

Writing NREL documents using LaTeX

A. Clifton, A. Platt, P. Fleming, M. Lawson

Monday 17th August, 2015

Executive Summary

This document is a guide to writing documents for publication by NREL using the LaTeX document preparation system. LaTeX is not WYSIWYG and has different reviewing and editing tools compared to typical word processing software. For this reason special care has to be taken when preparing NREL documents in LaTeX. This document serves both as a guide to implementing NREL's style and formatting guidelines in LaTeX, and as a template. This document is intended for people with some familiarity with LaTeX.

Acknowledgments

This document and the NREL LaTeX class file were developed by staff at the National Wind Technology Center, including Andrew Platt, Andrew Clifton, Andrew Ning, Mike Lawson, and Paul Fleming. Alexandra Lemke provided support relating to NREL communications. A first demonstration of an NREL class file was created by Chuck Booten from NREL's Electricity, Resources, and Building Systems Integration group, which inspired this effort. The class file and this template were developed as part of work on several NREL reports, journal articles, and conference publications.

We thank members of the TeX – LaTeX StackExchange site for useful suggestions concerning LaTeX and typography Anon. 2014.

This report was typeset using the LaTeX typesetting system originally developed by Leslie Lamport, based on TeX created by Donald Knuth.

Table of Contents

1 Requirements for NREL documents	3
1.1 NREL style guide	3
1.2 Formatting	3
2 Using LaTeX to make documents that meet NREL's requirements	4
2.1 What is LaTeX?	4
2.2 General Process	4
2.3 The NREL LaTeX style file	4
2.3.1 Getting nrel.cls	4
2.3.2 Installing nrel.cls	5
2.3.3 Using nrel.cls	5
2.3.4 Options in nrel.cls	5
2.3.5 Classes and packages in nrel.cls	5
2.3.6 PDF settings	6
2.4 Front, main, and back matter	6
2.5 Citations	7
2.6 NREL-style bibliographies	7
2.7 Putting it all together	8
2.8 Best practice in writing a document in LaTeX	8
3 Some LaTeX examples	10
3.1 Headings	10
3.1.1 Chapter	10
3.1.2 Sections	10
3.2 Body text	10
3.3 Mathematics	11
3.4 Cross references	11
3.5 Floats	11
3.5.1 Tables	11
3.5.2 Figures	12
3.5.3 Subfigures	12
3.6 Lists	12
3.7 Computer code	13
3.8 Creating a file structure	13
4 Preparing a .DOC or .DOCX file from LaTeX	14
4.1 The conversion process	14
4.2 Using latex2rtf to convert files	14
4.2.1 Using latex2rtf	14
4.2.2 Using latex2rtf and LaTeX together	15
4.2.3 Indexes	15
4.2.4 What to do when the conversion to rich text format fails	15
4.3 Using pandoc	16

5	Preparing a high-quality PDF from LaTeX	17
5.1	PDF tagging	17
5.2	Alt-text on images and equations	17
5.3	Embedded fonts	19

List of Figures

Figure 1.	NREL images	13
Figure 2.	NREL images	18

List of Tables

Table 1.	NREL's process for producing and reviewing LaTeX files	4
Table 2.	Packages supported by the nrel.cls class	6
Table 3.	An example table.	12
Table 4.	A comparison of latex2rtf and pandoc for document conversion.	14

1 Requirements for NREL documents

There are well-defined requirements for all documents that are published by NREL.

1.1 NREL style guide

The NREL in-house style is described at <http://www.nrel.gov/extranet/communications/styleguide.html>. This details the conventions that should be used when writing NREL documents.

1.2 Formatting

NREL publishes templates for reports and other technical documents. These are designed to be used with most common WYSIWYG programs and latex. Templates are posted online at <http://www.nrel.gov/extranet/communications/report-template.html> and updated regularly.

2 Using LaTeX to make documents that meet NREL's requirements

2.1 What is LaTeX?

LaTeX is a mark-up language that describes how a document should be prepared. Three things are needed to make a LaTeX document:

1. A source document, usually with extension *.tex*
2. Some packages and classes that help turn what's in the source document into something helpful
3. A compiler, also referred to as a working LaTeX installation.

At first glance the source document looks like a programming language, and that's because it is: LaTeX is not WYSIWYG, like many of the document preparation tools in common use today. A good analogy is html.

The wikibook at <http://en.wikibooks.org/wiki/LaTeX> is an excellent resource. There are also several internet forums such as tex.stackexchange.com that may be useful.

2.2 General Process

An outline of the process for producing NREL documents using LaTeX is given in Table 1. Please note that this process is subject to revision without warning.

Table 1. NREL's process for producing and reviewing LaTeX files

Phase	Lead	Steps	More Information
Draft	Author	<ol style="list-style-type: none">1. Prepare document in LaTeX using the <i>nrel.cls</i> class file2. Prepare PDF3. Convert the tex document to a word processing format (<i>.rfi</i>, <i>.doc</i>, <i>.docx</i>) using a tool such as <i>latex2rtf</i> or <i>pandoc</i>5. Archive all files, including:<ul style="list-style-type: none">• LaTeX source files• Images• Final PDF	Section 2.3 Section 5 Chapter 4
Review	Communications	Review the structure of the PDF Edit the supplied <i>.doc</i> or <i>.docx</i> file using track changes	
Revision	Author	Implements required changes in the LaTeX files. Create the final, tagged and structured PDF	
Publish	Publications	Combine the PDF with the appropriate cover sheet(s)	

2.3 The NREL LaTeX style file

A LaTeX class called *nrel.cls* has been written that implements the NREL formatting requirements in LaTeX.

2.3.1 Getting *nrel.cls*

The current version of *nrel.cls* can be downloaded from https://github.com/NREL/latex_editing. This is a public repository.

2.3.2 Installing *nrel.cls*

Place *nrel.cls* and *nrel bst* in the same directory as the LaTeX files you are trying to compile. This will make the files available to that project, only. This approach can be used on a desktop computer, or on a network drive, or for online collaborative tools such as www.writelatex.com or www.sharelatex.com. Advanced LaTeX users may wish to copy these files to their local tree.

2.3.3 Using *nrel.cls*

To use the class file, insert the following text in the preamble, before `\begin{document}`:

```
\newif\iflatex2rtf

\iflatex2rtf
% tell latex2rtf if this is an article or report
\documentclass[10pt,letterpaper]{report}
\input{NRELLatex2rtf.tex}
\else
\documentclass[draft,report]{nrel}
\fi
```

This tells LaTeX to use the correct class file, and defines a set of commands that will be used by *latextortf* to properly convert the latex to a rich text document for reviewing (see Chapter ??).

2.3.4 Options in *nrel.cls*

The line

```
\documentclass[draft,report]{nrel}
```

specifies the options (inside the square brackets) that will be passed to the *nrel* class. The options include:

book compile the document using the LaTeX *book* class. This is intended for longer documents and allows the use of chapters.

report compile the document using the LaTeX *report* class. This is intended for longer documents and allows the use of chapters.

article compile the document using the LaTeX *article* class. This is intended for shorter documents such as journal articles. This class does not support the use of chapters.

memoir compile the document using the LaTeX *memoir* class. This option is not recommended because of the challenge with later converting to RTF format for communications review.

draft add a ‘draft’ watermark to all pages and colours all links in blue.

10pt, 12pt set the font size accordingly. The default is 12 point.

letterpaper, a4paper set the paper size. the default is letter paper.

2.3.5 Classes and packages in *nrel.cls*

nrel.cls calls a variety of other packages. Packages are codes that modify the appearance or behaviour of LaTeX to achieve something. Table 2 lists the packages that are explicitly called by *nrel.cls* in the order they are called in. These packages often call other packages, so this is not an exhaustive list.

Table 2. Packages supported by nrel.cls. Unless otherwise stated, packages are not supported by latex2rtf.

Packages	options	functionality	latex2rtf support
nag		checks that packages are up to date and looks for bad habits in LaTeX code.	
geometry		sets page size and margins	✓
mathptmx		changes fonts	
helvet		changes fonts	
courier		changes fonts	
amsfonts,		supplies the AMS fonts, which are useful for mathematics	
amssymb			
booktabs			
graphicx		graphics handling, including .eps figures	✓
natbib	sort	handles citations and allows the \cite, \citet and \citet citation commands (see Section 2.5).	✓
fontenc	T1		
xcolor			
babel	english		
subfigure		provides the subfigure environment to produce sub figures	✓
hyphenat			
setspace			
parskip			
toclof	subfigure		
toclifbind	nottoc, notlot, notlof		
todonotes		inline and margin to-do notes	✓('to do' is prefaced with To Do: in the output)
listings			
caption			
pdfcomment		tool-tips. Also calls the package hyper ref	✓(the tool tip is suppressed)
accessibility	tagged	generates the document structure and tagging	

2.3.6 PDF settings

pdfinterwordspaceon Turns on inter-word spacing in PDFs. Requires TexLive 2014.

pdfminorversion=8 Sets the PDF version.

2.4 Front, main, and back matter

NREL's convention is to have Roman numerals in the front matter, and then arabic numerals in the main matter of the document (after the tables of contents, figures and tables). Tables and figures in the front matter are also numbered differently (Table A, B, C, ...) than in the main matter (Table 1, 2, 3, ...).

This change in page and float numbering is implemented using the \frontmatter, \mainmatter, and \backmatter commands in the document:

```
\begin{document}

\maketitle
\frontmatter
...
\renewcommand{\contentsname}{Table of Contents}
\tableofcontents
\clearpage
\listoffigures
\listoftables
\mainmatter
...
\backmatter
\end{document}
```

Page numbering in the front matter (i.e. the Abstract, Summary, and Foreword chapters or sections) starts at page 3 to allow for NREL cover pages.

If you don't use the `\frontmatter` commands, you may need to increment the page counter manually. To increment the counter n pages, use `\setcounter{page}{n}` after `\begin{document}`.

2.5 Citations

Use bibtex to organize references and store them in a single file (e.g. `/Documents/bibliography/bibliography.bib`). The bibliography will then contain entries with 'keys', like `Lamport_1986_a`. Authors can then insert citations to this key throughout their document, using different styles of citation. These are described in the `achicago.sty` documentation.

- `\cite{Lamport_1986_a}` prints Lamport 1986.
- `\citep{Lamport_1986_a}` prints (Lamport 1986).
- `\citet{Lamport_1986_a}` prints Lamport (1986).

To cite URLs, use the 'misc' style. For example, the bibtex entry for <http://tex.stackexchange.com> Anon. 2014 looks like this:

```
@misc{texstackexchange ,
  Author = {Anon.},
  Howpublished = {Accessed July 21, 2014: \url{http://tex.stackexchange.com}},
  Title = {\TeX — LaTeX Stack Exchange},
  Year = {2014}}
```

This format will allow you to include the date on which a URL was accessed.

The citations should work with journal articles, books (Lamport 1986), technical reports (Other and Nother 2014), and URLs (Anon. 2014).

2.6 NREL-style bibliographies

NREL uses "Chicago A" style-references. The `nrel.cls` file uses Biblatex to produce these references. To include a bibliography in the document, use the following commands where you want the bibliography to occur:

```
\bibliography{files/bibliography.bib}
...
\begin{document}
...
\cleardoublepage
\label{sec:Bib}
\printbibliography
...
\end{document}
```

2.7 Putting it all together

The source of your LaTeX document should probably look like this:

```
\newif\iflatex2rtf
\iflatex2rtf
% tell latex2rtf if this is an article or report
\documentclass[10pt,letterpaper]{report}
\input{NRELLatex2rtf.tex}
\else
\documentclass[draft,report]{nrel}
\fi
\title{Writing NREL documents using LaTeX}
\author{A. Clifton, A. Platt, P. Fleming, M. Lawson}
\bibliography{files/bibliography.bib}
\begin{document}
\maketitle
\frontmatter
\input{ExecSummary}
\tableofcontents
\clearpage
\listoffigures
\listoftables
\mainmatter
\input{Introduction}
\input{Theory}
\cleardoublepage
\label{sec:Bib}
\printbibliography

\end{document}
```

2.8 Best practice in writing a document in LaTeX

Create a structure before you get too far. Authors will find it easier to write documents and make changes if they separate the content of the document from the structure.

1. Each new LaTeX document should be placed in its own directory.
2. Create a main LaTeX file that just contains the preamble, custom commands and uses `input` to call the content. See Section 3.8 for an example where each `chapter` is contained in its own file. In an article, each `section` could be contained in its own file.

3. Keep the number of packages used to a minimum. If authors feel that something is desperately missing, they can contact the maintainers of the *nrel.cls* file. Not all packages can be used as they lack compatibility.

Focus on content, not appearance. Don't spend hours trying to adjust fonts, headers or spacing between lines.

1. The document produced should meet NREL's requirements if it is compiled using *nrel.cls*.
2. Don't throw in lots of `clearpages` or other commands to push material around. LaTeX is designed to handle that.
3. Resist the temptation to add or subtract space, change lengths or do other things to modify the layout.
4. Write!

3 Some LaTeX examples

This chapter includes examples of how to do common tasks using LaTeX. Although most users will be familiar with these commands and environments, these serve as a) a test of the class file and conversion process, and b) examples that are known to work with the class and conversion process. So, when all else fails, users can copy these examples and tailor them to their particular case.

Some of the material in this section is based on introductory material from [wikibooks](#), [sharelatex.com](#), and other sources. This should be reviewed before final publication.

3.1 Headings

LaTeX allows a very simple definition of the document's structure. This document has the following structure:

- Chapter 1: what is LaTeX?
 - Section 1: Headings
 - Section 2: Floats
 - Section 3: Mathematics
 - Section 4: Lists
- etc. ...

3.1.1 Chapter

To define a new chapter, simply write `\chapter{What is LaTeX?}`.

To use chapters, pass the `memoir`, `book`, or `report` option to `nrel.cls` (see Section 2.3.4).

3.1.2 Sections

If Chapters are the highest level headings in a document, sections come next, followed by subsections. Although there don't have to be chapters in a document, a LaTeX document does need to have Sections.

So:

```
\section{Headings}
LaTeX{} allows a very simple definition of the document's structure.
This document has the following structure:
...
\subsection{Chapter}
```

3.2 Body text

Body text does not need to be specially identified in LaTeX. Non-printing comments are identified in the source document(s) using the % symbol.

3.3 Mathematics

LaTeX is great at typesetting mathematics. The following example is taken from the www.writelatex.com website:

Making inline equations is easy. Let X_1, X_2, \dots, X_n be a sequence of independent and identically distributed random variables with $E[X_i] = \mu$ and $\text{Var}[X_i] = \sigma^2 < \infty$, and let

$$S_n = \frac{X_1 + X_2 + \dots + X_n}{n} = \frac{1}{n} \sum_i^n X_i$$

denote their mean. Then as n approaches infinity, the random variables $\sqrt{n}(S_n - \mu)$ converge in distribution to a normal $\mathcal{N}(0, \sigma^2)$.

Alternatively, if numbered equations are required, use the `equation` environment. For example:

```
\begin{equation}
y = mx + c \text{ \textnormal{.}}
\label{eqn:line}
\end{equation}
```

would give:

$$y = mx + c. \quad (3.1)$$

3.4 Cross references

Use labels and references to refer back and forth to figures, equations, tables and sections. For example, Eqn. `\ref{eqn:line}` gives a reference to Eqn. 3.1.

3.5 Floats

Floats are images, tables or other pieces of the document that are free to move to the best place in the document for them. Literally, they ‘float’. The two most common floats are the `tabular` environment (for tables) and the `figure` environment for figures.

3.5.1 Tables

Use the `tabular` environment to produce basic tables. Table 3 is produced using this code:

```
\begin{table} [!h]
\centering
\caption{An example table .}\label{tab:widgets}
\begin{tabular}{|r}
\hline
Item & Quantity \\
\hline
Widgets & 42 \\
Gadgets & 13 \\
\hline
\end{tabular}
\end{table}
```

Resist the temptation to stop table rows early. If all of the delimiters (`&`) are included in each row, the table will be complete and will better translate to RTF later.

Table 3. An example table.

Item	Quantity
Widgets	42
Gadgets	13

3.5.2 Figures

To include a figure in a document, use the `figure` environment and the `includegraphics` command.

```
\begin{figure}
\includegraphics[width=\textwidth]{figure's-file-name}
\caption{Caption goes here.}\label{fig:figuresLabel}
\end{figure}
```

3.5.3 Subfigures

Subfigures are implemented using the `subfig` package. Although this package is deprecated (apparently `subcaption` is now the preferred package), it plays fairly nicely with `latex2rtf` so will be used for the foreseeable future.

The `labels` in the example below allow us to make references using the `ref` command, both to the overall figure (Figure 1) and the subfigures (Figures 3.1(a) and 3.1(b)) directly. Unfortunately, `latex2rtf` does not allow multiple `labels` in a `Figure` environment, and so only the first label will be kept: therefore, it's best to just use a single label in any one `figure` environment.

```
\begin{figure}
\centering
\hfill
\subfigure[Wind turbines at the Forward Wind Energy Center in Fond du Lac
and Dodge Counties, Wisconsin. (Photo by Ruth Baranowski / NREL)
\label{fig:21206}]{\includegraphics[height=2.5in]{files/21206}}
\hfill
\subfigure[Aerial view of the National Wind Technology Center.
(Photo by Dennis Schroeder / NREL)\label{fig:20018}]
{\includegraphics[height=2.5in]{files/20018}}
\hfill
\caption{NREL images}\label{fig:NRELIImages}
\end{figure}
```

If a subfigure is split over two lines using `\`, make sure those symbols are on their own line.

3.6 Lists

To make lists with automatic numbering, use the `enumerate` environment:

1. Like this,
2. and like this.
...or bullet points ...
 - Like this,
 - and like this.



(a) Wind turbines at the Forward Wind Energy Center in Fond du Lac and Dodge Counties, Wisconsin. (Photo by Ruth Baranowski / NREL)



(b) Aerial view of the National Wind Technology Center. (Photo by Dennis Schroeder / NREL)

Figure 1. NREL images

3.7 Computer code

The `lstlisting` package has been loaded.

3.8 Creating a file structure

Use the `\input` command to import other files into your main file. For example, each of the chapters in this report could be in separate files, called *NRELRequirements* (Chapter 1), *LatexAtNREL* (Chapter 2), and so-on.

```
...
% content
\input{NRELRequirements}
\input{LatexAtNREL}
\input{LatexExamples}
\input{ConvertingToDoc}
...
```

4 Preparing a .DOC or .DOCX file from LaTeX

LaTeX users may find that they are required to convert their files into other formats for review or editing. This section describes how LaTeX files can be converted into files that can be used with Microsoft Word.

4.1 The conversion process

The NREL style has been designed to be converted into *.doc* or *.docx*. This can be achieved via one of two routes:

1. converting the latex file to rich text format (*.rtf*) using `latex2rtf`, then importing the file into a word processor, or
2. converting the latex file to doc using pandoc.

The performance of `Latex2rtf` and `pandoc` are compared in Table 4.

Table 4. A comparison of `latex2rtf` and `pandoc` for document conversion.

consideration	<code>latex2rtf</code>	<code>pandoc</code>
requires changes to latex preamble	yes	no
output format	.rtf	.docx
supports cross references	yes	no
supports subfigures	partially	no

At this time, `latex2rtf` is the recommended tool for converting LaTeX files that have been made with the NREL class file, into word processor-readable files.

4.2 Using `latex2rtf` to convert files

The `latex2rtf` program reads LaTeX files and converts common LaTeX commands into their RTF equivalent. It is effectively another LaTeX interpreter that knows a limited subset of LaTeX. See the documentation at <http://sourceforge.net/projects/latex2rtf/> for details.

4.2.1 Using `latex2rtf`

To convert a document from LaTeX to RTF, follow these steps:

1. Install `latex2rtf`
2. Compile the document in LaTeX using the NREL class with the `book`, `report`, or `article` option, remembering to update the bibliography and cross references. The sequence of commands is:
3. Convert the document to RTF format using `latex2rtf`. The example `.rtf` file included with this document is created as follows:

```
latex2rtf -o latex2rtf_demo.rtf -f3 intro_to_NREL_latex
```

4. Open the RTF file in Microsoft Word.
5. If the document contains tables of contents, tables of figures, tables of tables, or cross-references, select that text and update the fields.
6. Save the RTF file as a word-format document.

4.2.2 Using `latex2rtf` and `LaTeX` together

Because `latex2rtf` only knows a subset of `LaTeX`, it is important to account for this when preparing a `LaTeX` document. The biggest problem is the lack of many packages, which is why authors are encouraged to use the `NREL` class file, which is known to work well with `latex2rtf`. Sometimes, though, it is important to be able to remove formatting for compatibility with `latex2rtf`, and so the preamble to this document includes a check to see if `latex2rtf` is being used:

```
\newif\iflatexortf
\iflatexortf
    \documentclass[12pt,letterpaper]{report}
    \input{NRELLatex2rtf.tex}
\else
    \documentclass[report]{nrel}
\fi
```

If `latex2rtf` is used, the boolean, `\iflatexortf` will be TRUE and the commands will be interpreted as follows.

1. Set the document class to a generic `LaTeX` *article*, *report*, or *book*.
2. The file `NRELLatex2rtf.tex` will be called, which maps most of the commands that are enabled in `nrel.cls` to simpler versions that can be processed using `latex2rtf` (see Table 2).

Authors that use packages other than those listed in Table 2 may need to adjust the content of `NRELLatex2rtf.tex` according to their needs.

4.2.3 Indexes

Index entries will not be correctly converted to an `.rtf` file. `NRELLatex2rtf.tex` redefines the `index` command to do nothing when creating an `.rtf` file.

4.2.4 What to do when the conversion to rich text format fails

It is more than likely that the conversion to an `.rtf` file will fail at some point. There are a few ways to deal with this:

Convert early and often. Check that the document converts using `latex2rtf` every time a new environment is added.

Try section-by-section. Comment out the majority of the document and try to compile bit-by-bit. This will let you localize the error.

Check new packages. Please avoid using new packages. If a package has to be used, try the conversion immediately. If `latex2rtf` doesn't support the package, edit the file `NRELLatex2rtf.tex` to redefine those commands to something that will convert appropriately. Put `NRELLatex2rtf.tex` in the same directory as the `LaTeX` file to be converted.

Avoid custom commands. `latex2rtf` sometimes chokes on custom commands. A list of all recognized commands is available in the manual at <http://latex2rtf.sourceforge.net/latex2rtf.pdf>. If custom commands are used, they may need to be redefined to work with the commands that `latex2rtf` does recognize. This can also be done in `NRELLatex2rtf.tex`. You can check macros using the flag `-d2` when running `latex2rtf`.

Use copy-paste. Compile the whole document as a PDF, and save it somewhere. Then recompile using the reduced document that works with `latex2rtf`. Edit this in word and copy in the bits that killed the conversion.

Talk to a communications rep. If a document cannot be produced any other way than LaTeX with lots of packages, and `latex2rtf` just refuses to process it into a rich text file, discuss the process for having the PDF processed.

4.3 Using pandoc

Pandoc is a tool for converting documents between different formats. Pandoc is available at <http://johnmacfarlane.net/pandoc/>. Like `latex2rtf`, pandoc only supports a subset of LaTeX commands and is not guaranteed to work for all applications.

Pandoc has several advantages over `latex2rtf`:

- Pandoc supports conversion from LaTeX to many other document formats (e.g. .html as well as .docx). However, there are other well-established tools to go directly from LaTeX source code to other formats that may have better performance.
- Pandoc does not require any changes to the LaTeX source files.

To use pandoc, follow the installation instructions on the pandoc website, and then run pandoc from the command line. To convert this file, use the following command at the command line:

```
pandoc --o conversion_demo/pandoc_demo.docx intro_to_NREL_latex.tex --bibliography  
bibliography.bib --default-image-extension=.jpg
```

5 Preparing a high-quality PDF from LaTeX

If the author chooses to complete the publications process using LaTeX the author must incorporate feedback and edits in to the LaTeX source files and prepare the final PDF, following these guidelines.

5.1 PDF tagging

PDF tagging is a process whereby the components of the PDF document (headings, figures, tables, text) are marked so that a document reader can understand the document. This is useful when text to speech converters are being used. The process of tagging is also known as structuring, so that a tagged document might also be referred to as a structured document.

LaTeX does not prepare a tagged PDF document. The current solution to this is to use the tagging capability built in to Adobe's Acrobat Pro.

To prepare a tagged document, follow these steps:

1. Add tags. Go to the 'Advanced' menu. Select 'Accessibility', then 'Add tags to document'.
2. Add alternative text for figures. Context-click the Figure, select 'Properties', and fill in 'Alternate Text'. Alternatively, try the process outlined below.
3. Specify the document language. Go to the 'File' menu. Select 'Document Properties', then the 'Advanced' tab, 'Language' field. In some versions of Acrobat, the sequence is 'File', 'Properties', 'Reading Options', 'Language'.
4. Define tab order.
 - A. Go to the 'View' menu. Select 'Navigation tabs', then 'Pages'.
 - B. Click on any page, then type Ctrl-A (or Command-A on a Mac) to select all the pages.
 - C. Go to the 'Options' menu in the top right of the dialog box, and select 'Page Properties'
 - D. In the 'Tab Order' tab, select 'Use document structure'.
5. Make sure tables have headings.
 - A. Go to the 'View' menu. Select 'Navigation tabs', then 'Tags'.
 - B. Select the 'Tags' tab. This panel shows the document structure as a tree.
 - C. Navigate to the table cells that should be headers.
 - D. Check they have the type <TH>. If not, then right click on the header cell, select 'properties', select the 'Tag' tab, and change the value for 'Type' to <TH>.
6. Make sure all Chapters (or sections, if there are no chapters in the document) are correctly tagged.

5.2 Alt-text on images and equations

'Alt text' is a textual description of an equation, link or figure. The following short equation should pop-up some text when a user passes a mouse over it. This should work in most PDF readers:

$$a^2 + b^2 = c^2 \quad (5.1)$$

The alt text can be added after the PDF is compiled, or written in to the source document. The rest of this section describes how it can be added to the source and generated by LaTeX using the <http://www.ctan.org/pkg/pdfcomment> package. The general form of the command is:

```
\pdftooltip{<item>}{{<pop-up text>}}
```

The previous equation was generated using this code:

```
\begin{equation}
\pdftooltip{a^2+b^2=c^2}{An equation}
\end{equation}
```

The same approach can be used to create alt-text for images. For example, Figure 2 has been labeled. The code for this image is:

```
\begin{figure}={!h}
\centering
\hfill
\subfigure[Wind turbines at the Forward Wind Energy Center in Fond du Lac and Dodge Counties, Wisconsin. (Photo by Ruth Baranowski / NREL)]{\pdftooltip{\includegraphics[height=2.5in]{files/21206}}{This is an image}}
\hfill
\subfigure[Aerial view of the National Wind Technology Center. (Photo by Dennis Schroeder / NREL)]{\includegraphics[height=2.5in]{files/20018}}
\hfill
\caption{NREL images}\label{fig:NRELIImagesWithAltText}
\end{figure}
```



(a) Wind turbines at the Forward Wind Energy Center in Fond du Lac and Dodge Counties, Wisconsin. (Photo by Ruth Baranowski / NREL)



(b) Aerial view of the National Wind Technology Center. (Photo by Dennis Schroeder / NREL)

Figure 2. NREL images

Alt-text is not processed by `latex2rtf`. So, if the author anticipates finishing the publication solely as a .DOC or .DOCX file, they do not need to use alt-text.

5.3 Embedded fonts

NREL requires that all fonts be embedded in the final PDF. Check the PDF for embedded fonts using a PDF viewer. For example, in Adobe Acrobat Reader, look at the ‘fonts’ tag of the document properties. If any fonts are not shown as being an *embedded subset*, try the conversion again.

Encapsulated postscript figures are particularly prone to having undefined fonts. Check by compiling the document in draft mode, and seeing if the fonts are still present in the output PDF. To fix this problem, consider changing the *.eps* file to a *.png*. To do this ‘on the fly’, use this in the document’s preamble:

```
\usepackage{epstopdf}
\epstopdfDeclareGraphicsRule
{.eps}{png}{.png}{"convert eps:\SourceFile.\SourceExt png:\OutputFile"}
\AppendGraphicsExtensions{.png}
```

Bibliography

- Anon. 2014. *TeX- \LaTeX Stack Exchange*. Accessed July 21, 2014: <http://tex.stackexchange.com>.
- Lamport, Leslie. 1986. *\LaTeX : A Document Preparation System*. Addison-Wesley.
- Other, A. N., and Y. A. Nother. 2014. *A technical report*. Tech. rep. National Renewable Energy Laboratory.