

Writing NREL documents using \LaTeX

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

Wednesday 18th December, 2013

Abstract

L^AT_EX is a document preparation system that is very popular for preparing technical documents. However, it is *not* WYSIWYG and lacks some of the reviewing and editing tools of other word processing software. For this reason special care has to be taken when preparing NREL documents in L^AT_EX. This document serves as a template and combines a short introduction to L^AT_EX with details of how to implement NREL's style and formatting guidelines.

Table of Contents

1	Requirements for NREL documents	1
1.1	Use of english and other conventions	1
1.2	Formatting	1
2	How to Make L^AT_EX documents that meet NREL's requirements	2
2.1	What is L ^A T _E X?	2
2.2	General Process	2
2.3	The NREL L ^A T _E X style file	2
2.3.1	Getting <i>nrel.cls</i>	2
2.3.2	Installing <i>nrel.cls</i>	3
2.3.3	Using <i>nrel.cls</i>	3
2.3.4	Options in <i>nrel.cls</i>	3
2.3.5	Classes and packages in <i>nrel.cls</i>	4
3	Some LaTeX examples	5
3.1	Page numbering	5
3.2	Headings	5
3.2.1	Chapter	5
3.2.2	Sections	5
3.3	Body text	6
3.4	Mathematics	6
3.5	Cross references	6
3.6	Floats	6
3.6.1	Tables	7
3.6.2	Figures	7
3.6.3	Subfigures	7
3.7	Lists	8
3.8	Computer code	8
3.9	Creating a file structure	8
3.10	Bibliographies and citations	9
3.11	Best practice in writing a document in L ^A T _E X	9
4	Preparing a .DOC or .DOCX file from L^AT_EX	10
4.1	How does latex2rtf work?	10
4.2	Using latex2rtf	10
4.3	Using latex2rtf and L ^A T _E X together	10
4.4	Indexes	12
4.5	What to do when the conversion to RTF fails	12
5	Preparing a high-quality PDF from LaTeX	13
5.1	PDF tagging	13
5.2	Alt-text on images and equations	13
5.3	Embedded fonts	15

Bibliography	16
-------------------------------	-----------

List of Figures

Figure 1.	NREL images	8
Figure 2.	NREL images	14

List of Tables

Table 1.	NREL’s process for producing and reviewing \LaTeX files	2
Table 2.	Packages supported by the <i>nrel.cls</i> class	4
Table 3.	An example table.	7

1 Requirements for NREL documents

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

1.1 Use of english and other conventions

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 How to Make \LaTeX 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	Prepare document in \LaTeX using the NREL class file Prepare PDF Converts the tex document to RTF using <code>latex2rtf</code> Archive all files, including: <ul style="list-style-type: none">• tex source• images• PDF	Section 2.3 Section 5 Chapter 4
Review	Communications	Review the structure of the PDF Edit the supplied .DOC or .DOCX file using track changes	
Revision	Author	Implements required changes in the \LaTeX files.	
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://wind-dev.nrel.gov/svn/latex_editing/branches/nrel_class_template/. Users will be prompted to provide their NREL network username and password.

2.3.2 Installing nrel.cls

Any of the following methods can be used to install the class files on a computer.

All platforms place the *nrel.cls* and *nrel.bst* in the directory that the \LaTeX files are in. This will make the files available to that project, only.

Mac / OSX place the *nrel.cls* and *nrel.bst* in the user's library in the TeXMF tree. For example, place these two files in */Library/texmf/tex/latex/nrel*. This will make these files available to all projects.

Windows

Online, colalborative tools will require that *nrel.cls* and *nrel.bst* are uploaded to that website.

2.3.3 Using nrel.cls

To use the class file, insert the following text in the preamble:

```
% -----  
% PREAMBLE  
% -----  
\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 *latex2rtf* to properly convert the latex to a rich text document for reviewing (see Chapter 4).

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 L^AT_EX 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 the *nrel.cls* class. 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 L ^A T _E X code.	
geometry		sets page size and margins	✓
mathptmx		changes fonts	
helvet		changes fonts	
courier		changes fonts	
amsfonts, amssymb		supplies the AMS fonts, which are useful for mathematics	
booktabs			
graphicx		graphics handling, including .eps figures	✓
natbib	sort	handles citations and allows the \cite, \citep and \citet citation commands (see Section 3.10).	✓
fontenc	T1		
xcolor			
babel	english		
subfig		provides the subfloat environment to produce sub figures	✓ (subfloat is mapped to the subfigure command)
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)
caption			
pdfcomment		tool-tips. Also calls the package hyper ref	✓ (the tool tip is suppressed)

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.

3.1 Page numbering

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). This is implemented using the following structure in the document:

```
\begin{document}

\maketitle
\pagenumbering{roman}
\setcounter{page}{3}

\renewcommand{\contentsname}{Table of Contents}
\tableofcontents
\clearpage
\listoffigures
\listoftables
\clearpage
\pagenumbering{arabic}
```

3.2 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. ...

3.2.1 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).

3.2.2 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}
```

3.3 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.

3.4 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{\textit{rm}}{.}
\label{eqn:line}
\end{equation}
```

would give:

$$y = mx + c. \tag{3.1}$$

3.5 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.

3.6 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.

3.6.1 Tables

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

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

Table 3. An example table.

Item	Quantity
Widgets	42
Gadgets	13

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.

3.6.2 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{\label{fig:figuresLabel}Caption goes here.}
\end{figure}
```

3.6.3 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 1a and 1b) 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
\subfloat[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]{21206}}
\hfill
\subfloat[Aerial view of the National Wind Technology Center.
(Photo by Dennis Schroeder / NREL)]\label{fig:20018}}
{\includegraphics[height=2.5in]{20018}}
\hfill
\end{figure}
```

```
\caption{NREL images}\label{fig:NRELimages}
\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 1. NREL images

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

3.7 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.

3.8 Computer code

Use the `verbatim` environment to display code. Alternatives like `lstlisting` do not work. I've tried.

3.9 Creating a file structure

Use the `input` command to import other files into your main file.

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

```
\input{LatexExamples}
\input{ConvertingToWord}
...
```

3.10 Bibliographies and citations

Always use `biblatex` to organize references and store them in a single file (e.g. `/Documents/bibliography/bibliography.bib`). Add a bibliography to a document using the following commands:

```
\cleardoublepage
\bibliographystyle{nrel}
\label{sec:Bib}
\bibliography{/Users/me/Documents/bibliography/bibliography}
```

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:

- `\cite{Lamport_1986_a}` prints a simple Lamport (1986).
- `\citep{Lamport_1986_a}` puts parentheses around it (Lamport, 1986).
- `\citep[e.g.][] {Lamport_1986_a}` puts parentheses around it, and some text in there as well (e.g. Lamport, 1986).
- `\citet{Lamport_1986_a}` prints it inline, so that according to Lamport (1986),

The command `\bibliographystyle{nrel}` prints a bibliography that follows NREL’s style guide. See Chapter 5.3 for an example.

3.11 Best practice in writing a document in L^AT_EX

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 L^AT_EX document should be placed in it’s own directory.
2. Create a main L^AT_EX file that just contains the preamble, custom commands and uses `input` to call the content. See Section 3.9 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. L^AT_EX 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!

4 Preparing a .DOC or .DOCX file from L^AT_EX

The current preferred method of editing documents that have been produced using L^AT_EX is to use `latex2rtf` to convert the tex files to a rich-text format (RTF) document, and then to .DOC or .DOCX format. Communications will edit this document, which could then be used to complete the publications process.

4.1 How does `latex2rtf` work?

The `latex2rtf` program reads L^AT_EX files and converts common L^AT_EX commands into their RTF equivalent. It is effectively another L^AT_EX interpreter that knows a limited subset of L^AT_EX.

4.2 Using `latex2rtf`

To convert a document from L^AT_EX to RTF, follow these steps:

1. Install `latex2rtf`, following instructions at <http://sourceforge.net/projects/latex2rtf/>
2. Compile the document in L^AT_EX using the NREL class with the `book`, `report`, or `article` option, remembering to update the bibliography and cross references. The sequence of commands is:
 - A. `latex myfile.tex`
 - B. `bibtex myfile.tex`
 - C. `latex myfile.tex`
 - D. `latex myfile.tex`
3. Convert the document to RTF format using `latex2rtf`:

Mac use the terminal to navigate to the directory the files are in. At the prompt, type:

```
$> latex2rtf -o outputfilename.rtf -f3 inputfilename.tex
```

Windows

Online/collaborative tools Most online tools lack the ability to convert to RTF. Download the L^AT_EX files and compile them using the usual procedure.

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

4.3 Using `latex2rtf` and L^AT_EX together

Because `latex2rtf` only knows a subset of L^AT_EX, it is important to account for this when preparing a L^AT_EX 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\iflargetortf
\iflargetortf
\documentclass[12pt,letterpaper]{report}
\input{NRELLatex2rtf.tex}
\else
\documentclass[report]{nrel}
\fi

```

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

1. Set the document class to a generic L^AT_EX *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).

An example of the content of *NRELLatex2rtf.tex* follows:

```

% File NRELLatex2rtf.tex
% set margins
\usepackage[margin=1 in,letterpaper]{geometry}
% use citations
\usepackage[sort]{natbib}
% change the heading of the bibliography
\renewcommand{\bibsection}{\section{References}}
% redefine \pdfxtooltip so that it behaves differently with and without largetortf
\newcommand{\pdfxtooltip}[3][\]{#2}
%redefine the checkmark
\newcommand{\checkmark}{y\relax}
% redefine booktabs commands
\newcommand{\toprule}{\hline}
\newcommand{\midrule}{\hline}
\newcommand{\bottomrule}{\hline}
% redefine \href
\newcommand{\href}[2]{#1~ (\url{#2})}
% redefine \subfloat to match the \subfigure environment
\usepackage{subfigure}
\makeatletter
\newcommand{\subfloat}[2][\]{\subfigure{\textit{Subcaption: \protect{#1}}}{#2}}
%\newcommand{\subfloat}[3][\]{\subfigure{#1}{#2}{#3}}
% note that we can only have one '\label' in a figure environment
\makeatother
\newcommand{\subref}[1][\]{\ref{#1}}
% redefine \todo so that it gives something useful
\newcommand{\todo}[2][\]{\textbf{To Do:}~#2}
% deal with index entries:
\newcommand{\index}[1]{ }

```

Authors are encouraged to copy this approach, adjusting the content of *NRELLatex2rtf.tex* according to their needs.

4.4 Indexes

Index entries will not be correctly converted to an RTF document. If an index is required in the PDF version, redefine the index command so that it doesn't generate an output during the conversion to RTF. Try

```
\iflargetortf
\else
\newcommand{\index}[1]{}
\fi
```

in the preamble to kill index entries. Alternatively, the `\newcommand` could go in the *NRELLatex2rtf.tex* file.

4.5 What to do when the conversion to RTF fails

It is more than likely that the conversion to RTF 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, discuss the process for having the PDF processed.

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 $\langle\text{TH}\rangle$. If not, then right click on the header cell, select 'properties', select the 'Tag' tab, and change the value for 'Type' to $\langle\text{TH}\rangle$.
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:

$$a^2 + b^2 = c^2 \tag{5.1}$$

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 L^AT_EX using the <http://www.ctan.org/pkg/pdfcomment> package. The general form of the command is:

```
\pdf tooltips{<item>}{<pop-up text>}
```

The previous equation was generated using this code:

```
\begin{equation}
\pdf tooltips{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
\subfloat[Wind turbines at the Forward Wind Energy Center in Fond du Lac
and Dodge Counties, Wisconsin. (Photo by Ruth Baranowski / NREL)]
{\pdf tooltips{\includegraphics[height=2.5in]{21206}}{This is an image}}
~
\hfill
\subfloat[Aerial view of the National Wind Technology Center.
(Photo by Dennis Schroeder / NREL)]
{\includegraphics[height=2.5in]{20018}}
\hfill
\caption{NREL images}\label{fig:NRELimagesWithAltText}
\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 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

Lamport, L. (1986). *TEX: A Document Preparation System*. Addison-Wesley.