocumentType_tl
1885 \tl_new:N \g__UWMad_ThesisInfo_AdvisorName_tl
1886 \tl_new:N \g__UWMad_ThesisInfo_AdvisorPosition_tl
1887 \tl_new:N \g__UWMad_ThesisInfo_AdvisorAssociation_tl
1888 \tl_new:N \g__UWMad_ThesisInfo_AdvisorMarker_tl
1889 \tl_new:N \g__UWMad_ThesisInfo_Institution_tl

```

Set the document type default.

```

1890 \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {report}

```

Define some booleans for required information.

```

1891 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Title_bool
1892 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Author_bool
1893 \bool_new:N \g__UWMad_ThesisInfo_IsSet_DefenseDate_bool
1894 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Program_bool
1895 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
1896 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Institution_bool
1897 \bool_new:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool

```

Declare the user front-end for the title.

```

1898 \DeclareDocumentCommand \Title { m } {

```

Set the associated token list variable

```

1899     \tl_gset:Nn \g__UWMad_ThesisInfo_Title_tl {#1}

```

Pass it to the default L<sup>A</sup>T<sub>E</sub>X \title command.

```

1900     \title{#1}

```

Push the value to the MetaData |clist|.

```

1901     \UWMad_MetaData_PushToList:nn{pdftitle}    {#1}

```

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

```

1902     \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
1903         \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
1904     } { }

```

Tell the class this variable is now set.

```

1905     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Title_bool
1906 }

```

Similar flow to the \Title definition.

```

1907 \DeclareDocumentCommand \Author { m } {
1908     \tl_gset:Nn \g__UWMad_ThesisInfo_Author_tl {#1}
1909     \author{#1}
1910     \UWMad_MetaData_PushToList:nn{pdfauthor} {#1}
1911     \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
1912         \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
1913     } { }
1914     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Author_bool
1915 }

```

A simple setter command.

```

1916 \DeclareDocumentCommand \Program { m } {
1917     \tl_gset:Nn \g__UWMad_ThesisInfo_Program_tl {#1}
1918     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Program_bool
1919 }

```

A simple setter command.

```

1920 \DeclareDocumentCommand \Degree { m } {
1921     \tl_gset:Nn \g__UWMad_ThesisInfo_Degree_tl {#1}
1922     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
1923 }

```

Semantic names for the \Degree function.

```

1924 \DeclareDocumentCommand \Doctorate { } {
1925     \tl_gset:Nn \g__UWMad_ThesisInfo_Degree_tl {Doctor~of~Philosophy}

```



```

1926     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
1927 }
1928 \DeclareDocumentCommand \Masters { } {
1929     \tl_gset:Nn \g__UWMad_ThesisInfo_Degree_tl {Master's}
1930     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
1931 }
1932 \DeclareDocumentCommand \Bachelors { } {
1933     \tl_gset:Nn \g__UWMad_ThesisInfo_Degree_tl {Bachelor's}
1934     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Degree_bool
1935 }

```

A simple setter command.

```

1936 \DeclareDocumentCommand \DocumentType { m } {
1937     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {#1}
1938 }

```

Semantic names for the \DocumentType function.

```

1939 \DeclareDocumentCommand \Dissertation { } {
1940     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
1941         dissertation
1942     }
1943 }
1944 \DeclareDocumentCommand \DoctoralThesis { } {
1945     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
1946         doctoral~thesis
1947     }
1948 }
1949 \DeclareDocumentCommand \MastersThesis { } {
1950     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
1951         master's~thesis
1952     }
1953 }
1954 \DeclareDocumentCommand \Thesis { } {
1955     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
1956         thesis
1957     }
1958 }
1959 \DeclareDocumentCommand \Prelim { } {
1960     \tl_gset:Nn \g__UWMad_ThesisInfo_DocumentType_tl {
1961         preliminary~report
1962     }
1963 }

```

A simple setter command and aliases.

```

1964 \DeclareDocumentCommand \DefenseDate { m } {
1965     \tl_gset:Nn \g__UWMad_ThesisInfo_DefenseDate_tl {#1}
1966     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_DefenseDate_bool
1967 }
1968 \cs_gset_eq:NN \DefenceDate \DefenseDate

```

A simple setter command and alias.

```

1969 \DeclareDocumentCommand \Institution { m } {
1970     \tl_gset:Nn \g__UWMad_ThesisInfo_Institution_tl {#1}
1971     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Institution_bool
1972 }
1973 \cs_set_eq:NN \University \Institution

```

Define the optional user interface.

```

1974 \DeclareDocumentCommand \Department { m } {
1975     \tl_gset:Nn \g__UWMad_ThesisInfo_Department_tl {#1}
1976 }

```

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

```

1977 \msg_new:nnn { UWMadThesis } { ThesisInfo / UnsetInformation } {
1978     The~required~information~for~the~#1~is~not~set.
1979 }
1980 \DeclareDocumentCommand \IfInfoIsSetT { m +m } {
1981     \bool_if:cTF {g__UWMad_ThesisInfo_IsSet_ #1 _bool} {
1982         #2
1983     } {
1984         \msg_error:nnn
1985             { UWMadThesis }
1986             { ThesisInfo / UnsetInformation }
1987             {#1}
1988     }
1989 }

```

Define user accessors for thesis info.

```

1990 \DeclareDocumentCommand \TheTitle { } {
1991     \g__UWMad_ThesisInfo_Title_tl

```

```

1992 }
1993 \DeclareDocumentCommand \TheAuthor { } {
1994     \g__UWMad_ThesisInfo_Author_tl
1995 }
1996 \DeclareDocumentCommand \TheProgram { } {
1997     \g__UWMad_ThesisInfo_Program_tl
1998 }
1999 \DeclareDocumentCommand \TheDegree { } {
2000     \g__UWMad_ThesisInfo_Degree_tl
2001 }
2002 \DeclareDocumentCommand \TheDocumentType { } {
2003     \g__UWMad_ThesisInfo_DocumentType_tl
2004 }
2005 \DeclareDocumentCommand \TheDefenseDate { } {
2006     \g__UWMad_ThesisInfo_DefenseDate_tl
2007 }
2008 \cs_gset_eq:NN \TheDefenceDate \TheDefenseDate
2009 \DeclareDocumentCommand \TheInstitution { } {
2010     \g__UWMad_ThesisInfo_Institution_tl
2011 }
2012 \cs_set_eq:NN \TheUniversity \TheInstitution
2013 %
2014 \DeclareDocumentCommand \TheDepartment { } {
2015     \g__UWMad_ThesisInfo_Department_tl
2016 }
2017 \DeclareDocumentCommand \TheAdvisor { } {
2018     \g__UWMad_ThesisInfo_AdvisorName_tl
2019 }

```

## 7.3 Committee Member List

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

```

2020 \tl_new:N \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl
2021 \tl_gset:Nn \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl {,}
2022 \int_new:N \g__UWMad_ThesisInfo_CommitteeCount_int
2023 \coffin_new:N \g__UWMad_ThesisInfo_Committee_coffin
2024 \vcoffin_set:Nnn \g__UWMad_ThesisInfo_Committee_coffin {\textwidth}{\}
2025 \seq_new:N \g__UWMad_ThesisInfo_Committee_CoffinExpanders_seq

```

```

2026 \cs_new:Nn \__UWMad_ThesisInfo_Committee_AddMember:nnn {
2027   \seq_gput_right:Nn \g__UWMad_ThesisInfo_Committee_CoffinExpanders_seq {
2028     \vcoffin_set:Nnn \l_tmpa_coffin {\textwidth-1.01em} {
2029       #1
2030       \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl{}
2031       \
2032       \textsl{#2}
2033       \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl{}
2034       \
2035       \textsl{#3}
2036     }
2037     \coffin_join:NnnNnnnn
2038     \g__UWMad_ThesisInfo_Committee_coffin {l} {b}
2039     \l_tmpa_coffin {l} {t}
2040     {0pt}{-0.75em}
2041   }
2042 }
2043 \cs_new:Nn \__UWMad_ThesisInfo_Committee_AddAdvisor:nnn {
2044   \seq_gput_left:Nn \g__UWMad_ThesisInfo_Committee_CoffinExpanders_seq {
2045     \vcoffin_set:Nnn \l_tmpa_coffin {\textwidth-1.01em} {
2046       #1
2047       \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl{}
2048       \
2049       \textsl{#2}
2050       \g__UWMad_ThesisInfo_Committee_InfoSeparator_tl{}
2051       \
2052       \textsl{#3}
2053       \
2054       (\g__UWMad_ThesisInfo_AdvisorMarker_tl{})
2055     }
2056     \coffin_join:NnnNnnnn
2057     \g__UWMad_ThesisInfo_Committee_coffin {l} {b}
2058     \l_tmpa_coffin {l} {t}
2059     {0pt}{-0.75em}
2060   }
2061 }

```

Define the Advisor and Adviser user interface.

```

2062 \cs_new:Nn \UWMad_ThesisInfo_AdvisorInfo:nnn {
2063   \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorName_tl {#1}
2064   \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorPosition_tl {#2}
2065   \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorAssociation_tl {#3}
2066   \__UWMad_ThesisInfo_Committee_AddAdvisor:nnn{#1}{#2}{#3}
2067 }

```

```

2068 \DeclareDocumentCommand \Advisor { m m m } {
2069     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool
2070     \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorMarker_tl {Advisor}
2071     \UWMad_ThesisInfo_AdvisorInfo:nnn{#1}{#2}{#3}
2072 }
2073 \DeclareDocumentCommand \Adviser { m m m } {
2074     \bool_gset_true:N \g__UWMad_ThesisInfo_IsSet_Advisor_bool
2075     \tl_gset:Nn \g__UWMad_ThesisInfo_AdvisorMarker_tl {Adviser}
2076     \UWMad_ThesisInfo_AdvisorInfo:nnn{#1}{#2}{#3}
2077 }

```

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

```

2078 \DeclareDocumentCommand \CommitteeMember { m m m } {
2079     \int_gincr:N \g__UWMad_ThesisInfo_CommitteeCount_int
2080     \__UWMad_ThesisInfo_Committee_AddMember:nnn{#1}{#2}{#3}
2081 }

```

Define an author interface for printing the Committee member list.

```

2082 \DeclareDocumentCommand \PrintCommitteeMemberList { } {
2083     {
2084         \seq_map_inline:Nn \g__UWMad_ThesisInfo_Committee_CoffinExpanders_seq {
2085             ##1
2086         }
2087         \coffin_typeset:Nnnnn \g__UWMad_ThesisInfo_Committee_coffin
2088         {l}{t}{1em}{Opt}
2089     }
2090 }

```

## 7.4 PDF Metadata

Define metadata internals.

```

2091 \tl_new:N \g__UWMad_PDFMetaData_Subject_tl
2092 \tl_new:N \g__UWMad_PDFMetaData_Keywords_tl
2093 \tl_new:N \g__UWMad_PDFMetaData_Producer_tl
2094 \tl_new:N \g__UWMad_PDFMetaData_Creator_tl

```

Define user interface for setting metadata.

```

2095 \DeclareDocumentCommand \Subject { m } {
2096   \tl_gset:Nn \g__UWMad_PDFMetaData_Subject_tl {#1}
2097   \UWMad_MetaData_PushToList:nn{pdfsubject} {#1}
2098   \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
2099     \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
2100   } { }
2101 }
2102 \DeclareDocumentCommand \Keywords { m } {
2103   \tl_gset:Nn \g__UWMad_PDFMetaData_Keywords_tl {#1}
2104   \UWMad_MetaData_PushToList:nn{pdfkeywords} {#1}
2105   \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
2106     \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
2107   } { }
2108 }
2109 \DeclareDocumentCommand \Producer { m } {
2110   \tl_gset:Nn \g__UWMad_PDFMetaData_Producer_tl {#1}
2111   \UWMad_MetaData_PushToList:nn{pdfproducer} {#1}
2112   \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
2113     \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
2114   } { }
2115 }
2116 \DeclareDocumentCommand \Creator { m } {
2117   \tl_gset:Nn \g__UWMad_PDFMetaData_Creator_tl {#1}
2118   \UWMad_MetaData_PushToList:nn{pdfcreator} {#1}
2119   \bool_if:NTF \g__UWMad_MetaData_IsDocument_bool {
2120     \bool_gset_true:N \g__UWMad_MetaData_GenerateAux_bool
2121   } { }
2122 }

```

Define user interface for accessing metadata.

```

2123 \DeclareDocumentCommand \TheSubject { } {
2124   \g__UWMad_PDFMetaData_Subject_tl
2125 }
2126 \DeclareDocumentCommand \TheKeywords { } {
2127   \g__UWMad_PDFMetaData_Keywords_tl
2128 }
2129 \DeclareDocumentCommand \TheProducer { } {
2130   \g__UWMad_PDFMetaData_Producer_tl
2131 }
2132 \DeclareDocumentCommand \TheCreator { } {
2133   \g__UWMad_PDFMetaData_Creator_tl

```

2134 }  
2135 %

## Module 8

# Special Pages

## 8.1 MakeTitlePage

```

2136 % That phrase that occurs on every title page design the class author has seen
2137 \DeclareDocumentCommand \FulfillmentClause { } {
2138   {
2139     \setstretch{1.1}
2140     A~\TheDocumentType{}~submitted~in~partial~fulfillment~of~the~
2141     requirements~for~the~degree~of
2142   }
2143 }
2144
2145 \DeclareDocumentCommand \TitlePageTitle { } {
2146   \IfInfoIsSetT {Title} {
2147     {
2148       \LARGE
2149       \textsc {\TheTitle}
2150     }
2151   }
2152 }
2153
2154 \DeclareDocumentCommand \TitlePageAuthor { } {
2155   \IfInfoIsSetT {Author} {
2156     {
2157       \large
2158       by \[0.50em]
2159       \TheAuthor{}
2160     }
2161   }
2162 }
2163
2164 \DeclareDocumentCommand \TitlePageFulFillment { } {

```



```

2165     \FulfillmentClause{}
2166 }
2167
2168 \DeclareDocumentCommand \TitlePageDegree { } {
2169     \IfInfoIsSetT {Degree} {
2170         (\TheDegree{})
2171     }
2172 }
2173
2174 \DeclareDocumentCommand \TitlePageProgram { } {
2175     \IfInfoIsSetT {Program} {
2176         \TheProgram{}
2177     }
2178 }
2179
2180 \DeclareDocumentCommand \TitlePageInstitution { } {
2181     \IfInfoIsSetT {Institution} {
2182         at~the                \\\[0.50em]
2183         \textsc{\TheInstitution{}} \\\[0.50em]
2184         \the\year
2185     }
2186 }
2187
2188 \DeclareDocumentCommand \TitlePageDefenseDate { } {
2189     \IfInfoIsSetT {DefenseDate} {
2190         Date~of~final~oral~examination:~\TheDefenseDate{}
2191     }
2192 }
2193
2194
2195 \DeclareDocumentCommand \MakeTitlePage { } {
2196     \clearpage
2197     \thispagestyle{empty}
2198     \begin{center}
2199         \TitlePageTitle{}        \\\[1.0em]
2200         \TitlePageAuthor{}       \\\[1.0em]
2201         \vfill
2202         \TitlePageFulFulfillment{} \\\[1.0em]
2203         \TitlePageDegree{}       \\\[1.0em]
2204         \TitlePageProgram{}      \\\[1.0em]
2205         \vfill
2206         \TitlePageInstitution{}
2207         \vfill
2208     \end{center}
2209     \TitlePageDefenseDate{}\\[-1em]
2210     \PrintCommitteeMemberList{}

```

```

2211     \cleardoublepage
2212 }
2213
2214
2215
2216
2217

```

## 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.

```

2218 \cs_new:Nn \__UWMad_LicensePage_StartPage: {
2219     \clearpage
2220     \thispagestyle{empty}
2221     \tex_hbox:D{}
2222     \tex_vfill:D
2223     \phantomsection
2224 }
2225 %

```

### 8.2.1 Copyright

```

2226 \bool_new:N \l__UWMad_Copyright_UseCopyright_bool
2227 \cs_set_eq:NN \CopyrightSymbol \copyright
2228
2229 \cs_set:Nn \__UWMad_Copyright_LicenseText: {
2230     \begin{center}
2231         Copyright~\CopyrightSymbol{}~
2232         \l__UWMad_LicensePage_Year_tl{}~
2233         by~
2234         \l__UWMad_LicensePage_Owner_tl{}
2235     \end{center}
2236 }
2237 %

```

```

2238 %
2239 %
2240 %

```

## 8.2.2 Creative Commons

```

2241 %    Token lists
2242 \tl_new:N    \l__UWMad_CCLicense_Porting_tl
2243 \tl_new:N    \l__UWMad_CCLicense_Version_tl
2244 \tl_new:N    \l__UWMad_CCLicense_TypeAbbreviation_tl
2245 \tl_new:N    \l__UWMad_CCLicense_TypeWords_tl
2246 \tl_new:N    \l__UWMad_CCLicense_URL_Front_tl
2247 \tl_new:N    \l__UWMad_CCLicense_URL_Middle_tl
2248 \tl_new:N    \l__UWMad_CCLicense_URL_Back_tl
2249 \tl_new:N    \l__UWMad_CCLicense_URL_tl
2250 \tl_new:N    \l__UWMad_CCLicense_http_tl
2251 \tl_new:N    \l__UWMad_CCLicense_URLText_tl
2252 %
2253 %    Booleans
2254 \bool_new:N \l__UWMad_CCLicense_UseCreativeCommons_bool
2255 \bool_new:N \l__UWMad_CCLicense_UseAttribution_bool
2256 \bool_new:N \l__UWMad_CCLicense_UseShareAlike_bool
2257 \bool_new:N \l__UWMad_CCLicense_UseNoDerivatives_bool
2258 \bool_new:N \l__UWMad_CCLicense_UseNonCommercial_bool
2259 \bool_new:N \l__UWMad_CCLicense_IsValid_bool
2260 \bool_set_true:N \l__UWMad_CCLicense_UseAttribution_bool
2261 %
2262 %    Valid license types
2263 \cs_new:cn {l__UWMad_CCLicense_Valid_by :} {}
2264 \cs_new:cn {l__UWMad_CCLicense_Valid_by-sa :} {}
2265 \cs_new:cn {l__UWMad_CCLicense_Valid_by-nd :} {}
2266 \cs_new:cn {l__UWMad_CCLicense_Valid_by-nc :} {}
2267 \cs_new:cn {l__UWMad_CCLicense_Valid_by-nc-sa :} {}
2268 \cs_new:cn {l__UWMad_CCLicense_Valid_by-nc-nd :} {}
2269 %
2270 %    Defaults
2271 \tl_gset:Nn \l__UWMad_CCLicense_Porting_tl {
2272     International
2273 }
2274 \tl_gset:Nn \l__UWMad_CCLicense_Version_tl {
2275     4.0

```

```

2276 }
2277 %
2278 %   URL definitions
2279 \tl_set:Nn \l__UWMad_CCLicense_URL_Front_tl {
2280     creativecommons.org/licenses
2281 }
2282 \tl_set:Nn \l__UWMad_CCLicense_URL_Middle_tl {
2283     /\l__UWMad_CCLicense_TypeAbbreviation_tl
2284 }
2285 \tl_set:Nn \l__UWMad_CCLicense_URL_Back_tl {
2286     /\l__UWMad_CCLicense_Version_tl
2287 }
2288 \tl_set:Nn \l__UWMad_CCLicense_URL_tl {
2289     http://
2290     \l__UWMad_CCLicense_URL_Front_tl
2291     \l__UWMad_CCLicense_URL_Middle_tl
2292     \l__UWMad_CCLicense_URL_Back_tl
2293 }
2294 \tl_set:Nn \l__UWMad_CCLicense_http_tl {
2295     http://
2296 }
2297 %
2298 %
2299 \tl_set:Nn \l__UWMad_CCLicense_URLText_tl {
2300     Creative~Commons~
2301     \l__UWMad_CCLicense_TypeWords_tl{ }~
2302     \l__UWMad_CCLicense_Version_tl{ }~
2303     \l__UWMad_CCLicense_Porting_tl{ }
2304 }
2305 %
2306 %
2307 %
2308 %   Type Creator
2309 \cs_new:Nn \__UWMad_CCLicense_CreateType: {
2310
2311     \bool_if:NTF \l__UWMad_CCLicense_UseAttribution_bool {
2312
2313         \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
2314             by
2315         }
2316         \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
2317             Attribution
2318         }
2319
2320     } { }
2321

```

```

2322     \bool_if:NTF \l__UWMad_CCLicense_UseNonCommercial_bool {
2323
2324         \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
2325             -nc
2326         }
2327         \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
2328             -NonCommercial
2329         }
2330
2331     } { }
2332
2333     \bool_if:NTF \l__UWMad_CCLicense_UseShareAlike_bool {
2334
2335         \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
2336             -sa
2337         }
2338         \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
2339             -ShareAlike
2340         }
2341
2342     } { }
2343
2344     \bool_if:NTF \l__UWMad_CCLicense_UseNoDerivatives_bool {
2345
2346         \tl_put_right:Nn \l__UWMad_CCLicense_TypeAbbreviation_tl {
2347             -nd
2348         }
2349         \tl_put_right:Nn \l__UWMad_CCLicense_TypeWords_tl {
2350             -NoDerivatives
2351         }
2352
2353     } { }
2354 }
2355 %
2356 %
2357 %
2358 %   Type Validator
2359 \cs_new:Nn \__UWMad_CCLicense_CheckTypeValidity: {
2360     \cs_if_exist:cTF {
2361         l__UWMad_CCLicense_Valid_
2362         \l__UWMad_CCLicense_TypeAbbreviation_tl :
2363     } {
2364
2365         \bool_set_true:N \l__UWMad_CCLicense_IsValid_bool
2366
2367     } {

```

```

2368
2369 \msg_new:nnn {UWMadThesis} {CCLicense / InvalidLicenseType} {
2370   The~license~type~`\l__UWMad_CCLicense_TypeAbbreviation_tl'~
2371   is~not~a~valid~Creative~Commons~license.
2372 }
2373 \msg_error:nn {UWMadThesis} {CCLicense / InvalidLicenseType}
2374
2375 }
2376 }
2377 %
2378 %
2379 %
2380 % Page Printer
2381 \cs_new:Nn \__UWMad_CCLicense_LicenseText: {
2382   \begin{center}
2383     \setstretch{1.05}
2384     This~work~is~released~under~a~
2385     \href {\l__UWMad_CCLicense_URL_tl} {
2386       \l__UWMad_CCLicense_URLText_tl
2387     }~
2388     license.\\[0.1em]
2389     \l__UWMad_LicensePage_Owner_tl{},~
2390     \l__UWMad_LicensePage_Year_tl{}
2391   \end{center}
2392 }
2393 %

```

### 8.2.3 LicensePage Proper

```

2394 %
2395 \tl_new:N \l__UWMad_LicensePage_Year_tl
2396 \tl_new:N \l__UWMad_LicensePage_Owner_tl
2397 %
2398 \tl_set:Nn \l__UWMad_LicensePage_Owner_tl {
2399   \g__UWMad_ThesisInfo_Author_tl
2400 }
2401 \tl_set:Nn \l__UWMad_LicensePage_Year_tl {
2402   \the\year
2403 }
2404 %
2405 %

```

```

2406 %
2407 \DeclareDocumentEnvironment {LicensePage} { } {
2408 %
2409 %
2410 %
2411     \DeclareDocumentCommand \LicenseOwner { m } {
2412         \tl_set:Nn \l__UWMad_LicensePage_Owner_tl {
2413             ##1
2414         }
2415     }
2416     \DeclareDocumentCommand \TheLicenseOwner { } {
2417         \l__UWMad_LicensePage_Owner_tl
2418     }
2419 %
2420     \DeclareDocumentCommand \LicenseYear { m } {
2421         \tl_set:Nn \l__UWMad_LicensePage_Year_tl {
2422             ##1
2423         }
2424     }
2425     \DeclareDocumentCommand \TheLicenseYear { } {
2426         \l__UWMad_LicensePage_Year_tl
2427     }
2428 %
2429 %
2430 %
2431 \DeclareDocumentCommand \Copyright { } {
2432     \bool_set_true:N \l__UWMad_Copyright_UseCopyright_bool
2433 }
2434 \cs_set_eq:NN \AllRightsReserved \Copyright
2435 %
2436 %
2437 %
2438 % User front ends
2439 \DeclareDocumentCommand \CreativeCommons { } {
2440     \bool_set_true:N \l__UWMad_CCLicense_UseCreativeCommons_bool
2441 }
2442 \DeclareDocumentCommand \Attribution { } {
2443     \bool_set_true:N \l__UWMad_CCLicense_UseAttribution_bool
2444 }
2445 \DeclareDocumentCommand \NonCommercial { } {
2446     \bool_set_true:N \l__UWMad_CCLicense_UseNonCommercial_bool
2447 }
2448 \DeclareDocumentCommand \ShareAlike { } {
2449     \bool_set_true:N \l__UWMad_CCLicense_UseShareAlike_bool
2450 }
2451 \DeclareDocumentCommand \NoDerivs { } {

```

```

2452     \bool_set_true:N \l__UWMad_CCLicense_UseNoDerivatives_bool
2453 }
2454 %
2455 %
2456 \DeclareDocumentCommand \CCVersion { m } {
2457     \tl_set:Nn \l__UWMad_CCLicense_Version_tl {##1}
2458 }
2459 %
2460 \DeclareDocumentCommand \CCPorting { m } {
2461     \tl_set:Nn \l__UWMad_CCLicense_Porting_tl {##1}
2462 }
2463 %
2464 \DeclareDocumentCommand \CCURL { m } {
2465     \tl_set:Nn \l__UWMad_CCLicense_URL_Front_tl {##1}
2466     \tl_set:Nn \l__UWMad_CCLicense_URL_Middle_tl {/.}
2467     \tl_set:Nn \l__UWMad_CCLicense_URL_Back_tl {}
2468 }
2469 %
2470 \DeclareDocumentCommand \CCURLText { m } {
2471     \tl_set:Nn \l__UWMad_CCLicense_URLText_tl {##1}
2472 }
2473 %
2474 %
2475 } {
2476
2477     \bool_if:nTF {
2478         \l__UWMad_CCLicense_UseCreativeCommons_bool &&
2479         \l__UWMad_Copyright_UseCopyright_bool
2480     } {
2481         \msg_new:nnn { UWMadThesis } { SpecialPages / MultipleLicenses } {
2482             Both~Creative~Commons~and~Copyright~have~been~declared.~
2483             Please,~pick~one.
2484         }
2485         \msg_error:nn { UWMadThesis } { SpecialPages / MultipleLicenses }
2486     } { }
2487
2488
2489
2490     \bool_if:nTF \l__UWMad_CCLicense_UseCreativeCommons_bool {
2491
2492         \__UWMad_CCLicense_CreateType:
2493         \__UWMad_CCLicense_CheckTypeValidity:
2494         \bool_if:nTF \l__UWMad_CCLicense_IsValid_bool {
2495             \cs_new_eq:NN
2496                 \__UWMad_LicensePage_LicenseText:
2497                 \__UWMad_CCLicense_LicenseText:

```



```

2498         } { }
2499
2500     } { }
2501
2502
2503
2504     \bool_if:NTF \l__UWMad_Copyright_UseCopyright_bool {
2505         \cs_new_eq:NN
2506             \__UWMad_LicensePage_LicenseText:
2507             \__UWMad_Copyright_LicenseText:
2508     } { }
2509
2510
2511
2512     \cs_if_exist:NTF \__UWMad_LicensePage_LicenseText: {
2513         \__UWMad_LicensePage_StartPage:
2514         \vbox_to_ht:nn {0.3333\textheight} {
2515             \__UWMad_LicensePage_LicenseText:
2516         }
2517     } { }
2518
2519
2520 }
2521 %

```

## Module 9

# Relative Directory Input

## 9.1 Declarations and Initializations

Variable declarations and default initializations for Chapter directories.

```

2522 \int_new:N    \g__UWMad_RelativeDirectory_Chapter_Count_int
2523 \tl_new:N     \g__UWMad_RelativeDirectory_Chapter_Prefix_tl
2524 \tl_new:N     \g__UWMad_RelativeDirectory_Chapter_Suffix_tl
2525 \tl_new:N     \g__UWMad_RelativeDirectory_Chapter_CurrentPath_tl
2526 \tl_new:N     \g__UWMad_RelativeDirectory_Chapter_CurrentName_tl
2527 \tl_new:N     \g__UWMad_RelativeDirectory_Chapter_ParentPath_tl
2528 \tl_gset:Nn   \g__UWMad_RelativeDirectory_Chapter_ParentPath_tl {}

```

Variable declarations and default initializations for Section directories.

```

2529 \int_new:N    \g__UWMad_RelativeDirectory_Section_Count_int
2530 \tl_new:N     \g__UWMad_RelativeDirectory_Section_Prefix_tl
2531 \tl_new:N     \g__UWMad_RelativeDirectory_Section_Suffix_tl
2532 \tl_new:N     \g__UWMad_RelativeDirectory_Section_CurrentPath_tl
2533 \tl_new:N     \g__UWMad_RelativeDirectory_Section_CurrentName_tl
2534 \tl_new:N     \g__UWMad_RelativeDirectory_Section_ParentPath_tl
2535 \tl_gset:Nn   \g__UWMad_RelativeDirectory_Section_ParentPath_tl {
2536     \g__UWMad_RelativeDirectory_Chapter_CurrentPath_tl/
2537 }

```

Variable declarations and default initializations for Subsection directories.

```

2538 \int_new:N    \g__UWMad_RelativeDirectory_Subsection_Count_int
2539 \tl_new:N     \g__UWMad_RelativeDirectory_Subsection_Prefix_tl
2540 \tl_new:N     \g__UWMad_RelativeDirectory_Subsection_Suffix_tl
2541 \tl_new:N     \g__UWMad_RelativeDirectory_Subsection_CurrentPath_tl

```

```

2542 \tl_new:N    \g__UWMad_RelativeDirectory_Subsection_CurrentName_tl
2543 \tl_new:N    \g__UWMad_RelativeDirectory_Subsection_ParentPath_tl
2544 \tl_gset:Nn \g__UWMad_RelativeDirectory_Subsection_ParentPath_tl {
2545     \g__UWMad_RelativeDirectory_Section_CurrentPath_tl/
2546 }

```

Variable declaration for graphics inclusion

```

2547 \tl_new:N    \g__UWMad_RelativeDirectory_Graphics_DirectoryName_tl
2548 \tl_new:N    \g__UWMad_RelativeDirectory_Graphics_Extension_tl
2549 \tl_new:N    \g__UWMad_RelativeDirectory_Graphics_BaseName_tl

```

Variable declarations for search options.

```

2550 \bool_new:N \g__UWMad_RelativeDirectory_CycleThrough_Graphics_bool
2551 \bool_new:N \g__UWMad_RelativeDirectory_CycleThrough_Files_bool

```

Miscellaneous variable initializations for the system

```

2552 \tl_new:N    \g__UWMad_RelativeDirectory_File_CurrentName_tl
2553 \tl_new:N    \g__UWMad_RelativeDirectory_OptionalPath_tl
2554 \seq_new:N    \g__UWMad_RelativeDirectory_PathStack_Files_seq
2555 \seq_new:N    \g__UWMad_RelativeDirectory_PathStack_Graphics_seq
2556 \bool_new:N    \g__UWMad_RelativeDirectory_IsFileFound_bool

```

Miscellaneous control sequence initializations for the system.

```

2557 \cs_new:Nn \UWMad_RelativeDirectory_Chapter_SetName: {}
2558 \cs_new:Nn \UWMad_RelativeDirectory_Section_SetName: {}
2559 \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.

```

2560 \cs_new:Nn \UWMad_RelativeDirectory_SetName_Increment_Hook_Chapter: {
2561     \int_gset:cn {g__UWMad_RelativeDirectory_Section_Count_int}{0}
2562     \int_gset:cn {g__UWMad_RelativeDirectory_Subsection_Count_int}{0}
2563 }
2564 \cs_new:Nn \UWMad_RelativeDirectory_SetName_Increment_Hook_Section: {
2565     \int_gset:cn {g__UWMad_RelativeDirectory_Subsection_Count_int}{0}
2566 }
2567 \cs_new:Nn \UWMad_RelativeDirectory_SetName_Increment_Hook_Subsection: {}

```

Directory name-setting functions.

```

2568 \cs_new:Nn \UWMad_RelativeDirectory_SetName_None:n {
2569     \tl_gset:cx {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl} {
2570         \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Prefix_tl}
2571         \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Suffix_tl}
2572     }
2573 }
2574 \cs_new:Nn \UWMad_RelativeDirectory_SetName_Increment:n {
2575     \use:c{UWMad_RelativeDirectory_SetName_Increment_Hook_ #1 :}
2576     \int_gincr:c {g__UWMad_RelativeDirectory_ #1 _Count_int}
2577     \tl_gset:cx {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl} {
2578         \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Prefix_tl}
2579         \int_to_arabic:n{
2580             \int_use:c{g__UWMad_RelativeDirectory_ #1 _Count_int}
2581         }
2582         \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Suffix_tl}
2583     }
2584 }
2585 \cs_new:Nn \UWMad_RelativeDirectory_SetName_Same:n {
2586     \tl_gset:cx {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl} {
2587         \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Prefix_tl}
2588         \g__UWMad_RelativeDirectory_File_CurrentName_tl
2589         \tl_use:c {g__UWMad_RelativeDirectory_ #1 _Suffix_tl}
2590     }
2591 }

```

Name and path setter.

```

2592 \cs_new:Nn \UWMad_RelativeDirectory_SetNameAndPath:n {
2593
2594     \tl_gclear:c {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}

```

```

2595 \tl_gclear:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
2596
2597 \tl_if_blank:VTF {\g__UWMad_RelativeDirectory_OptionalPath_tl} {
2598   \use:c {UWMad_RelativeDirectory_ #1 _SetName:}
2599 } {
2600   \tl_gset_eq:cN
2601     {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
2602     \g__UWMad_RelativeDirectory_OptionalPath_tl
2603 }
2604 \tl_gset:cx {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl} {
2605   \tl_use:c {g__UWMad_RelativeDirectory_ #1 _ParentPath_tl}
2606   \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
2607 }
2608 }

```

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.

```

2609 \cs_new:Nn \__UWMad_RelativeDirectory_StackPush_Default:n {
2610   \tl_gset_eq:Nc
2611     \g_tmpa_tl
2612     {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
2613   \tl_if_blank:VTF {\g_tmpa_tl} { } {
2614     \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Files_seq {
2615       \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
2616     }
2617     \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Graphics_seq {
2618       \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
2619     }
2620   }
2621 }
2622 \cs_new:Nn \__UWMad_RelativeDirectory_StackPush_Files:n {
2623   \tl_gset_eq:Nc
2624     \g_tmpa_tl
2625     {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
2626   \tl_if_blank:VTF {\g_tmpa_tl} { } {
2627     \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Files_seq {
2628       \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
2629     }
2630   }
2631 }

```

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.

```

2632 \cs_new:Nn \__UWMad_RelativeDirectory_StackPush:n {
2633     \__UWMad_RelativeDirectory_StackPush_Default:n{#1}
2634 }

```

Pre-stack update functions for the supported sections.

```

2635 \cs_new:Nn \UWMad_RelativeDirectory_UpdateStack_Chapter_PreHook: {
2636     \seq_gclear:N \g__UWMad_RelativeDirectory_PathStack_Files_seq
2637     \seq_gclear:N \g__UWMad_RelativeDirectory_PathStack_Graphics_seq
2638 }
2639 \cs_new:Nn \UWMad_RelativeDirectory_UpdateStack_Section_PreHook: {}
2640 \cs_new:Nn \UWMad_RelativeDirectory_UpdateStack_Subsection_PreHook: {}

```

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

```

2641 \cs_new:Nn \UWMad_RelativeDirectory_UpdateStack:n {
2642     \use:c {UWMad_RelativeDirectory_UpdateStack_ #1 _PreHook:}
2643     \UWMad_RelativeDirectory_SetNameAndPath:n{#1}
2644     \__UWMad_RelativeDirectory_StackPush:n{#1}
2645 }

```

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.

```

2646 \cs_new:Nn \UWMad_RelativeDirectory_IncludeFile_CycleThrough: {
2647     \seq_map_inline:Nn \g__UWMad_RelativeDirectory_PathStack_Files_seq {
2648         \tl_gset:Nx \g_tmpa_tl {
2649             ./##1/
2650             \g__UWMad_RelativeDirectory_File_CurrentName_tl
2651         }
2652         \bool_if:NTF \g__UWMad_RelativeDirectory_IsFileFound_bool { } {
2653             \file_if_exist:nTF { \g_tmpa_tl } {
2654                 \file_input:n{ \g_tmpa_tl }
2655                 \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
2656                 \seq_map_break:
2657             } { }
2658         }
2659     }
2660 }
2661 \cs_new:Nn \UWMad_RelativeDirectory_IncludeFile_CheckDeepest: {

```

```

2662
2663 \seq_get:NN
2664   \g__UWMad_RelativeDirectory_PathStack_Files_seq
2665   \g_tmpa_tl
2666 \tl_gset:Nx \g_tmpa_tl {
2667   ./\g_tmpa_tl/
2668   \g__UWMad_RelativeDirectory_File_CurrentName_tl
2669 }
2670 \file_if_exist:nTF {\g_tmpa_tl} {
2671   \file_input:n{\g_tmpa_tl}
2672   \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
2673 } { }
2674 }

```

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  $\text{\TeX}$  directory for the file and issue a warning if it is not found.

```

2675 \msg_new:nnn { UWMadThesis }{ RelativeDirectory / FileNotFound } {
2676   The~requested~file~'#1'~was~not~found~in~the~current~search~stack~nor~the~
2677   main~LaTeX~directory~for~the~job~'\c_job_name_tl'.
2678 }
2679 \cs_new:Nn \UWMad_RelativeDirectory_IncludeFile: {
2680   \bool_gset_false:N \g__UWMad_RelativeDirectory_IsFileFound_bool
2681
2682   \bool_if:NTF \g__UWMad_RelativeDirectory_CycleThrough_Files_bool {
2683     \UWMad_RelativeDirectory_IncludeFile_CycleThrough:
2684   } {
2685     \UWMad_RelativeDirectory_IncludeFile_CheckDeepest:
2686   }
2687   \bool_if:NTF \g__UWMad_RelativeDirectory_IsFileFound_bool { } {
2688     \file_if_exist:nTF {\g__UWMad_RelativeDirectory_File_CurrentName_tl} {
2689       \file_input:n{ \g__UWMad_RelativeDirectory_File_CurrentName_tl }
2690       \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
2691     } {
2692       \msg_warning:nnx
2693         { UWMadThesis }
2694         { RelativeDirectory / FileNotFound }
2695         { \g__UWMad_RelativeDirectory_File_CurrentName_tl }
2696     }
2697   }
2698 }

```

### 9.2.2 Graphics Inclusion

This code copies the existing `\includegraphics` command such that it can be used in a compatible way with the  $\text{\LaTeX} 2_{\varepsilon}$  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.

```

2699 \cs_new_eq:NN
2700     \__UWMad_RelativeDirectory_IncludeGraphics_Original:n
2701     \includegraphics
2702 \cs_undefine:N
2703     \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.

```

2704 \cs_new:Nn \__UWMad_RelativeDirectory_StackPush_FilesAndGraphics:n {
2705     \tl_gset_eq:Nc
2706         \g_tmpa_tl
2707         {g__UWMad_RelativeDirectory_ #1 _CurrentName_tl}
2708     \tl_if_blank:VTF {\g_tmpa_tl} { } {
2709         \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Files_seq {
2710             \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
2711         }
2712         \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Graphics_seq {
2713             \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}
2714         }
2715         \seq_gpush:Nx \g__UWMad_RelativeDirectory_PathStack_Graphics_seq {
2716             \tl_use:c {g__UWMad_RelativeDirectory_ #1 _CurrentPath_tl}/
2717             \g__UWMad_RelativeDirectory_Graphics_DirectoryName_tl
2718         }
2719     }
2720 }

```

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.

```

2721 \cs_new:Nn \UWMad_RelativeDirectory_IncludeGraphics_CycleThrough:n {
2722
2723   \UWMad_File_GetExtension:nNN
2724     {\g__UWMad_RelativeDirectory_File_CurrentName_tl}
2725     \g__UWMad_RelativeDirectory_Graphics_BaseName_tl
2726     \g__UWMad_RelativeDirectory_Graphics_Extension_tl
2727
2728   \seq_map_inline:Nn \g__UWMad_RelativeDirectory_PathStack_Graphics_seq {
2729
2730     \tl_gset:Nx \g_tmpa_tl {
2731       ./##1/
2732       \g__UWMad_RelativeDirectory_File_CurrentName_tl
2733     }
2734
2735     \bool_if:NTF \g__UWMad_RelativeDirectory_IsFileFound_bool { } {
2736       \file_if_exist:nTF { \g_tmpa_tl } {
2737         \tl_gset:Nx \g_tmpa_tl {
2738           ./##1/
2739           \g__UWMad_RelativeDirectory_Graphics_BaseName_tl
2740         }
2741         \__UWMad_RelativeDirectory_IncludeGraphics_Original:nn
2742           [ #1 ] {\g_tmpa_tl}
2743         \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
2744         \seq_map_break:
2745       } { }
2746     }
2747   }
2748 }
2749 \cs_new:Nn \UWMad_RelativeDirectory_IncludeGraphics_CheckDeepest:n {
2750
2751   \seq_get:NN
2752     \g__UWMad_RelativeDirectory_PathStack_Graphics_seq
2753     \g_tmpa_tl
2754
2755   \tl_gset:Nx \g_tmpb_tl {
2756     ./\g_tmpa_tl/
2757     \g__UWMad_RelativeDirectory_File_CurrentName_tl
2758   }
2759
2760   \UWMad_File_GetExtension:nNN {\g_tmpb_tl}
2761     \g__UWMad_RelativeDirectory_Graphics_BaseName_tl
2762     \g__UWMad_RelativeDirectory_Graphics_Extension_tl
2763
2764   \file_if_exist:nTF { \g_tmpb_tl } {

```

```

2765     \__UWMad_RelativeDirectory_IncludeGraphics_Original:nn
2766     [ #1 ]
2767     { \g__UWMad_RelativeDirectory_Graphics_BaseName_tl }
2768     \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
2769   } { }
2770 }

```

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  $\text{\TeX}$  directory and issue a warning if it is still not found.

```

2771 \msg_new:nnn { UWMadThesis }{ RelativeDirectory / GraphicNotFound } {
2772   The~requested~graphic~'#1'~was~not~found~in~the~current~search~stack~nor~
2773   the~main~LaTeX~directory~for~the~job~'\c_job_name_tl'.
2774 }
2775 \cs_new:Nn \UWMad_RelativeDirectory_IncludeGraphics:n {
2776   \bool_gset_false:N \g__UWMad_RelativeDirectory_IsFileFound_bool
2777   \bool_if:NTF \g__UWMad_RelativeDirectory_CycleThrough_Graphics_bool {
2778     \UWMad_RelativeDirectory_IncludeGraphics_CycleThrough:n{#1}
2779   } {
2780     \UWMad_RelativeDirectory_IncludeGraphics_CheckDeepest:n{#1}
2781   }
2782   \bool_if:NTF \g__UWMad_RelativeDirectory_IsFileFound_bool { } {
2783     \file_if_exist:nTF {\g__UWMad_RelativeDirectory_File_CurrentName_tl} {
2784       \__UWMad_RelativeDirectory_IncludeGraphics_Original:nn
2785       [ #1 ]
2786       { \g__UWMad_RelativeDirectory_File_CurrentName_tl }
2787       \bool_gset_true:N \g__UWMad_RelativeDirectory_IsFileFound_bool
2788     } {
2789       \msg_warning:nxx
2790         { UWMadThesis }
2791         { RelativeDirectory / GraphicNotFound }
2792         { \g__UWMad_RelativeDirectory_File_CurrentName_tl }
2793     }
2794   }
2795 }

```

### 9.2.3 Key-Value Option Definitions

Being the key definitions

```
2796 \keys_define:nn { UWMadThesis / RelativeDirectory } {
```

Chapter prefix and suffix keys.

```
2797     chapter-directory-prefix      .tl_gset:N =
2798         \g__UWMad_RelativeDirectory_Chapter_Prefix_tl,
2799     chapter-directory-prefix      .default:n =,
2800     chapter-directory-suffix      .tl_gset:N =
2801         \g__UWMad_RelativeDirectory_Chapter_Suffix_tl,
2802     chapter-directory-suffix      .default:n =,
```

Chapter naming conventions

```
2803     chapter-directory-name        .choice:,
2804     chapter-directory-name / none .code:n = {
2805         \cs_gset:Nn \UWMad_RelativeDirectory_Chapter_SetName: {
2806             \UWMad_RelativeDirectory_SetName_None:n{Chapter}
2807         }
2808     },
2809     chapter-directory-name / same .code:n = {
2810         \cs_gset:Nn \UWMad_RelativeDirectory_Chapter_SetName: {
2811             \UWMad_RelativeDirectory_SetName_Same:n{Chapter}
2812         }
2813     },
2814     chapter-directory-name / increment .code:n = {
2815         \cs_gset:Nn \UWMad_RelativeDirectory_Chapter_SetName: {
2816             \UWMad_RelativeDirectory_SetName_Increment:n{Chapter}
2817         }
2818     },
2819     chapter-directory-name        .default:n = none,
```

Section prefix and suffix keys.

```
2820     section-directory-prefix      .tl_gset:N =
2821         \g__UWMad_RelativeDirectory_Section_Prefix_tl,
2822     section-directory-prefix      .default:n =,
2823     section-directory-suffix      .tl_gset:N =
2824         \g__UWMad_RelativeDirectory_Section_Suffix_tl,
2825     section-directory-suffix      .default:n =,
```

Section naming conventions

```

2826 section-directory-name .choice:,
2827 section-directory-name / none .code:n = {
2828     \cs_gset:Nn \UWMad_RelativeDirectory_Section_SetName: {
2829         \UWMad_RelativeDirectory_SetName_None:n{Section}
2830     }
2831 },
2832 section-directory-name / same .code:n = {
2833     \cs_gset:Nn \UWMad_RelativeDirectory_Section_SetName: {
2834         \UWMad_RelativeDirectory_SetName_Same:n{Section}
2835     }
2836 },
2837 section-directory-name / increment .code:n = {
2838     \cs_gset:Nn \UWMad_RelativeDirectory_Section_SetName: {
2839         \UWMad_RelativeDirectory_SetName_Increment:n{Section}
2840     }
2841 },
2842 section-directory-name .default:n = none,

```

Subsection prefix and suffix keys.

```

2843 subsection-directory-prefix .tl_gset:N =
2844     \g__UWMad_RelativeDirectory_Subsection_Prefix_tl,
2845 subsection-directory-prefix .default:n =,
2846 subsection-directory-suffix .tl_gset:N =
2847     \g__UWMad_RelativeDirectory_Subsection_Suffix_tl,
2848 subsection-directory-suffix .default:n =,

```

Subsection naming conventions

```

2849 subsection-directory-name .choice:,
2850 subsection-directory-name / none .code:n = {
2851     \cs_gset:Nn \UWMad_RelativeDirectory_Subsection_SetName: {
2852         \UWMad_RelativeDirectory_SetName_None:n{Subsection}
2853     }
2854 },
2855 subsection-directory-name / same .code:n = {
2856     \cs_gset:Nn \UWMad_RelativeDirectory_Subsection_SetName: {
2857         \UWMad_RelativeDirectory_SetName_Same:n{Subsection}
2858     }
2859 },
2860 subsection-directory-name / increment .code:n = {
2861     \cs_gset:Nn \UWMad_RelativeDirectory_Subsection_SetName: {
2862         \UWMad_RelativeDirectory_SetName_Increment:n{Subsection}
2863     }
2864 },

```

```
2865 subsection-directory-name .default:n = none,
```

Graphics directory keys.

```
2866 graphics-directory-name .code:n = {
2867   \tl_gset:Nn \g__UWMad_RelativeDirectory_Graphics_DirectoryName_tl {
2868     #1
2869   }
2870   \tl_if_blank:nTF { #1 } {
2871     \cs_gset:Nn \__UWMad_RelativeDirectory_StackPush:n {
2872       \__UWMad_RelativeDirectory_StackPush_Default:n{##1}
2873     }
2874   } {
2875     \cs_gset:Nn \__UWMad_RelativeDirectory_StackPush:n {
2876       \__UWMad_RelativeDirectory_StackPush_FilesAndGraphics:n{##1}
2877     }
2878   }
2879 },
2880 the-only-graphics-directory .code:n = {
2881   \bool_set_false:N
2882     \g__UWMad_RelativeDirectory_CycleThrough_Graphics_bool
2883   \seq_gclear:N \g__UWMad_RelativeDirectory_PathStack_Graphics_seq
2884   \seq_gpush:Nn \g__UWMad_RelativeDirectory_PathStack_Graphics_seq {
2885     #1
2886   }
2887   \cs_gset:Nn \UWMad_RelativeDirectory_UpdateStack_Chapter_PreHook: {
2888     \seq_gclear:N \g__UWMad_RelativeDirectory_PathStack_Files_seq
2889   }
2890   \cs_gset:Nn \__UWMad_RelativeDirectory_StackPush:n {
2891     \__UWMad_RelativeDirectory_StackPush_Files:n{##1}
2892   }
2893 },
```

Path search keys.

```
2894 cycle-file-paths .bool_gset:N =
2895   \g__UWMad_RelativeDirectory_CycleThrough_Files_bool,
2896 cycle-file-paths .default:n = false,
2897 cycle-graphic-paths .bool_gset:N =
2898   \g__UWMad_RelativeDirectory_CycleThrough_Graphics_bool,
2899 cycle-graphic-paths .default:n = true
2900 }
```

Set the default values for the keys.

```

2901 \keys_set:nn { UWMadThesis / RelativeDirectory } {
2902     chapter-directory-prefix,
2903     chapter-directory-suffix,
2904     section-directory-prefix,
2905     section-directory-suffix,
2906     subsection-directory-prefix,
2907     subsection-directory-suffix,
2908     chapter-directory-name,
2909     section-directory-name,
2910     subsection-directory-name,
2911     cycle-file-paths,
2912     cycle-graphic-paths
2913 }

```

### 9.3 User Front Ends

```

2914 \DeclareDocumentCommand \IncludeChapter { o m } {
2915     \IfValueTF { #1 } {
2916         \tl_gset:Nn \g__UWMad_RelativeDirectory_OptionalPath_tl {#1}
2917     } { }
2918     \tl_gset:Nn \g__UWMad_RelativeDirectory_File_CurrentName_tl {#2}
2919     \UWMad_RelativeDirectory_UpdateStack:n{Chapter}
2920     \UWMad_RelativeDirectory_IncludeFile:
2921     \tl_gclear:N \g__UWMad_RelativeDirectory_OptionalPath_tl
2922 }
2923 \DeclareDocumentCommand \IncludeSection { o m } {
2924     \IfValueTF { #1 } {
2925         \tl_gset:Nn \g__UWMad_RelativeDirectory_OptionalPath_tl {#1}
2926     } { }
2927     \tl_gset:Nn \g__UWMad_RelativeDirectory_File_CurrentName_tl {#2}
2928     \UWMad_RelativeDirectory_UpdateStack:n{Section}
2929     \UWMad_RelativeDirectory_IncludeFile:
2930     \tl_gclear:N \g__UWMad_RelativeDirectory_OptionalPath_tl
2931 }
2932 \DeclareDocumentCommand \IncludeSubsection { o m } {
2933     \IfValueTF { #1 } {
2934         \tl_gset:Nn \g__UWMad_RelativeDirectory_OptionalPath_tl {#1}
2935     } { }
2936     \tl_gset:Nn \g__UWMad_RelativeDirectory_File_CurrentName_tl {#2}
2937     \UWMad_RelativeDirectory_UpdateStack:n{Subsection}
2938     \UWMad_RelativeDirectory_IncludeFile:

```

```

2939     \tl_gclear:N \g__UWMad_RelativeDirectory_OptionalPath_tl
2940 }
2941 \DeclareDocumentCommand \IncludeGraphics { o m } {
2942     \tl_gset:Nn \g__UWMad_RelativeDirectory_File_CurrentName_tl {#2}
2943     \IfValueTF { #1 } {
2944         \UWMad_RelativeDirectory_IncludeGraphics:n{#1}
2945     } {
2946         \UWMad_RelativeDirectory_IncludeGraphics:n{}
2947     }
2948 }
2949 \cs_new_eq:NN
2950     \includegraphics
2951     \IncludeGraphics

2952 \ExplSyntaxOff

```

# Change History

1.0

General: Hello ..... [1](#)