A Guide to Using LaTeX Templates in LyX

Ole M. Brastein (ole@bedko.com)

July 3, 2018

Nomenclature

WYSIWYG	What you see is what you get
TDS	TeX Directory Structure

Preface

This guide was written as part of the authors learning process with LyX. The goal of this guide is to help interested users to apply existing LaTeX and BibTeX styles and packages in the LyX editor. Additionally we discuss some relevant details of configuration of LyX in general.

Readers of this guide are encouraged to contact the author with any questions, suggestions or comments, in particular if the information in this document is found insufficient or incorrect.

Three cases are used to highlight several challenges encountered when configuring the LyX editor. It is hoped that these examples can be generalized also to other styles.

The author would like to thank Prof. Bernt Lie for his help and guidance in both writing this guide and for testing the suggested procedures in his own work.

Chapter 1

Introduction

1.1 LaTeX and Templates

One of the reasons for the widespread use of LaTeX is the possibility to apply templates and style packages to a document. By decoupling the writing of a manuscript from its typesetting, the creative process of writing is separated from the task of confirming to prescribed publication standards of formatting and layout. Perhaps the biggest challenge preventing even more widespread use of LaTeX is its dependence on specific formatting codes. These are similar to a programming language and are used to describe the non-textual parts of a manuscript. For users who are accustomed to the comfort of a "What you see is what you get" (WYSIWYG) style word processor such as Microsoft Word, the use of LaTeX may seem a somewhat daunting challenge.

1.2 LyX and LaTeX Templates

A simpler alternative is to use a graphical user interface tool built on top of a LaTeX back-end. There are many such tools to choose from. One option is to use LyX, a self proclaimed "manuscript processor" rather then a word processor. Unlike Word, LyX is not strictly WYSIWYG but rather "What you see is what you mean" (WYSIWYM). This slight spin on the familiar WYSIWYG acronym is indicative of the way LyX works. By offering a standard on-screen layout to be used while writing, the task of manuscript writing is again separated from the typesetting and layout. However, LyX is not just a front-end for LaTeX. It uses its own file formats and it is built on top of the LaTeX system MiKTeX. As such, it is not directly compatible with standard LaTeX, but the difference is not large. LyX supports all the features of the LaTeX language. In fact, a LyX document can be exported into standard `.tex` files.

Since one of the main reasons for using LaTeX is the application of different layout files it is of interest to verify that such files can indeed be applied to a LyX document. As is often the case with free (and in fact open source) software, this process is not straight forward. In the following, a detailed description is given of how to import a LaTeX class and style package and use it from within LyX as a layout. The demonstration is given for a Windows PC. Since some of the challenges in this process is locating the correct folders for placement of files, some adaption to individual systems must be expected.

1.3 Directories and File Types

Two particular *base directories* on your Windows computer will be of particular importance in the set-up, and will be referred to in the sequel:

1. *Base MiKTeX* directory[1]: typically `C:\Program Files (x86)\MiKTeX 2.9`,
2. *Base LyX* directory[2]: typically `C:\Program Files (x86)\LyX 2.2`.

There are two — sometimes three — file types that are used by the LaTeX system and one that is used by LyX to define the typesetting of a manuscript, as listed in Table 1.1.

Table 1.1: File types used by LaTeX and LyX.

File type	Description
`.cls`	Defines a LaTeX document class
`.sty`	Defines additional styles for a document (aka “package”)
`.cfg`	LaTeX configuration file
`.layout`	LyX layout file

As shown in Table 1.1, a type or *class* of document is defined by a `.cls` file. Additional styling can be defined in the `.sty` file. Typical, a *class* file defines the structure of a manuscript, such as an article or a book. It provides definitions for typesetting of the different sections of the manuscript. Packages, or *style* files, are used in addition to a class file. It amends the class file with additional functionality, such as graphics or customization of headers in default LaTeX classes. *Configuration* files are sometimes used for specifying page size and margins settings, as well as font definitions.

LyX adds an additional file type called `.layout`. This is a combination of a class file and style file in order to form a complete layout description. It is also possible to add more definitions of formatting directly in the layout files, but this discussion is beyond the scope of this guide. The defined layout can then be used inside LyX to specify the typesetting of a document. There are many predefined `.layout` files available in the `Resources>layouts` folder, located in the root directory for LyX (usually `C:\Program Files (x86)\LyX 2.2`). These can be used as a starting point for creating new `.layout` files customized for a particular publication or institute. Similarly, there are a many class and style files located in the MiKTeX folder (usually `C:\Program Files (x86)\MiKTeX 2.9`).

Observe that LaTeX uses a strictly defined structure on its folder tree where `.cls` and `.sty` files are stored. The root directory of any tree must contain a `tex` folder, which in turn contains a `latex` folder. This structure is termed “TDS compliant” by LaTeX (TDS is an acronym for TeX Directory Structure). Users can create their own trees to store new class and style files outside of the MiKTeX root folder. As discussed in Chapter 3 of this document, there can be additional folders located in the TDS tree, such as folders for storing reference databases and bibliography styles for the BibTeX system. The interested reader should refer to specific LaTeX manuals for more details on the TDS tree.

1.4 Case Studies

Three LaTeXclasses are used in this guide as examples on how to configure LyX. In this subsection each case is presented and some details unique to each case is discussed. A

[1] Currently, MiKTeX v. 2.9

[2] Currently LyX v. 2.2.3

detailed step-by-step procedure given in Chapter 2. Readers without prior LATEXor LyX experience are referred to Chapter 2 for details.

1.4.1 SIMS/Modelica Conference Style

As an example of how to set up LyX to use both class and style files for a conference paper, this guide demonstrates the configuration of LyX using the templates for SIMS[3] and Modelica conferences. Specifically, the template for the SIMS2018 conference is used as a case study.

The SIMS2018 LATEXtemplates can be downloaded as a zipped file from `http://www.scansims.org/media-files/conference-templates-master.zip`. Click your way into the LATEX folder. Here you find a class file `sims.cls` and a style file `titlesec.sty`. In addition, you need the `resources` folder with the `listings-sims.cfg` file.

Later in the document, the location where these files/directories should be stored, will be discussed.

The full .layout file can look something like the following:

```
#% Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[sims]{SIMS Article(sims)}
# \DeclareCategory{Articles}
# Input general definitions
Input stdclass.inc

Preamble
        \hypersetup{
        colorlinks,
        linkcolor=black,
        urlcolor=black,
        citecolor=black,
        pdfpagelayout = SinglePage,
        pdfcreator = pdflatex,
        pdfproducer = pdflatex}
EndPreamble

ClassOptions
        PageStyle       ""
End
```

The above layout file will be discussed in detail in Chapter 2. The SIMS style contains a `\bibstyle` command, hence no additional bibstyle should be specified in LyX. Therefore the bibliography style must be left blank, as discussed in Chapter 3. The last section of the layout file sets the ClassOptions, specificaly the PageStyle as empty. This prevents page numbers to be generated.

1.4.2 Automatica Journal Style

Another example of a LATEXclass file, taken from the Elsevier Journal Automatica, can be obtained from `http://autsubmit.com/documents/nonhtml/autart.cls`. This layout requires only a single .cls file with no further dependencies.

```
#% Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[autart]{Article Automatica (autart)}
```

[3] Scandinavian Simulation Society

```
# \DeclareCategory{Articles}
# Input general definitions
Input stdclass.inc
```

The above .layout file is simple. It only needs the inclusion of the autart.cls file. How to set up such a layout file, and how to make it work in LyX will be discussed in Chapter 2.

1.4.3 USN Thesis Styles

A third example of relevant LaTeXclass and style files are those used by *University of South-Eastern Norway* (USN). These can be obtained from GitHub at `https://github.com/dietmarw/USN-LaTeX`. Observe that in online *GitHub* using `Right Click > Save Link As ...` downloads the `html` code of the page that displays the `.cls` file, *and not the* `.cls` *file* which is of course what we need. Please ensure that files are downloaded using appropriate GitHub procedures.

There are separate document class files for bachelor, master and PhD theses. Observe that the three class files for bachelor, master and PhD theses uses a common base style. This base style is referred to from within each of the three theses types and must therefore exist in your local TDS structure. However, it is not necessary to include it in the LyX layout file. A `.layout` file for the PhD Thesis can be created simply like follows:

```
#% Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[USN-PhD]{USN PhD Thesis (USN-PhD)}
# \DeclareCategory{USN-Thesis}
# Input general definitions
Input stdclass.inc
```

The details of how to create the above .layout file is discussed in Chapter 2. However, in the case of the USN classes there is a font problem when using them withing LyX. The default LaTeXprocessor in LyX is *pdflatex*. The USN styles reports a fatal error related to fonts when *pdflatex* is used, as shown in Fig. 1.1. Hence it is necessary to specify another output format. After some initial testing it appears that XeTeX works with the USN PhD document class.

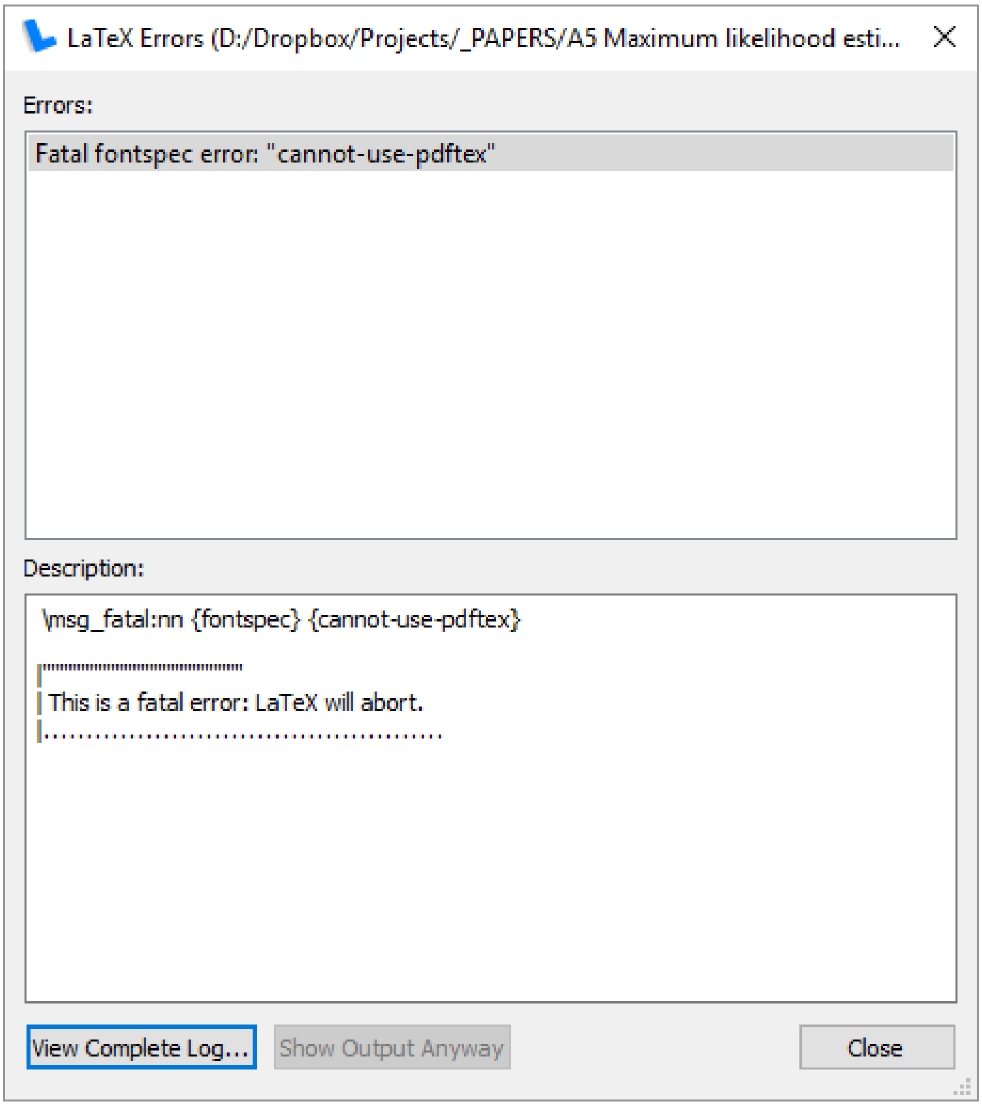
Ideally we want the change in output format to be configured in the .layout file. This could potentially be done by adding a line to the layout file:

```
OutputFormat XeTeX
```

Unfortunately, there appears to be a known bug in LyX (v2.2.3), see: `https://tex.stackexchange.com/questions/154719/set-default-output-format-in-layout-in-lyx`.

When the OutputFormat specification is used in a .layout file, it somehow confuses LyX such that the *View* button and the corresponding Ctrl+R shortcut is disabled. Hence, a better alternative is a manual operation for each document to use the USN style. The root problem is the use of `\fontspec` in the USN class styles in LaTeX, which uses non-TeX fonts. This is not supported by pdflatex, hence another rendering tool must be used. XeTeX or LuaTeX seems to be the recommended choices. There are two ways that we can tell LyX to use *non-TeX fonts.*

The first possibility is to select `Document > Settings > Fonts` and check the box for using non-Tex fonts, as shown in Fig. 1.2. This may be the preferred choice as it most closely matches the actual problem, i.e. that `\fontspec` in USN styles requires non-TeX fonts. When non-TeX fonts LyX reverts to another default OutputFormat for PDF's; PDF (XeTeX). Hence another way to solve the `\fontspec` problem with the USN styles is to specify the output format directly. This can be done by selecting the menu options



LaTeX Errors (D:/Dropbox/Projects/_PAPERS/A5 Maximum likelihood esti...
Errors:
Description:
View Complete Log...
Show Output Anyway
Close

Fatal fontspec error: "cannot-use-pdftex"

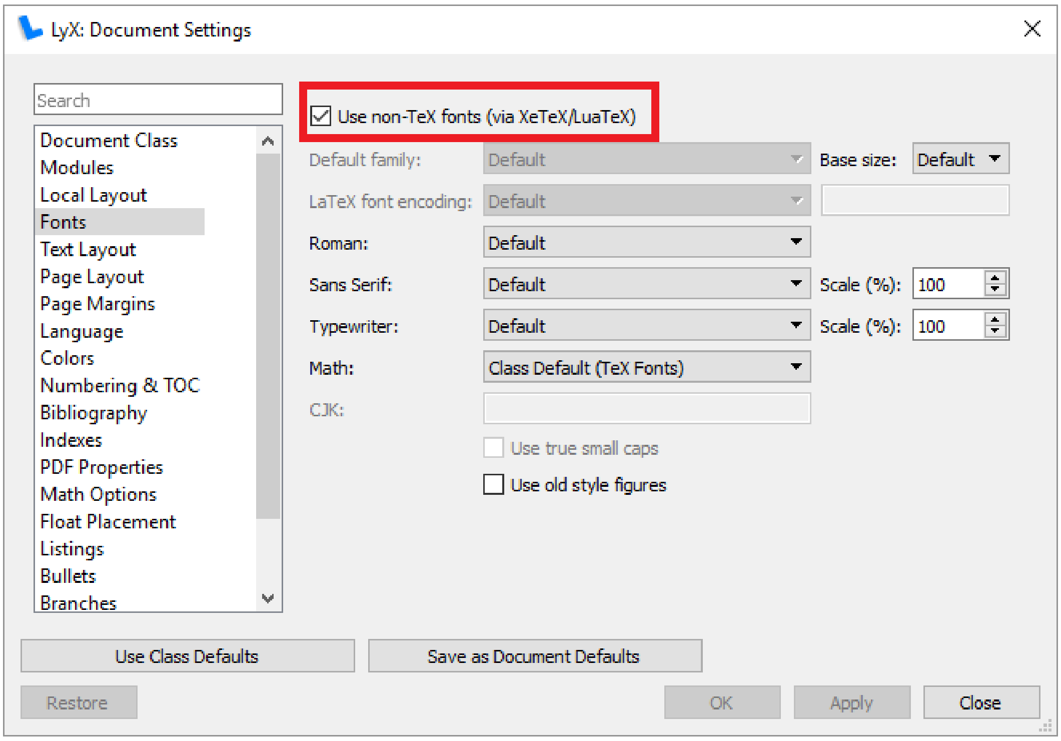
```
\msg_fatal:nn {fontspec} {cannot-use-pdftex}

|''''''''''''''''''''''''''''''''''''''''''''''
| This is a fatal error: LaTeX will abort.
|..............................................
```

Figure 1.1: Fatal error if pdflatex is used.

`Document > Settings > Formats`. From the drop-down list "Default output format" select an alternative specification, e.g. PDF(XeTeX), as shown in Fig. 1.3.

Either method works for the USN PhD style, and the result is the same (using XeTeX as PDF generator). A third possibility is to do the export without using the *View* button or *Ctrl+R* shortcut, by instead selecting the menu `File > Export > PDF (XeTeX)`. It is however more convenient to ensure that the *View* button and *Ctrl+R* shortcut works as usual.

A final step to make the USN PhD style work is to solve the dependency on the USN logo. These are obtained as a .pd file from `https://github.com/dietmarw/USN-LaTeX/tree/master/logos`. There are both Norwegian and English versions. The logo files must be downloaded and stored in the same folder as the .lyx document, *not* in a sub-folder. typically a *resources* sub-folder is used for figures. This is different from the SIMS style, where the dependency files are in expected to be in the *resources* sub-folder.



☑ Use non-TeX fonts (via XeTeX/LuaTeX)

Default family: Default

Base size: Default

LaTeX font encoding: Default

Roman: Default

Sans Serif: Default

Scale (%): 100

Typewriter: Default

Scale (%): 100

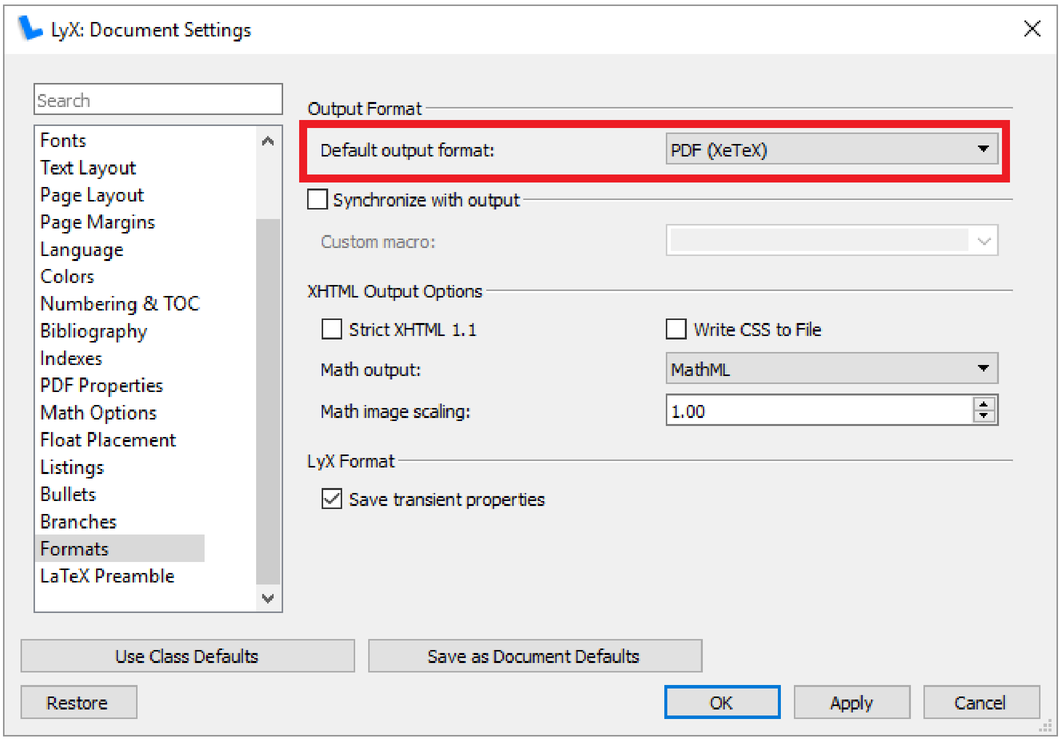
Math: Class Default (TeX Fonts)

CJK:

□ Use true small caps

□ Use old style figures

Figure 1.2: Tell LyX to use non-TeX fonts.



Output Format

Default output format: PDF (XeTeX)

□ Synchronize with output

Custom macro:

XHTML Output Options

□ Strict XHTML 1.1

□ Write CSS to File

Math output: MathML

Math image scaling: 1.00

LyX Format

☑ Save transient properties

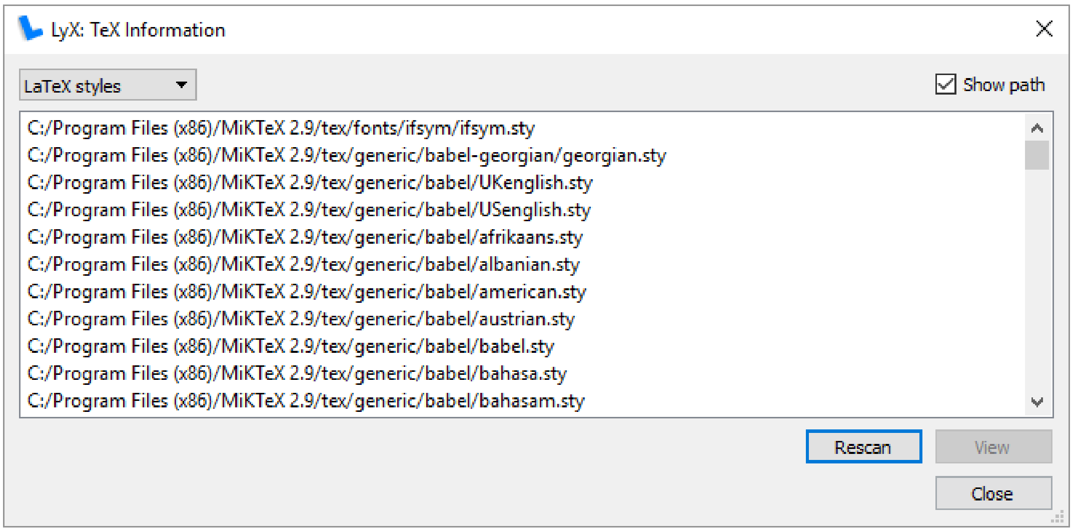
Figure 1.3: Selecting different output formats.

Chapter 2

Step by Step Procedure

2.1 Inspecting Available LaTeX Classes and Styles

Before a new LaTeX package can be added to LyX, it must first be added to the underlying LaTeX processor. LyX relies on MiKTeX[1] to do the processing of a manuscript, hence it can only use classes and styles that are also known to MiKTeX. To get a list of LaTeX (and also BibTeX) classes and styles supported on the computer system in question, users can go to the LyX menu `Tools > TeX Information`. As shown in Fig. 2.1 the TeX Information dialog provides a detailed list of all installed class and style files, as well as their full path — if `Show path` is checked.



```
C:/Program Files (x86)/MiKTeX 2.9/tex/fonts/ifsym/ifsym.sty
C:/Program Files (x86)/MiKTeX 2.9/tex/generic/babel-georgian/georgian.sty
C:/Program Files (x86)/MiKTeX 2.9/tex/generic/babel/UKenglish.sty
C:/Program Files (x86)/MiKTeX 2.9/tex/generic/babel/USenglish.sty
C:/Program Files (x86)/MiKTeX 2.9/tex/generic/babel/afrikaans.sty
C:/Program Files (x86)/MiKTeX 2.9/tex/generic/babel/albanian.sty
C:/Program Files (x86)/MiKTeX 2.9/tex/generic/babel/american.sty
C:/Program Files (x86)/MiKTeX 2.9/tex/generic/babel/austrian.sty
C:/Program Files (x86)/MiKTeX 2.9/tex/generic/babel/babel.sty
C:/Program Files (x86)/MiKTeX 2.9/tex/generic/babel/bahasa.sty
C:/Program Files (x86)/MiKTeX 2.9/tex/generic/babel/bahasam.sty
```

Figure 2.1: Screenshot of the TeX Information dialog for a particular system. In the top left corner there is a selection for what type of files to list. In the right hand corner a selection allows to show the full path of each file.

Presumably, the style of interest is not listed in the TeX Information dialog.

2.2 Creating a TDS compliant directory tree for LaTeX files

Before adding new classes and styles (e.g., for SIMS), a custom TeX Directory Structure (TDS) compliant root is needed. MiKTeX already contains such a folder structure for

[1] ...or TeXLive

pre-installed class and style files. It is possible to add the files in that folder structure with administrative rights, but it is better to put your special class and style files elsewhere!

The root of the TDS compliant tree is known as the TEX and METAFONT directory, the TeXMF directory. To install new classes and styles, it is thus recommended to create a local TeXMF directory, e.g., named LocalTeXMF — in the examples to follow, the LocalTeXMF directory is created as follows: D:\Dropbox_LIBRARIES\LocalTeXMF.

Next, before any class and style files are installed, the TDS structure must be installed under the LocalTeXMF directory: first, the LocalTeXMF directory must hold a sub-folder named tex, and this directory tex must in turn hold a sub-folder named latex. See Fig. 2.2.

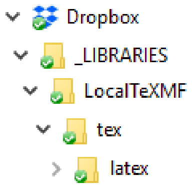


Figure 2.2: Example of a TDS Compliant folder structure.

At this stage, the TDS structure has been created. However, we are not done until we have made MiKTeX aware of the TDS structure and where to find our class and style files. *Before* any files are added to the latex directory, we should make MiKTeX aware of our LocalTeXMF directory. In other words: we need to add the LocalTeXMF directory to MiKTeX's list of directories in which to search for class and style files. This is done through the MiKTeX Settings tool.

The easiest way to start the MiKTeX Settings tool is to click in the Windows search icon on the task bar and start typing MiKTeX... — something similar to Fig. 2.3 shows up — possibly with the wording (Admin) appended.

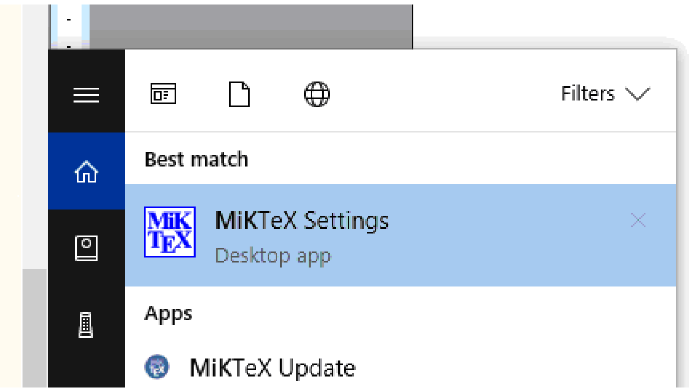
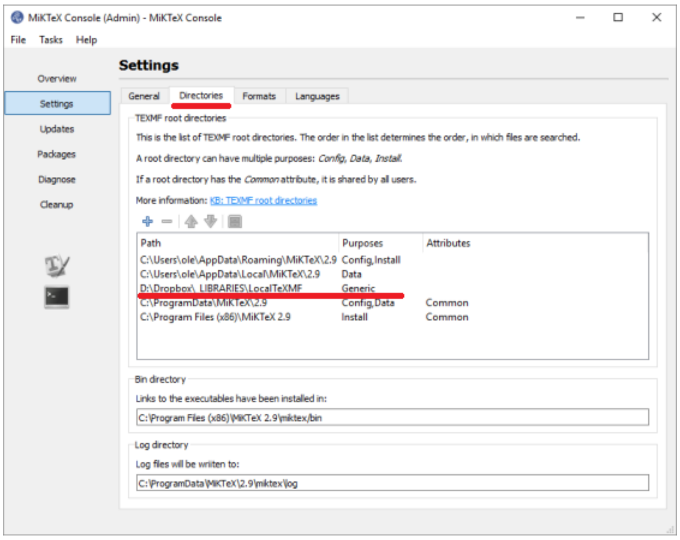


Figure 2.3: MiKTeX Settings can be found by Windows search.

Launch the MiKTeX Settings tool by clicking on the the icon.



Settings

TEXMF root directories

This is the list of TEXMF root directories. The order in the list determines the order, in which files are searched.

A root directory can have multiple purposes: *Config, Data, Install.*

If a root directory has the *Common* attribute, it is shared by all users.

More information: KB: TEXMF root directories

Path	Purposes	Attributes
C:\Users\ole\AppData\Roaming\MiKTeX\2.9	Config,Install	
C:\Users\ole\AppData\Local\MiKTeX\2.9	Data	
D:\Dropbox_LIBRARIES\LocalTeXMF	Generic	
C:\ProgramData\MiKTeX\2.9	Config,Data	Common
C:\Program Files (x86)\MiKTeX 2.9	Install	Common

Bin directory

Links to the executables have been installed in:

C:\Program Files (x86)\MiKTeX 2.9\miktex/bin

Log directory

Log files will be writen to:

C:\ProgramData\MiKTeX\2.9\miktex\log

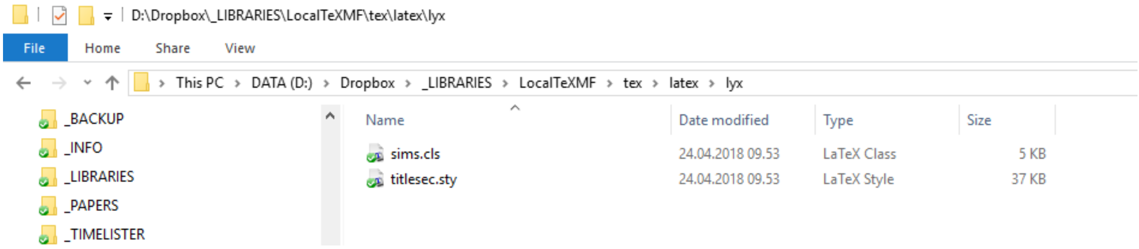
Figure 2.4: Contents of the `Directories` tab in MiKTeX Settings.

In the MiKTeX Console tool, click on the `Directories` tab and then click on the `+` symbol, then navigate the File Explorer and select the `LocalTeXMF` directory that you just created[2], and click `OK`. If the chosen folder does not contain the appropriate TDS compliant folder structure, there will be a warning message. However, if you have followed the instructions, it should look like in Fig. 2.4.

2.3 Adding Class and Style Files, etc.

Now that MiKTeX has been informed of a new location to search for class and style files, the relevant style files must be added. These can be added directly in the new `latex` subdirectory under the `LocalTeXMF` directory. However, it is recommended to first create a `lyx` subdirectory under the `latex` directory, and place the class and style files which are dependencies of a LyX layout in the `lyx` subdirectory.

For the case of the SIMS style, the class file `sims.cls` and the additional style file `titlesec.sty` should thus be placed in the `lyx` directory, as shown in Fig. 2.5.


D:\Dropbox_LIBRARIES\LocalTeXMF\tex\latex\lyx
File
Home
Share
View
This PC › DATA (D:) › Dropbox › _LIBRARIES › LocalTeXMF › tex › latex › lyx
_BACKUP
_INFO
_LIBRARIES
_PAPERS
_TIMELISTER

Name	Date modified	Type	Size
sims.cls	24.04.2018 09.53	LaTeX Class	5 KB
titlesec.sty	24.04.2018 09.53	LaTeX Style	37 KB

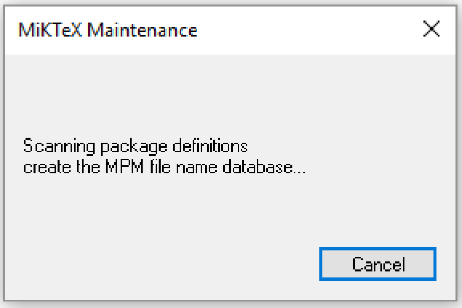
Figure 2.5: Adding new style files to the chosen TDS tree.

Observe that for the SIMS template, directory `resources` holding the `listings-sims.cfg` file does not belong in the TDS tree! See below!

Once the class and style files have been added to the `lyx` folder, we need to use the MiKTeX Settings tool to re-scan all its folders for the new class and style files. This is done as follows: click in the Windows search icon on the task bar and start typing MiKTeX... — something similar to Fig. 2.3 shows up — launch the MiKTeX Settings

[2] NOT the `tex` or `latex` directories, but the `LocalTeXMF` directory!

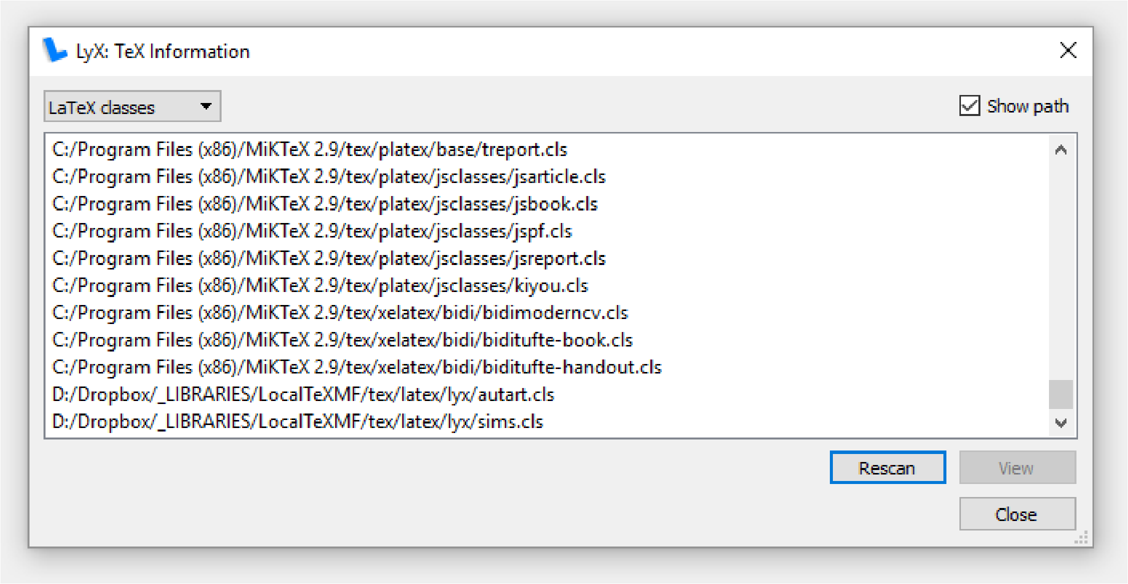
tool by clicking on the the icon. Next, choose the pull-down menu `Tasks`, and select `Refresh file name database`. A popup dialog as in Fig. 2.6 is shown while the system is working.



```
Scanning package definitions
create the MPM file name database...
```

Figure 2.6: Scanning for package definitions.

The final step of adding the new packages to LaTeX is to verify that LyX can now see them. This can be done by using the TeX Information dialog under the `Tools` menu in LyX, as shown in Fig. 2.7. Note that it may be necessary to click the `Rescan` button. A popup dialog as in Fig. 2.6 is shown while the system is working.



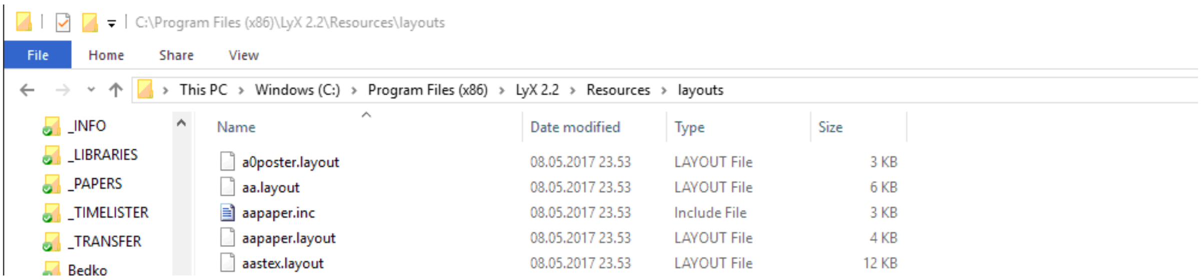
```
C:/Program Files (x86)/MiKTeX 2.9/tex/platex/base/treport.cls
C:/Program Files (x86)/MiKTeX 2.9/tex/platex/jsclasses/jsarticle.cls
C:/Program Files (x86)/MiKTeX 2.9/tex/platex/jsclasses/jsbook.cls
C:/Program Files (x86)/MiKTeX 2.9/tex/platex/jsclasses/jspf.cls
C:/Program Files (x86)/MiKTeX 2.9/tex/platex/jsclasses/jsreport.cls
C:/Program Files (x86)/MiKTeX 2.9/tex/platex/jsclasses/kiyou.cls
C:/Program Files (x86)/MiKTeX 2.9/tex/xelatex/bidi/bidimoderncv.cls
C:/Program Files (x86)/MiKTeX 2.9/tex/xelatex/bidi/biditufte-book.cls
C:/Program Files (x86)/MiKTeX 2.9/tex/xelatex/bidi/biditufte-handout.cls
D:/Dropbox/_LIBRARIES/LocalTeXMF/tex/latex/lyx/autart.cls
D:/Dropbox/_LIBRARIES/LocalTeXMF/tex/latex/lyx/sims.cls
```

Figure 2.7: Verifying that the new class and/or style is now available in LaTeX.

You should now be able to see the `sims.cls` file under the `LocalTeXMF/tex/latex/lyx` directory by scrolling to the bottom of the LaTeX classes.

2.4 Creating a new LyX layout

Now that MiKTeX has been made aware of the new class and style files, it is time to configure a layout file for LyX. Start at the base LyX directory (e.g., `C:\Program Files (x86)\LyX 2.2`) locate the `Resources`, and then the `layouts` folder where LyX stores all its layout files. Unlike MiKTeX, LyX wants all layout files stored in the same folder. Figure 2.8 shows an example of the `Resources > layouts` folder located under the base LyX directory.


C:\Program Files (x86)\LyX 2.2\Resources\layouts
File
Home
Share
View
This PC › Windows (C:) › Program Files (x86) › LyX 2.2 › Resources › layouts
_INFO
_LIBRARIES
_PAPERS
_TIMELISTER
_TRANSFER
Bedko

Name	Date modified	Type	Size
a0poster.layout	08.05.2017 23.53	LAYOUT File	3 KB
aa.layout	08.05.2017 23.53	LAYOUT File	6 KB
aapaper.inc	08.05.2017 23.53	Include File	3 KB
aapaper.layout	08.05.2017 23.53	LAYOUT File	4 KB
aastex.layout	08.05.2017 23.53	LAYOUT File	12 KB

Figure 2.8: LyX Resource layout folder.

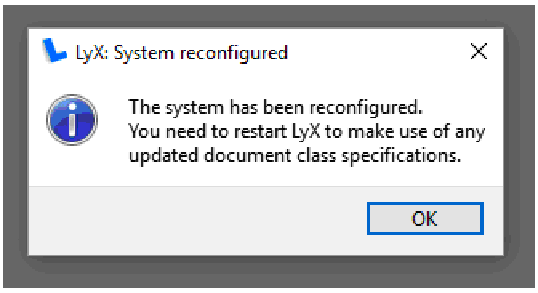
Note that this folder is protected by Windows — adding files to this folder will require administrator privileges. It may not be possible to save a file directly to the protected folder. Instead save the layout file to, e.g., `Desktop` and copy it to the `Resource > layouts` folder manually.

Thus, first a `sims.layout` file must be developed, and then this file must be copied into the `layouts` folder as described above. In general, creating a layout file is complicated. A good example of a layout file is the `aa.layout` included in LyX v.2.2. However, for the SIMS template, we only need the simplest layout file. Thus, use an ASCII editor and insert the following code into the file, and save the file as `sims.layout` in the LyX layouts folder. Some general suggestions for extensions that can be added to the layout file are given in Section 2.6.

```
#% Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[sims,titlesec.sty]{SIMS Article(sims)}
# \DeclareCategory{Articles}
# Input general definitions
Input stdclass.inc
```

The first line above is only a comment. The second line contains the definition of the class. The keyword `\DeclareLaTeXClass` defines the link between a LyX layout and LaTeX class and style files. In the square brackets `[]`, add first the name of the cls file, without the `.cls` suffix. Then list any additional style files with the `.sty` suffix. This is the convention used in the pre-installed layout files. In the curly braces `{}`, define the name of the layout as will be shown in LyX. By convention, an arbitrary chosen name is used with the actual LaTeX `.cls` file name in parenthesis. The third line defines the category of the new layout. This determines where in LyX list of layouts it will be located. Here, `Articles` is chosen since the new layout is created for SIMS2018 conference. The last two lines of the layout file is first a comment declaring that what follows is general definitions of input files, followed by a inclusion of a standard class file. These two lines appear to be included in all the predefined layouts and should be kept as-is.

Once the layout file has been created, saved and copied into the `Resources > layouts` director, the next step is to refresh LyX's list of layouts. This is done from the Tools menu in LyX. Select the option `Reconfigure`, which will launch a re-scan of all the layout files.



LyX: System reconfigured

The system has been reconfigured.
You need to restart LyX to make use of any updated document class specifications.

Figure 2.9: LyX reconfiguration complete.

The reconfiguration is complete when LyX pops a message as in Fig. 2.9. Before the new layout appears as an option in `LyX Document Settings`, LyX must be restarted.

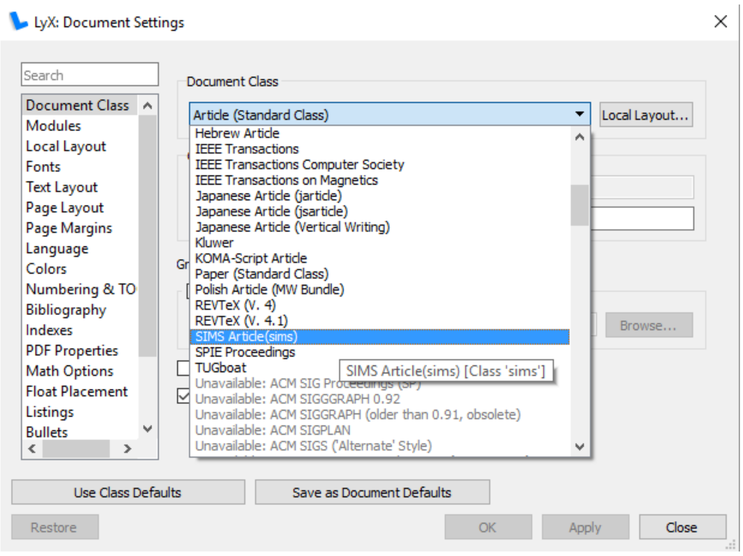


Figure 2.10: Selecting the new layout for a document.

Now the newly created layout is available as a selection from the `Document > Settings > Document Class` menu in LyX. If the underlying LaTeX files are not found, the layout may still appear in the list, but in gray with the pre-fix "unavailable". This indicates a problem with the LaTeX system, such as invalid TDS folder structure or missing class or style package files.

2.5 On dependencies on resources

Some LaTeX style packages, such as the `sims` example used here has dependencies on certain resources. The `sims.cls` file requires a configuration file `listings-sims.cfg` which must be located in a folder named `resources`. This `resources` folder must be located in the same directory as the `.lyx` document, and the `resources` folder must contain the `listings-sims.cfg` filed downloaded as part of the SIMS template.

The `resources` folder may also contain other document relevant files, e.g., figures, etc. — or figures may be imported from other directories.

2.6 Extending the layout file

There are a large variety of settings that can be configured through the layout files of LyX. One useful possibility is to add a LaTeX preamble that will be used for all documents of a particular layout. For example, to get rid of the red, green and blue frames around hyperlinks, use the LaTeXcommand `\hypersetup` by adding the following to the layout file after `Input stdclass.inc`:

```
Preamble
        \hypersetup{
        colorlinks,
        linkcolor=black,
        urlcolor=black,
        citecolor=black,
        pdfpagelayout = SinglePage,
        pdfcreator = pdflatex,
        pdfproducer = pdflatex}
EndPreamble
```

This preamble is taken from the example article contained in the SIMS style package. By adding this to e.g. the SIMS Layout file for LyX we don't need to add it to the preamble of the document itself. Presumably, any LaTeX command can be added to the default preamble for our LyX layout using the above configuration. It is supposedly possible to achieve the same result using `Document > Settings > PDF Properties > Use hyperref support` from the LyX menu system, but this option appears to not work correctly. Hence it is better to use raw LaTeXcode in the preamble defined in the .layout file to achieve the same result.

Observe that after making changes to a .layout file, LyX needs to be updated by accessing the menu option `Tools > Reconfigure`, the same way that was done previously when creating a new `.layout` file.

2.6.1 Calligraphic fonts

There appears to be an issue with the SIMS style which makes calligraphic fonts italic. A possible remedy for this is to add

```
\DeclareMathAlphabet{\mathcal}{OMS}{cmsy}{m}{n}
```

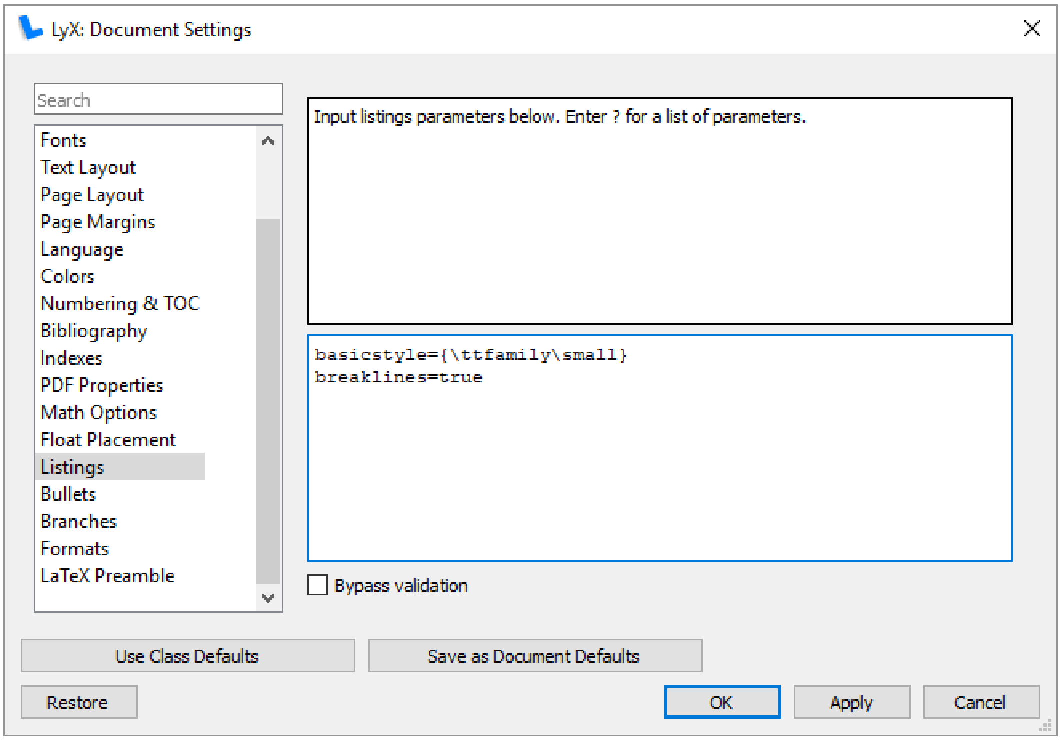
to the preamble, as discussed here: `https://tex.stackexchange.com/questions/67881/resetting-mathcal-font-to-default` (viewed 22.05.2018).

2.7 Other settings

2.7.1 Program listing style

There are several other settings in LyX that may be of use. One such example is the possibility to customize program listings. As default, inserted program listings are printed using the same fonts as regular text, which makes it difficult to distinguish program listings from regular text, especially when used inline. It is possible to set the font for each program listing manually, but this process can be labor intensive. A better option is to select the document settings menu `Document > Settings > Listing` and supply default formatting options here, as shown in Fig. 2.11.

Observe that there are exists a rather large selection of settings for the Program Listings package, including possibility to format code according to a specific language.



Input listings parameters below. Enter ? for a list of parameters.

```
basicstyle={\ttfamily\small}
breaklines=true
```

□ Bypass validation

Figure 2.11: How to set font and size of program listings.

Chapter 3

BibTeX Styles

3.1 What is BibTeX?

The most popular bibliography system used with LaTeX is BibTeX. This system is natively supported in LyX. A BibTeX system consist of a database .bib file where a set of references are stored. A tool such as JabRef can be used to setup and maintain the database. One your references are added to the BibTeX database, they can be added to your LyX document.

3.2 Add a BibTeX bibliography

The first step is to add a bibliography by selecting the menu `Insert > List/TOC > BibTeX Bibliography` as shown in 3.1.

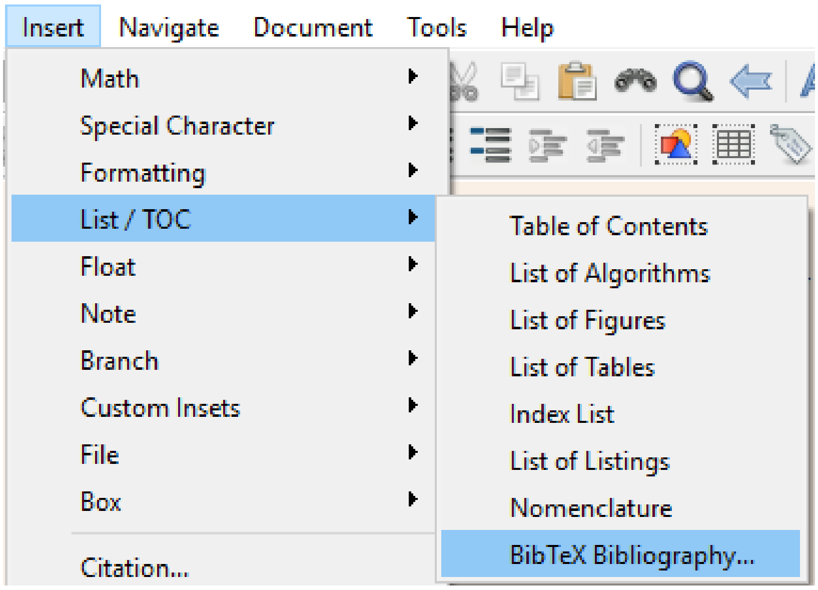
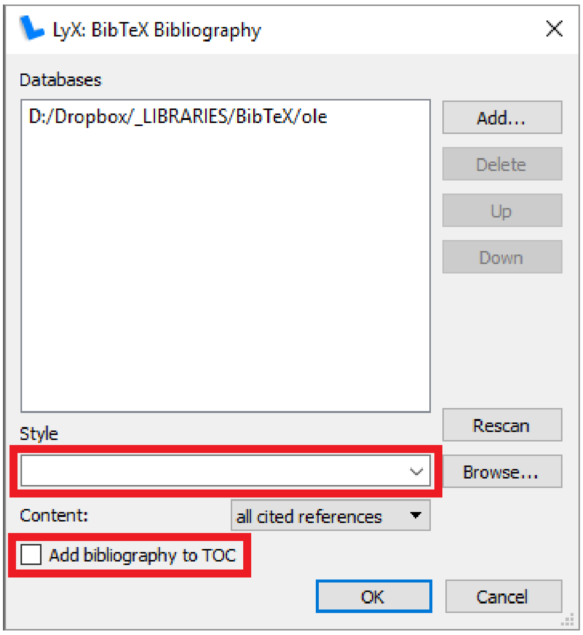


Figure 3.1: Insert a BibTeX bibliography by selecting the LyX menu `Insert>`.

This menu selection shows the BibTeX Bibliography configuration tool shown in Fig. 3.2. Here we will add one or more .bib database files to our bibliography. Note that only citations that are found in the BibTeX databases added here can be used in the document later.



Databases
D:/Dropbox/_LIBRARIES/BibTeX/ole

Style

Content: all cited references

□ Add bibliography to TOC

Figure 3.2: Setup BibTeX databases for the bibliography.

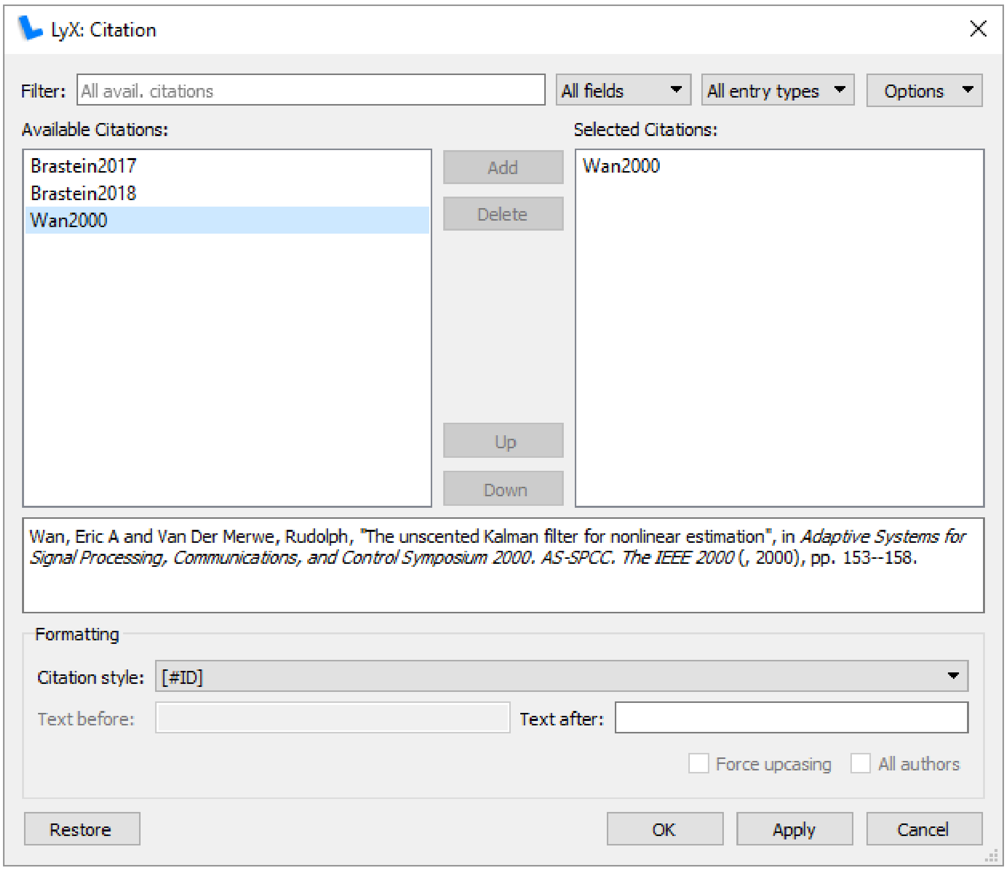
To add a new BibTeX database, click the `Add` button. This opens a list of all known BibTeX databases on your system. If the file you want to add is not present in the list, it can be added manually by using the `Browse` button and navigating to the correct file. If you store your .bib files within a TDS tree that is recognized by MiKTeX, the database will be shown in the list. A good choice is therefore to store your .bib files in the LocalTexMF folder `LocalTeXMF/bibtex/bib`. Observe that just as .cls files must be stored in `LocalTeXMF/tex/latex` there are requirements for sub-folders for .bib files.

Once all the necessary BibTeX databases has been added, a bibliography style can be chosen. The default choice is *plain* which gives a bibliography where the references are numbered. For a author/year style, use instead *plainnat*, or choose another bibliography style according to the specifications of your journal. Some document classes, such as our SIMS example, already have a bibliography style included. In such cases, the style selection in the dialog of Fig. 3.2 *must be kept empty* in order to avoid a compilation error due to multiple `\bibstyle` commands. Observe also that the `Add bibliography to TOC` is *not* checked. This option is known to create problems with certain LaTeX styles such as SIMS. If checked, this option introduces a dependency on `bibtotoc.bst` which LyX cannot find. Further, it adds another `\bibstyle` command and therefore causes the same problem as specifying a non-empty `Style` field.

After completing the bibliography configuration, click on the `ok` button. There should now be a `BibTeX Generated Bibliography` tag inserted into your LyX document.

3.3 Inserting citations

Once the bibliography is configured, adding a citation is done from the LyX menu `Insert > Citation` , or alternatively using the corresponding menu bar button. Either option opens the citation dialog shown in Fig. 3.3.



Available Citations:
Brastein2017
Brastein2018
Wan2000

Selected Citations:
Wan2000

Wan, Eric A and Van Der Merwe, Rudolph, "The unscented Kalman filter for nonlinear estimation", in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000* (, 2000), pp. 153--158.

Citation style: [#ID]

Text before:

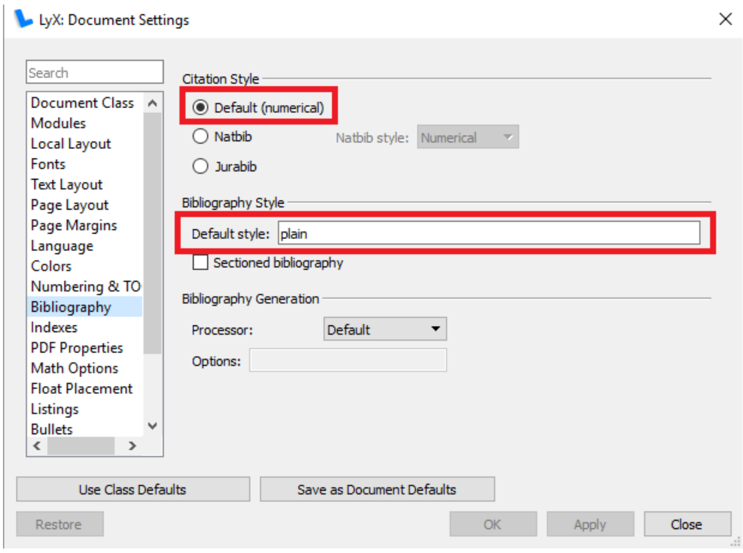
Text after:

□ Force upcasing □ All authors

Figure 3.3: Insert a citation.

The left side of this dialog list all the references in the configured BibTeX databases. Select the references you want to insert and click the `Add` button and click `ok` when done. The reference should now appear in LyX.

Observe that it is possible to further configure citations in the text by selecting the LyX menu `Document > Settings > Bibliography` as shown in Fig. 3.4.



Citation Style

● Default (numerical)

○ Natbib Natbib style: Numerical

○ Jurabib

Bibliography Style

Default style: plain

□ Sectioned bibliography

Bibliography Generation

Processor: Default

Options:

Figure 3.4: Let the citation style remain as default to avoid duplicate invocations of the natbib package, if that package is already used in the class or style files.

Here, options for the *natbib* package can be configured. This allows more options for each citation, such as e.g. choosing whether or not the author name should be located within brackets. Observe however that if the *natbib* package is already configured in the .cls file, using the LyX configuration tool can cause a conflict which in turn produces a compilation error. The latter is the case for the SIMS style example used here.

3.3.1 On BibTeX formatting

When setting up the BibTeX database it is important to keep in mind that the defined text here will eventually be imported to a LaTeX document. Hence, certain characters should be avoided. For example, the use of a dash (-) is problematic since this character exists outside the 0-127 range of the ASCII table (there is a difference between dash and minus). Instead use the "math mode" syntax `$-$` which forces a dash type glyph that is ASCII compatible. There are also other symbols that can be problematic for LaTeX such as ö and ô. Instead, use the special symbol and escapes characters defined here: `http://www.bibtex.org/SpecialSymbols/`.

Chapter 4

Tips and tricks

4.1 Review

LyX supports reviewing manuscripts by marking changes in a way similar to that used in Microsoft Word. First, the review toolbar must be enabled, as shown in Fig. 4.1.

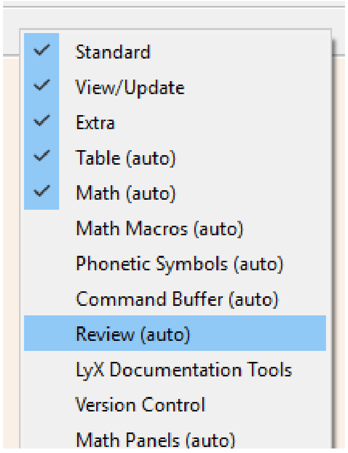


Figure 4.1: Enable review toolbar.

When the toolbar is enabled, a panel of buttons is available for controlling the marking of changes, as shown in Fig.

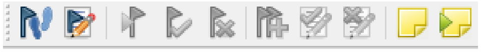
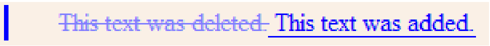


Figure 4.2: Buttons for controlling the review process.

The first button toggles the automatic marking of changes. When turned on, LyX will automatically mark any changes to the manuscript. Additions are presented as blue underlined text while deletions are marked with light blue text with and a strikethrough. An example is shown in Fig. 4.3.



~~This text was deleted.~~ This text was added.

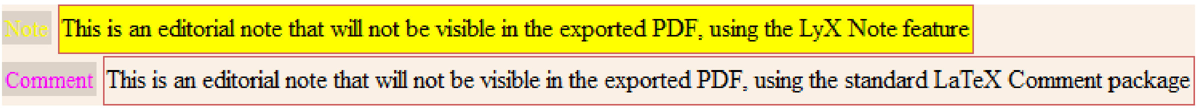
Figure 4.3: Example of review changes with deleted and added text

The second button toggles whether or not the marked changes are exported when creating PDFs.

The next three buttons are used for navigating to the next marked change in the text, accepting the change or rejecting the change, respectively. Note that both accepting and rejecting changes are applied to any marked text, thus it is possible to accept/reject multiple changes at once. Once a change has been accepted/rejected, this decision can be undone using the standard Undo command (or Ctrl + z).

Button six is used to launch a merge changes *wizard*. This tool searches through the manuscript for any changes and prompts the user for an accept/reject decision. Button seven and eight can similarly be used to change or reject *all* changes in the manuscript.

The last two buttons can be used to add editorial notes in the text, and to navigate to the next such note. Notes are yellow highlighted text that is not exported to the PDF and can be used for writing comments in the manuscript for future reference. Under the menu `Insert > Note` There are some other possible comment/notes options that can also be used for similar purposes. The yellow highlighted note is a LyX feature, while the comment option is a LaTeXstandard feature. Neither should be confused with the % symbol used in LaTeXto comment out raw code. An example of both types of notes/comments are shown in Fig. 4.4.



This is an editorial note that will not be visible in the exported PDF, using the LyX Note feature

This is an editorial note that will not be visible in the exported PDF, using the standard LaTeX Comment package

Figure 4.4: LyX and LaTeXstyle notes and comments.

Chapter 5

Conclusion

When writing for publications, in particular for scientific journals and conferences, confirming to a specific layout is a strict requirement. While there are advantages in simplicity of Microsoft Word, the end result can hardly compare with that of a proper LaTeX manuscript. For users not inclined to invest the necessary time to learn raw LaTeX, the use of a front-end system such as LyX can be a good choice. The drawback is that these systems, unlike Microsoft Word, requires some effort to configure correctly. As this guide has shown, there are some steps that must be executed in order to add new LaTeX layouts for use in LyX. However, the end result is generally worth the extra effort. Once a user learns to navigate the configuration and setup of LyX, the creative writing process is just as easy as in Microsoft Word.