

<i>PharmTeX</i>	<b>User Guide</b>
Report Title:	PharmTeX User Guide
Date Issued:	04 OCT 2023
Version:	2.1
Author:	Christian Hove Claussen
Contact:	<a href="mailto:contact@pharmtex.org">contact@pharmtex.org</a>
Website:	<a href="http://pharmtex.org">http://pharmtex.org</a>

## TABLE OF CONTENTS

LIST OF TABLES .....	4
LIST OF FIGURES .....	4
LIST OF APPENDICES .....	6
ABBREVIATION AND DEFINITION OF TERMS .....	7
1. STARTUP CHECKLIST .....	8
2. INTRODUCTION .....	8
3. SETTING UP OR UPDATING PHARMTEX .....	8
4. TESTING THAT PHARMTEX WORKS .....	10
5. WORKING ON A DOCUMENT .....	13
6. WRITING IN LATEX CODE .....	14
7. RESERVED LATEX SYMBOLS .....	15
8. STRUCTURE OF A PHARMTEX REPORT .....	15
9. PHARMTEX CLASS OPTIONS .....	19
10. SETTING THE PAPER SIZE AND MARGINS .....	20
11. SECTIONS .....	21
11.1. Subsections .....	21
11.1.1. Subsubsections .....	21
12. CROSS REFERENCES .....	21
13. GLOSSARY ENTRIES .....	22
14. CITATIONS .....	23
15. FOOTNOTES .....	23
16. BULLET POINTS AND NUMBERED ITEMS .....	24
17. EQUATIONS .....	25
18. EXTERNAL HYPERLINKS .....	27
19. FIGURES .....	27
19.1. Figures from Image Files .....	27

19.2. Figure Arrays .....	28
19.3. Inline-Produced Figures .....	29
20. TABLES .....	30
20.1. Tables from Delimiter Files .....	30
20.2. Inline Tables .....	35
20.3. Multicolumn and Multirow Tables .....	35
20.4. Table Notes .....	36
20.5. Tables Based on Image Files .....	37
20.6. Excel2LaTeX Tables .....	37
20.7. Font Size and Whitespace .....	39
21. INCLUDING A CHILD DOCUMENT .....	39
22. INCLUDING A TEXT FILE INLINE .....	39
23. R KNITR SUPPORT .....	40
24. DEFINING CUSTOM COMMANDS .....	41
25. RUNNING PERL CODE IN PHARMTEX .....	42
26. CONVERSION TO OR FROM MICROSOFT WORD FORMAT .....	42
26.1. Conversion to Microsoft Word from the Finalized PDF .....	42
26.2. Conversion of Advanced Equations and Mathematical Symbols .....	42
26.3. Other Conversion Tools .....	43
27. COMPILING THE SYNOPSIS AS A SEPARATE DOCUMENT .....	44
28. INCLUDING NLMIXR OUTPUT IN PHARMTEX .....	44
29. INTEGRATING EXTERNAL FILE SYSTEMS WITH PHARMTEX .....	45
30. COMMON WARNINGS AND ERRORS .....	46
31. REFERENCES .....	48
LIST OF ITEMS .....	54

## LIST OF TABLES

Table 1.	Key files in the "pharmtex" subfolder .....	9
Table 2.	Key files in a report folder .....	10
Table 3.	Texstudio F-key functionality .....	13
Table 4.	PharmTeX class options.....	20
Table 5.	PharmTeX class settings for margins .....	20
Table 6.	Table from file .....	31
Table 7.	Table from file with header substitution .....	31
Table 8.	Table from file with default justification.....	32
Table 9.	Table from file with column width scaling .....	33
Table 10.	Table from file with column swapping.....	33
Table 11.	Table from file with LaTeX code .....	34
Table 12.	Table from file with LaTeX code (no compilation) .....	34
Table 13.	Inline table .....	35
Table 14.	Multicolumn/row table.....	36
Table 15.	Inline table with table note.....	37
Table 16.	Table created from an image file.....	37
Table 17.	Excel2LaTeX table.....	38
Table 18.	Common warnings and errors .....	47

## LIST OF FIGURES

Figure 1.	PharmTeX software bundle location .....	9
Figure 2.	Screenshot of Texstudio .....	12
Figure 3.	Visual predictive check.....	28
Figure 4.	Residual plots.....	29
Figure 5.	Example of inline figure.....	30
Figure 6.	R figure .....	41

Figure 7. Remote file system artifact ..... 45

## LIST OF APPENDICES

Appendix 1. INCLUDING AN EXTERNAL PDF .....	49
Appendix 2. INCLUDING CODE .....	53

## ABBREVIATION AND DEFINITION OF TERMS

<b>Notation</b>	<b>Description</b>
AGPLv3	GNU Affero General Public License Version 3.
CI	Confidence interval.
CL	Clearance.
nlmixr	Nonlinear Mixed Effects Models in Population Pharmacokinetics and Pharmacodynamics.
NONMEM	NONlinear Mixed Effects Modeling.
PD	Pharmacodynamic(s).
PK	Pharmacokinetic(s).
QA	Quality assurance.
%RSE	Relative standard error expressed as percent.

## 1. STARTUP CHECKLIST

If you are the kind of user that always throws the manual over the shoulder and installs "freestyle", this is the section for you. However, it is strongly recommended to read at least some of this user guide at some point if you want to familiarize yourself with PharmTeX.

1. Download latest PharmTeX software bundle from [pharmtex.org](http://pharmtex.org) under "Downloads", double-click the ZIP to open it, and drag-and-drop the "pharmtex" folder within to `C:\Users\USERNAME` on Windows 10/11 and `/home/USERNAME` on Linux. You need to install the "libfuse2" package on Ubuntu Linux as of 22.04 (used by QPDF and Texstudio AppImages).
2. Download the example report from [pharmtex.org](http://pharmtex.org) under "Downloads" and unzip to your preferred location.
3. Double-click "executex.bat" in Windows or double-click "executex" or run "bash executex" in Linux in the example report folder to start up PharmTeX and Texstudio.
4. Press F1 in Texstudio to do a full compile.

## 2. INTRODUCTION

PharmTeX is an open-source framework for creating publishing-ready reports directly from figure and table files. The framework is based on [LaTeX](http://www.latex-project.org/), the gold standard for typesetting scientific documents. PharmTeX is released under the GNU Affero General Public License Version 3 (AGPLv3).

This user guide has the objective of giving you as a PharmTeX user the ability to:

- Set up PharmTeX on your computer.
- Initialize a report.
- Use PharmTeX features to put in various key report components.
- Finalize a report to make it ready for publishing.

## 3. SETTING UP OR UPDATING PHARMTEX

PharmTeX consists of three main components:

1. A software bundle.
2. A set of class configuration files.
3. A document folder with the files that are edited by the user.

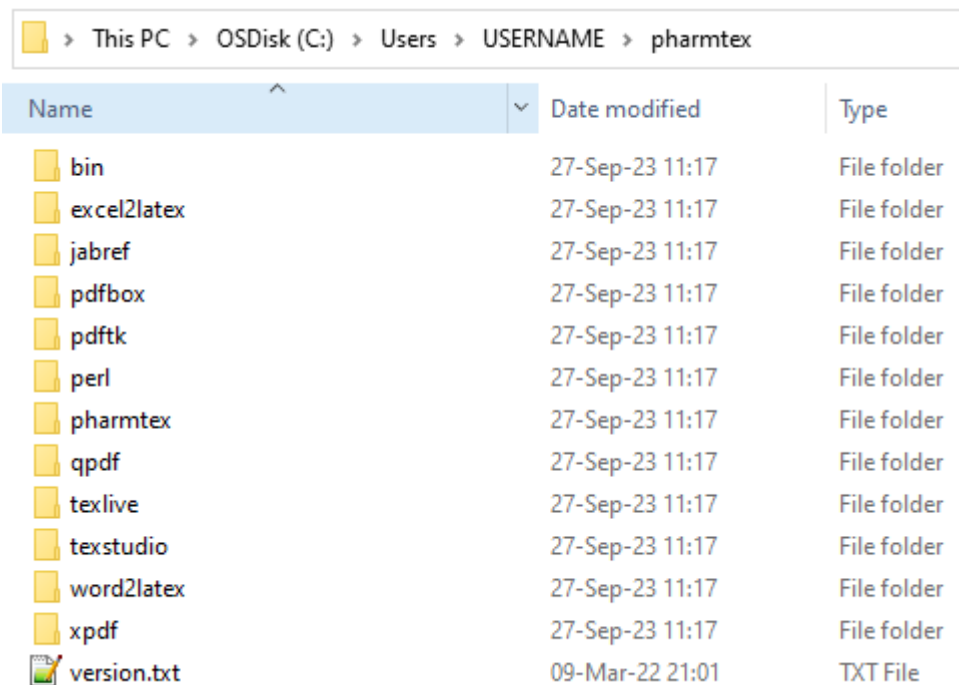
In this guide, we will assume that you are using the PharmTeX software bundles. They are available on [pharmtex.org](http://pharmtex.org) under Downloads. Currently, versions for Windows and Linux are available. You need to install the "libfuse2" package on Ubuntu Linux as of 22.04 (used by QPDF and Texstudio AppImages). Please download the one suitable for your operating system.

Once the software bundle ZIP file is downloaded, double-click it to open it and drag-and-drop the "pharmtex" folder within to `C:\Users\USERNAME` on Windows 10/11



and `/home/USERNAME` on Linux. When you are done, the location and contents on Windows 10/11 should be as shown in [Figure 1](#), with USERNAME matching your Windows login name:

**Figure 1. PharmTeX software bundle location**



[Item 4](#). PharmTeX software bundle location.

The class files are located in a subfolder also called "pharmtex" in the root of the software bundle folder, `C:\Users\USERNAME\pharmtex\pharmtex` on Windows 10/11 and `/home/USERNAME/pharmtex/pharmtex` on Linux. NOTE that the class folder is also called "pharmtex" and is one layer down from the main root folder called "pharmtex". This folder holds the main configuration files for PharmTeX that modify the default behavior of LaTeX and the other software in the bundle. See [Table 1](#) for a description of each file.

**Table 1. Key files in the "pharmtex" subfolder**

File	Type	Description
cheatsheet.pdf	PharmTeX code snippets	Use this file to find code snippets for commonly used elements like figures and tables
downloadartifacts.pl	Perl script for remote file system	This is where you can put your code that PharmTeX will execute to download files from an external location during the compilation process
LICENSE	License agreement ( <a href="#">AGPLv3</a> )	License agreement for PharmTeX (don't worry, it's free and open-source!)
PharmTeX.cls	PharmTeX class file	This is where most of the LaTeX settings for PharmTeX are stored
PharmTeX.ini	Initialization for class file	This file is used to compile the class file
PharmTeX.sty	PharmTeX package file	Storage of additional settings for PharmTeX

**Table 1. Key files in the "pharmtex" subfolder**

File	Type	Description
pmx.bst	Configuration for reference list	Settings for the bibliography
runlatex.bat	Execution script for Windows	This script is used by PharmTeX to initialize and in Windows
runlatex.pl	Perl main control script	This is the main Perl control script for executing LaTeX in PharmTeX
runlatex.sh	Execution script for Linux	This bash script is used by PharmTeX to initialize and in Linux

Inline table. Key files in the "pharmtex" subfolder.

The latest version of the software bundles is always updated with the latest class version upon download from [pharmtex.org](http://pharmtex.org), but you may want to update the class files later on. You can see the class version in the top of "PharmTeX.cls" (open the file with Notepad in Windows or a similar editor in Linux).

To update to the latest software bundle, delete or rename the old bundle folder (C:\Users\USERNAME\pharmtex on Windows 10/11 and /home/USERNAME/pharmtex on Linux) and follow the procedure described in the beginning of this section to download the newest bundle.

If you want to update the PharmTeX class files, download latest version from [pharmtex.org](http://pharmtex.org) under "Downloads" and use the contents of the ZIP file to overwrite the files in C:\Users\USERNAME\pharmtex\pharmtex on Windows 10/11 and /home/USERNAME/pharmtex/pharmtex on Linux.

#### 4. TESTING THAT PHARMTEX WORKS

Once the software bundle is in place, please download the example document ZIP file from [pharmtex.org](http://pharmtex.org) (located under Downloads). Double-click the ZIP file and drag-and-drop the "example" folder to any location on your computer, e.g. in your "Documents" folder. Inside the "example" folder you will see the files in [Table 2](#):

**Table 2. Key files in a report folder**

File	Type	Description
document.tex	Main report file	This is where you write your report.
preamble.tex	Preamble file	Used to store settings for the document.
synopsis.tex	Synopsis file	Write your report synopsis here. You may delete it if you do not want a synopsis.
<b>executex.bat</b>	Windows start file	Double-click this file in Windows to start up PharmTeX. You may delete it if you are running Linux.
<b>executex</b>	Linux start file	Double-click this file in Linux (needs to executable permissions) or use "bash executex" in a terminal to start up PharmTeX. You may delete it if you are running Windows.
glossary.tex	Global glossary	Contains global glossary definitions provided by the PharmTeX creators.

**Table 2. Key files in a report folder**

File	Type	Description
references.bib	Reference database file	Holds the list of references to cite in PharmTeX. Is maintained using JabRef.

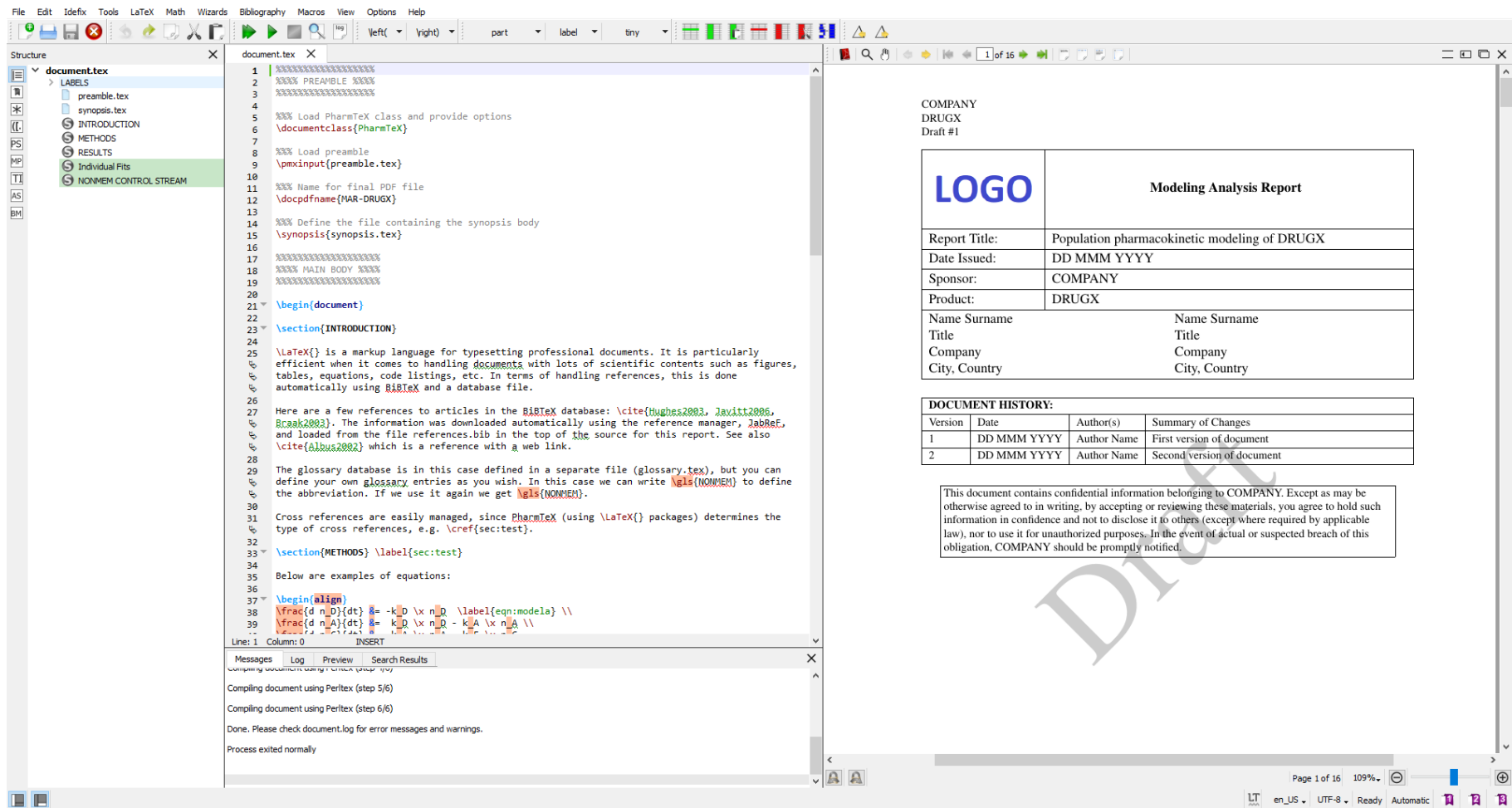
Inline table. Key files in a report folder.

All other files in a PharmTeX document folder will be files that you add, e.g. figures, tables, etc. Note, in the example document folder we have also added some test input files (figures tables, code listings) that you can use to try out PharmTeX. The example also has a test synopsis ("synopsis.tex") in a separate file, and a file called "preamble.tex" which holds a lot of the formatting and setup information for the document.

Next, we will test the example report. Inside the "example" folder, double-click the "executex.bat" file if you are running Windows. Viewing of file extensions may be disabled on your computer, in which case you simply try both files named "executex" and see which ones executes (only the BAT file will work correctly). You may get the warning "Windows protected your PC", in which case you have to select "Run anyway" (only the first time you run PharmTeX on the computer). On Linux, double-click the "executex" file (no file extension) or start it in a terminal using `./executex`. It will open a terminal windows that will prompt you for the name of the document you want to edit. Write "document" as in the name of the main report file and press enter. You do not need to enter the extension ".tex". This will open up Texstudio and the "document.tex" file. This has to be done every time when opening a PharmTeX document. Do not double-click the TEX file to open it as PharmTeX will not initialize correctly.

Once the document is open in Texstudio, press F1 (you may need to enable Fn-Lock/FnLk on your keyboard (Fn+FnLk) to gain access to the F-keys) on your keyboard and you will see it says "Process started" in the bottom left window of Texstudio. Once done, Texstudio will take you to the place in the PDF document (right window in [Figure 2](#)) that corresponds to the point in the source code (left window in [Figure 2](#)). You can also go from source to PDF by pressing F3. If you press the "Ctrl" button on your keyboard and left-click in the PDF window (right), Texstudio will take you to the same place in the source code (left). Any compilation errors will show up in the box below the source window. Make sure to check the "Log" pane in the box below the source for any warnings.

**Figure 2. Screenshot of Texstudio**



**Item 5. Screenshot of Texstudio.**

## 5. WORKING ON A DOCUMENT

When creating a new PharmTeX report, it is recommended to always start from a template. A standard blank report with some default sections is available in the download file "template.zip" on [pharmtex.org](http://pharmtex.org) under "Downloads". This can be used as your starting point for a new report.

Contemplating [Figure 2](#), Texstudio consists of a source window where you write your report text (left/middle) and an output window where you can see the compiled PDF file (right). The window to the very left show the structure of the document, i.e. which sections etc. are present. If you press the little symbols on the far left, the left window will show different menus with shortcuts to various mathematical symbols. These symbols can be used in mathematical environments which are covered in [Section 17](#). Use the F-keys for the following functionality:

**Table 3. Texstudio F-key functionality**

Key	Function	Description
F1	Full compile	Performs a full compilation to ensure that all page numbers, cross references, citations, hyperlinks, etc. are working properly.
F2	Fast compile	Use this key to compile minor changes, i.e. most just added plain text or check your code for errors. Page numbers etc. may not be accurate until F1 is used.
F3	View PDF	Placing the cursor somewhere in the source window and pressing F3 will take you the approximate corresponding place in the PDF file.
F4	Clear files	Deletes all temporary files created by PharmTeX, essentially allowing you to start over. It will not delete your figure, table and other files you have manually added.
F5	Start JabRef	Starts up an instance of JabRef. It will also open any BibTeX database(s) you have in the report folder.
F6	Open cheatsheet	Opens a PDF with a cheatsheet containing handy copy snippets that can be copied into the document.
F7	Recompile Class	Recompiles the class file (PharmTeX.cls). Used if you choose to modify the class file (not recommended) and want to see the results. Also triggers an "mktexlsr" command to initialize new custom package added to "pharmtex" subfolder.
F8	Open Tex Live package manager	Opens the Tex Live package manager (tlshell) that you can use to install more LaTeX packages.
F12	Finalize report	Does a full compile and applies the submission-ready components of PharmTeX. Once done, all temporary files will be cleared.

Inline table. Texstudio F-key functionality.

Note that F7 is a tool advanced users/developers that should not be used unless you have a specific reason for doing so. This reason could be that you have added a new custom package (STY file) of settings for the reference list (BST file) to the "pharmtex" subfolder in the pharmtex root folder, e.g. to have settings for a company or institution template that modify the default PharmTeX layout. In this case you just need to use F7 the first time you add the file. If you have just modified the custom files the PharmTeX compilation process will take care of the update.

Until finalization, the watermark "Draft" will appear and all hyperlinks will show. Upon finalization, hyperlinks to the same page and external hyperlinks are disabled to comply with the PharmTeX submission-ready standards. If the document has not been finalized, cross references and page numbers may be inaccurate, so please use the F12 key before issuing the report. Note that compiling using F1 will also produce correct cross references and page numbers, but F12 is required for the document to be submission-ready.

The next sections will now cover how different PharmTeX/LaTeX elements are added to the report. If you are familiar with LaTeX, some sections be skipped.

## 6. WRITING IN LATEX CODE

LaTeX and PharmTeX work by reading a text document (extension TEX) and translating plain text and LaTeX code into a typeset PDF file. The LaTeX engine recognizes certain symbol as initiators of commands, most the `\` symbol. This symbol initiates a LaTeX function that takes a certain number of input arguments in curly brackets, `{}`. The typical use of such a function looks like the following:

```
\function{argument1}{argument2}{argument3}...
```

LaTeX ignores new lines and more than one space in compiling the code, with the exception of a double new line (an empty line) between text paragraphs:

```
This is line no. 1.  
  
This is line no. 2.
```

This is line no. 1.

This is line no. 2.

This means that the following two are equivalent:

```
\function{argument1}{argument2}{argument3}
```

and

```
\function  
{argument1}  
{argument2}  
{argument3}
```

Functions can take optional arguments in square brackets `[]` that can be omitted if not used to allow the function defaults to apply:

```
\function  
[optional]  
{argument1}  
{argument2}  
{argument3}
```

## 7. RESERVED LATEX SYMBOLS

The following symbols are reserved in LaTeX: `_ ^ % \ & # $ { }`. They initiate subscript, superscript, a comment, a command, and other elements. Therefore, they need to be reconfigured to be printed in the document by using this syntax in the same order:

```
\_ \^{} \% \textbackslash{} \& \# \$ \{ \}
```

This produces the desired symbols correctly:

```
_ ^ % \ & # $ { }
```

## 8. STRUCTURE OF A PHARMTEX REPORT

A PharmTeX document is a text file with the extension "TEX". The file is edited in Textstudio and begins with the following lines:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
%%% PREAMBLE %%%  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
  
%% Load PharmTeX class and provide options  
\documentclass{PharmTeX}  
  
%% Load preamble  
\pmxinput{preamble.tex}
```

The `\documentclass` loads the PharmTeX class while the `\pmxinput{preamble.tex}` command loads the contents of the file "preamble.tex" into the document at that point. Looking at the first lines of "preamble.tex", we see:

```
%%%%%%%%%%  
%%% SETTINGS %%%  
%%%%%%%%%%  
  
%% Load PharmTeX packages  
\usepackage{PharmTeX}  
  
%% Logo and style files  
\logo[scale=0.5]{logo.png}  
\refstyle{pmx.bst}  
\reffile{references.bib}  
\glsfile{glossary.tex}
```

The `\usepackage` command loads the PharmTeX package (with the Perl code). The next four commands load the logo image file, reference list formatting file (BST), reference database (BIB), and the global glossary.

The next lines shown below define the header and footer for the document. Additionally, the watermark can be modified if needed. The default is "Draft" for non-finalized documents and none for finalized documents.

```
%%%%%%%%%%  
%%% HEADER, FOOTER, AND WATERMARK %%%  
%%%%%%%%%%  
  
%% Text for header  
\headertext  
{COMPANY}  
{DRUGX}  
{Draft \#1}  
  
%% Information for footer  
\footertext{COMPANY CONFIDENTIAL}  
  
%% Add watermark (uncomment).  
%\watermark{Confidential}
```

The lines below provide information needed for the cover page of the document. Besides the type, title, and date, additional lines can be added to the coverpage default table using the `\coverline` command. Multiple authors can be added by repeated use of the `\docauthor` command, issuing a `\newline` command within the `\docauthor` command when as needed. If a more customized, artistic cover page is preferred, a ready-to-use PDF can be created elsewhere and loaded using the command `\coverpage[optional page no]{file}`. The warning box can be modified (or omitted) as needed. The same for the `\versionhistory`.



```
%%%%%%%%%%
%%% COVER PAGE %%%
%%%%%%%%%%

%% Information for document cover page table
\doctype{Modeling Analysis Report}
\doctitle{Population pharmacokinetic modeling of DRUGX}
\docdate{DD MMM YYYY} % you can use \today to use today's date

%% Additional info for coverpage table
\coverline{Sponsor:}{COMPANY}
\coverline{Product:}{DRUGX}

%% Authors
\docauthor{
Name Surname\newline
Title\newline
Company\newline
City, Country
}
\docauthor{
Name Surname\newline
Title\newline
Company\newline
City, Country
}

%% Version history
\versionhistory{
1 & DD MMM YYYY & Author Name & First version of document \\ \
hline
2 & DD MMM YYYY & Author Name & Second version of document \\ \
hline
}

%% Warning box
\warningbox{This document contains confidential information
belonging to COMPANY. Except as may be otherwise agreed to in
writing, by accepting or reviewing these materials, you agree to
hold such information in confidence and not to disclose it to
others (except where required by applicable law), nor to use it
for unauthorized purposes. In the event of actual or suspected
breach of this obligation, COMPANY should be promptly notified.}

%% Include customized PDF coverpage: \coverpage[optional page no
]{file}
%\coverpage[1]{coverpage.pdf}
```

The setup of the approval page for the report is similar to that of the cover page. The command `\approverline` can be used to add information to the approval page while the

`\docapprover` can be used to add approvers. When the document is finalized (F12 key), a file called "sigpage.pdf" is automatically outputted along with the report itself. This page can be simply the approval page in a single-PDF format and can be used to send around to the document approvers. Once signed, it can be loaded in place of the approval page using the `\signaturepage`[optional page no]{file} command, thus producing a publishing-ready report. If needed, a quality assurance (QA) statement can be included as the very last page of the document using the PharmTeX command `\qapage`[optional page no]{file}.

```
%%%%%%%%%%
%%% APPROVAL PAGE %%%
%%%%%%%%%%

%% Additional info for approval page table
\approverline{Authors:}{N. Surname, N. Surname}

%% Document approvers
\docapprover{
Name Surname\newline
Title\newline
Company\newline
City, Country
}
\docapprover{
Name Surname\newline
Title\newline
Company\newline
City, Country
}

%% Include signed approval PDF page: \signaturepage[optional page
no]{file}
%\signaturepage[1]{./signaturepage.pdf}

%% Include signed Quality Assurance Statement: \qapage[optional
page no]{file}. Uncomment the line below as is to create a
placeholder page (blank). This makes sure that the page numbers
are correct when sending the approval page for signatures.
%\qapage[1]{./qastatement.pdf}
```

Looking again back in the main "document.tex", `\docpdfname` command is used to set the name of the finalized PDF file (produced using the F12 key). The synopsis is defined inside the file "synopsis.tex" which is loaded with the command `\synopsis`{synopsis.tex}. The synopsis contains multiple sections, subsections, etc. Note that, as shown below, the starred version of the command to start sections should be used, i.e. `\section*`. This suppresses the number of the section as is the intent in a PharmTeX synopsis.

```
\synopsis{  
\section*{OBJECTIVE (S) }  
  
\section*{STUDY OVERVIEW}  
  
\subsection*{Study Design}  
}
```

The main document starts after the declaration of:

```
\begin{document }
```

After that, sections can be added, see [Section 11](#) on how to do this. Following all the main sections of the document, the reference list is added using the following command:

```
\references
```

The appendices are initialized using the command below. This causes all the counters for section, figure, table, equation, etc. to be reset for the appendices and an "A" to be added in front of the counter (e.g. "Section A1.1"). Appendices are added just like sections. See examples of appendices in [Appendix 1](#) or [Appendix 2](#).

```
\appendix
```

At the very end of the document you will find the LIST OF ITEMS that holds an overview of all included files in the report, their full absolute or relative paths, and modification date. This list is added using the following command:

```
\itemlist
```

The document is then ended by the following command:

```
\end{document }
```

## 9. PHARMTEX CLASS OPTIONS

The command `\documentclass` used in the first line of a TEX file supports various options for changing a PharmTeX document globally using square brackets as shown below. Each option for turning on a feature has a "no" equivalent, i.e. "appendix" vs. "noappendix". Unless indicated in the table, each option is turned on per default.

```
\documentclass [option1,option2,option3] {PharmTeX}
```

**Table 4. PharmTeX class options**

Option	Description	Default
(no)appendix	Show appendix.	
(no)coverpage	Show cover page.	
(no)qapage	Show quality assurance page.	
(no)toc	Show TABLE OF CONTENTS	
(no)lof	Show LIFT OF FIGURES	
(no)lot	Show LIST OF TABLES	
(no)glossary	Show ABBREVIATION AND DEFINITION OF TERMS (glossary)	
(no)synopsis	Show SYNOPSIS.	
(no)sigpage	Show APPROVAL PAGE.	
(no)samepagelinks	Allow same page hyperlinks in finalized document.	no
(no)glosshyper	Show glossary hyperlinks in finalized document.	
(no)webhyper	Show web hyperlinks in finalized document.	no
(no)artificid	Show artifact IDs.	
(no)artihyper	Show artifact ID hyperlinks.	no
(no)itemid	Show item IDs.	
(no)itemref	Show hyperlinks for items.	
(no)draftmode	Forced enabling/disabling of draft mode.	
(no)fast	Fast mode: Do not include any external files.	no
(no)synonly	Synopsis only mode (see <a href="#">Section 27</a> ).	no
(no)watermark	Show watermark.	
(no)logo	Show logo.	
(no)expdf	Include external PDF documents.	
(no)figcapbelow	Put figure captions below the figures (default above).	no
(no)eqnimg	Convert all equations to images (see <a href="#">Section 26</a> ).	no
(no)mathimg	Convert all inline math to images (see <a href="#">Section 26</a> ).	no
(no)convimg	Convert inline math to images only in some cases (see <a href="#">Section 26</a> ).	no

Inline table. PharmTeX class options. Unless indicated in the table, each option is turned on per default.

And exanple syntax could be:

```
\documentclass[notoc,nosynopsis]{PharmTeX}
```

## 10. SETTING THE PAPER SIZE AND MARGINS

PharmTeX supports two standard paper sizes: `isopaper` (A4, default) and `letterpaper` (US Letter). These can be added (e.g. `isopaper`) along with other options as described in [Section 9](#). In addition, the margins and distances to the header and footer can be modified using the following commands (should be placed before `\begin{document}`):

**Table 5. PharmTeX class settings for margins**

Command	Description
<code>\pmx{top}{2.54 cm}</code>	Distance from paper top edge to top of body text.

**Table 5. PharmTeX class settings for margins**

Command	Description
<code>\pmxbottom{2.54 cm}</code>	Distance from paper bottom edge to bottom of body text.
<code>\pmxleft{3.175 cm}</code>	Distance from paper left edge to left boundary of body text.
<code>\pmxright{2.54 cm}</code>	Distance from paper right edge to right boundary of body text.
<code>\pmxhead{0.25 cm}</code>	Distance from header text bottom to top of body text.
<code>\pmxfoot{1 cm}</code>	Distance from footer text top to bottom of body text.

Inline table. PharmTeX class settings for margins. The default value is shown for each command.

Other common units such as mm (millimeters), in (inches), and pt (points) can also be used.

## 11. SECTIONS

To set up sections in the report (like the present one), one uses the following code:

```
\section{SECTIONS} \label{sec:sections}
```

The `\label{sec:sections}` is a label for a cross reference which will be explained in [Section 12](#). The levels are subsection, subsection, paragraph, and subparagraph as shown below:

```
\subsection{Subsections}  
\subsubsection{Subsubsections}  
\paragraph{Paragraphs}  
\subparagraph{Subparagraphs}
```

### 11.1. Subsections

#### 11.1.1. Subsubsections

##### 11.1.1.1. Paragraphs

###### 11.1.1.1.1. Subparagraphs

In PharmTeX only levels up to and including subsubsection are added to the TABLE OF CONTENTS.

## 12. CROSS REFERENCES

Cross references to sections, figures, tables, etc. are handled by using the `\cref` command of the [cleveref](#) LaTeX package, e.g. by referencing the section labeled earlier:

```
See \cref{sec:sections}.
```

See [Section 11](#).

When using the `\cref` command, LaTeX/PharmTeX will automatically determine which text should precede the reference number, i.e. if it is a Section, Equation, Figure, etc.

There are various other ways to use do cross references, e.g. to pages without or with capitalization:

```
See \cpageref{sec:sections} for reference. \Cpageref{sec:sections}
has the same reference. See \nref{sec:sections}.
```

See [page 21](#) for reference. [Page 21](#) has the same reference. See [11](#).

Note that `\Cref` and `\cref` both produce uppercase first letter for Section, Figure, etc., as opposed to `\cpageref` and `\Cpageref`. To only show the reference number, use the `\nref` command.

**Note.** PharmTeX has been set up to disable hyperlinks leading to the same page as well as external hyperlinks when the document is compiled for finalization (using the F12 key in Texstudio).

### 13. GLOSSARY ENTRIES

Automatic glossary entries are supported using the LaTeX package [glossaries](#). A global glossary database can be downloaded along with a template document from [pharmtex.org](#), in this case called `glossary.tex`. But you can define your own glossary entries as you wish by editing the file or creating your own. You can also add additional glossary entries to the specific document you are working on, e.g. by adding the following code before

```
\begin{document}:
```

```
\newacronym{NONMEM}{NONMEM}{NONlinear Mixed Effects Modeling}
```

We can then use the abbreviation as follows:

```
The acronym \gls{NONMEM} is defined. Using it again prints \gls{
NONMEM} and a hyperlink.
```

The acronym NONlinear Mixed Effects Modeling (NONMEM) is defined. Using it again prints [NONMEM](#) and a hyperlink.

Besides adding the hyperlink, an entry is also added to the ABBREVIATION AND DEFINITION OF TERMS section of the document. Glossary entries can also capitalize letters and use plural forms of the acronyms. If the simple `\newacronym` was used to define it, the plural form will just put a "s" after the word, e.g.

```
\Glspl{CL} and \gls{CL}.
```

Clearances (CLs) and [CL](#).

In the above case, Clearance (defined in the global glossary) was capitalized due to the capital G in `\Glspl` and put in plural form to the "pl" in `\Glspl`. More advanced definitions of acronyms are also possible, e.g. with pharmacokinetics (defined in the

global glossary):

```
\newglossaryentry{PK}{type=\acronymtype,name={PK},description={  
pharmacokinetic},plural={PK},firstplural={pharmacokinetics (PK)},  
first={pharmacokinetic (PK)}}
```

We can then use it as shown below:

```
\Glspl{PK} and \gls{PK}.
```

Pharmacokinetics (PK) and [PK](#).

In this case, we did not want an "s" after PK in the abbreviation in the plural form and defined it not to have this using `\newglossaryentry`. Had we used `\newacronym`, this plural "s" would have been applied automatically.

**Note.** PharmTeX has been set up to disable hyperlinks for glossary entries when the document is compiled for finalization (using the F12 key in Textstudio).

## 14. CITATIONS

Citations are supported using the [natbib](#) package in LaTeX. They can be added using the following code:

```
See \cite{Albus2002}.
```

See [\[1\]](#).

The key `Albus2002` should then be defined in the BibTeX database (BIB file) that can be edited using JabRef (by pressing F5 in Textstudio) or indeed any reference manager that supports BibTeX (e.g. [Mendeley](#)). Multiple reference entries will be compressed when possible:

```
See \cite{Hughes2003, Javitt2006, Braak2003}.
```

See [\[2–4\]](#).

Once used, entries are added to the "References" section of the report. The formatting of this list is determined by the BST formatting file loaded using `\pmxbst`. In the template documents on [pharmtex.org](#), `pmx.bst` is used. Other formatting files can also be used.

## 15. FOOTNOTES

Footnotes can be added as shown below:

```
This is a footnote\footnote{Testing a footnote}.
```

This is a footnote<sup>1</sup>.

Note the actual footnote in the bottom of this page. For footnotes inside a table, i.e. table notes that lead to the table description, see [Section 20.4](#).

## 16. BULLET POINTS AND NUMBERED ITEMS

This is an example of bullet points:

```
\begin{itemize}
  \item Text.
  \item Text.
\end{itemize}
```

- Text.
- Text.

For numbered items, we can use:

```
\begin{enumerate}
  \item Text.
  \item Text.
\end{enumerate}
```

1. Text.
2. Text.

To add additional levels, we can do:

```
\begin{itemize}
\item Text.
  \begin{itemize}
    \item Text.
    \item Text.
  \end{itemize}
\item Text.
\end{itemize}
```

- Text.
  - Text.
  - Text.
- Text.

This procedure is the same for bullets. Since the `itemize` and `enumrate` environments

---

<sup>1</sup>Testing a footnote



are native LaTeX commands, they can be customized as per the wishes of the user (symbols, levels, etc.). We refer to the vast online LaTeX documentation for such customizations.

## 17. EQUATIONS

Equations can be created in LaTeX in a multitude of ways, but the recommended approach in PharmTeX is as shown below using the `align` environment from the `amsmath` package:

```
\begin{align}
y = \sqrt{x} \quad \label{eqn:sqrt}
\end{align}
```

The equation can then be referenced using:

$$y = \sqrt{x} \tag{1}$$

The equation can then be referenced using:

```
See \cref{eqn:sqrt}.
```

See [Equation 1](#).

Mathematical symbols and equations need to be used within a mathematical mode environment like `align`. Here are a few basics:

```
\begin{align}
a_{1} = \frac{x^{2} + 1}{3} + \left( \sqrt{\frac{x}{z}} \right)
\end{align}
```

$$a_1 = \frac{x^2 + 1}{3} + \left( \sqrt{\frac{x}{z}} \right) \tag{2}$$

You can also create equation arrays with the "=" symbol aligned:

```
\begin{align}
\frac{dn_D}{dt} &= -k_D n_D \quad \label{eqn:modela} \quad \\
\frac{dn_A}{dt} &= k_D n_D - k_A n_A \quad \\
\frac{dn_C}{dt} &= k_A n_A - k_E n_C
\end{align}
```

$$\frac{dn_D}{dt} = -k_D n_D \tag{3}$$

$$\frac{dn_A}{dt} = k_D n_D - k_A n_A \tag{4}$$

$$\frac{dn_C}{dt} = k_A n_A - k_E n_C \quad (5)$$

You can label the line(s) that you needed to reference. If you do not want an equation number for a given line, use the PharmTeX shorthand `\nn` on that given line, in this case the middle one, just before the newline command (`\n`):

```
\begin{align}
\frac{dn_D}{dt} &= -k_D n_D \n
\frac{dn_A}{dt} &= k_D n_D - k_A n_A \nn \n
\frac{dn_C}{dt} &= k_A n_A - k_E n_C
\end{align}
```

$$\frac{dn_D}{dt} = -k_D n_D \quad (6)$$

$$\frac{dn_A}{dt} = k_D n_D - k_A n_A$$

$$\frac{dn_C}{dt} = k_A n_A - k_E n_C \quad (7)$$

You can also create a subequation array:

```
\begin{subequations}
\label{eqn:full}
\begin{align}
\frac{dn_C}{dt} &= -k_A n_C (\alpha - n_S) \label{eqn:line1} \n
\alpha_{12} &= 3x^2 + 4x - 1 \label{eqn:line2}
\end{align}
\end{subequations}
```

$$\frac{dn_C}{dt} = -k_A n_C (\alpha - n_S) \quad (8a)$$

$$\alpha_{12} = 3x^2 + 4x - 1 \quad (8b)$$

Here, both the individual equations and the system as a whole can be referenced using:

```
See \cref{eqn:line1}, \cref{eqn:line2}, and \cref{eqn:full}.
```

See [Equation 8a](#), [Equation 8b](#), and [Equation 8](#).

If you have a very long equation, you may want to split it over two or more lines. This can be achieved using the following code. Note that the behaviour of `\n` inside the `split` environment is to break the long equation line at that particular spot. Also, both equation lines now have the same equation number positioned between the two lines.

```
\begin{align}
\begin{split}
p(x) = 3 \ x \ x^6 + 14 \ x \ x^5y + 590 \ x \ x^4 \ x \ y^2 + 19 \ x \ x^3 \ x \ y
^3 \ \
- 12 \ x \ x^2 \ x \ y^4 - 12 \ x \ x \ x \ y^5 + 2 \ x \ y^6 - a^3 \ x \ b^3
\end{split}
\end{align}
```

$$p(x) = 3x^6 + 14x^5y + 590x^4y^2 + 19x^3y^3 - 12x^2y^4 - 12xy^5 + 2y^6 - a^3b^3 \quad (9)$$

We can use inline math in between your text encapsulated in \$ symbols, e.g.:

```
See $n_{D}$ and $c_C = \frac{n_C}{v}$. Or $c_C = \dfrac{n_C}{v}$.
```

See  $n_D$  and  $c_C = \frac{n_C}{v}$ . Or  $c_C = \dfrac{n_C}{v}$ .

In this case, compare the effect of `\frac` and `\dfrac` in the size of the fraction. In \$ math mode, many symbols such as square root and fractions are printed smaller compared to mathematical environments such as `align`, unless explicitly changed such as using the command `\dfrac`.

We refer the user to the vast online documentation for creating mathematical symbols and equations in LaTeX, e.g. [here](#).

## 18. EXTERNAL HYPERLINKS

Below are an example of showing a hyperlinked url and hyperlinked text:

```
\href{http://www.mathworks.com}{MATLAB} vs. \url{http://www.
mathworks.com}.
```

**MATLAB** vs. <http://www.mathworks.com>.

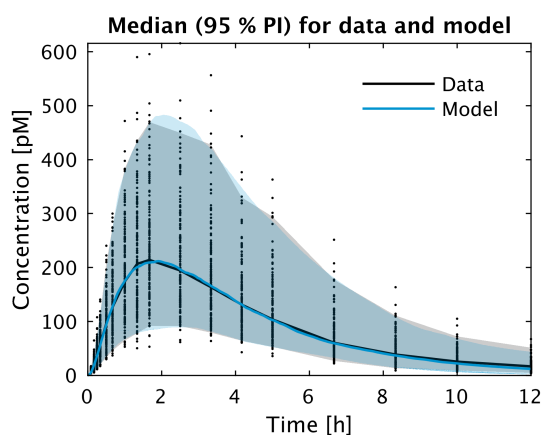
## 19. FIGURES

### 19.1. Figures from Image Files

A figure can be loaded from an image file, in this case PNG, as shown below:

```
\pmxfigure
{figure.png} % filename
{fig:vpc} % cross reference key
{Visual predictive check} % caption
{Figure description.} % description
{scale=0.9} % figure options, e.g. scaling
```

**Figure 3. Visual predictive check**



[Item 6](#). Figure description.

In this case the figure is located in the same folder as the report itself. However, instead of doing this, PharmTeX can be pointed to an absolute location, e.g. `C:/path/vpc.png` on Windows or `/path/vpc.png` on Linux, or a relative path, e.g. `../path/vpc.png` on both Windows and Linux. When using relative paths, ensure that the paths are made in reference to the main TEX file.

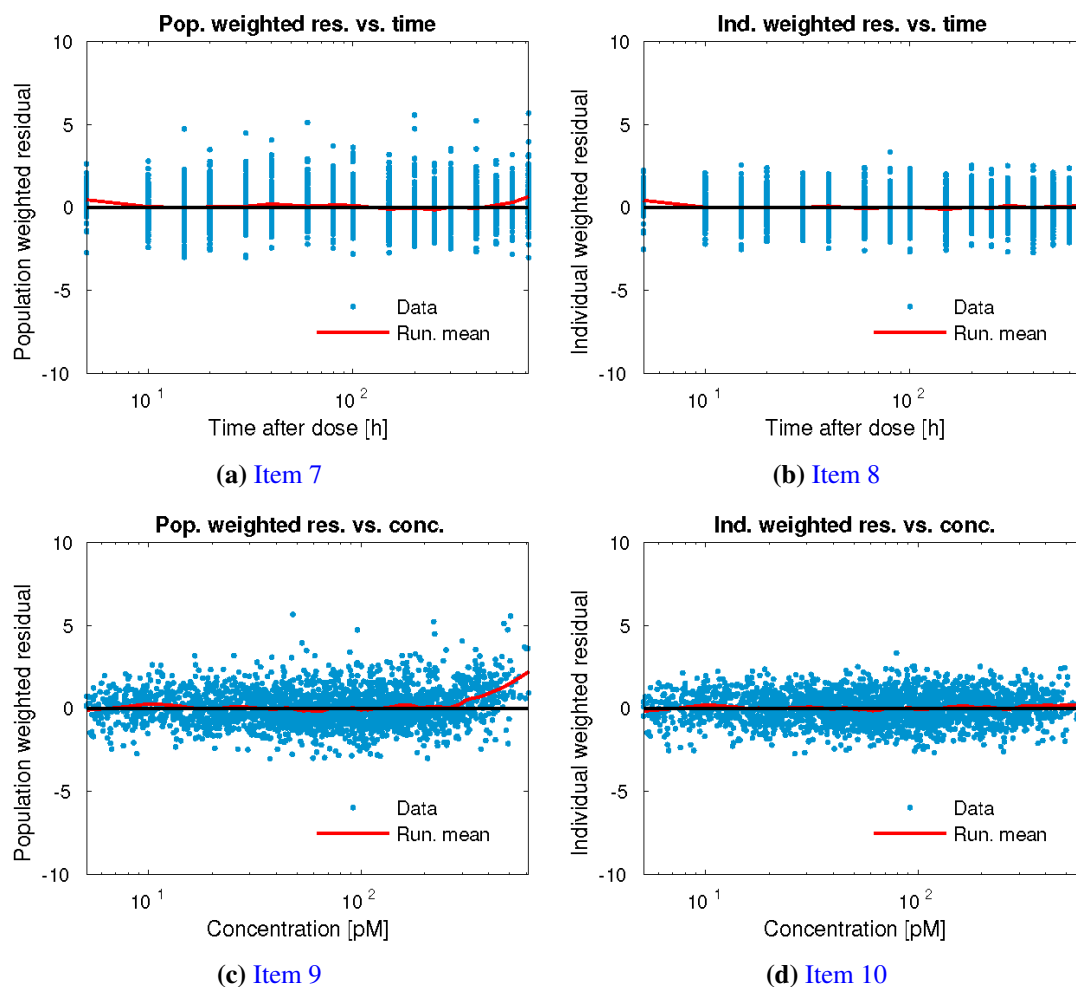
Figure options as supported by the `\includegraphics` LaTeX command can be used, e.g. `scale`, `width`, and `height`. An item number is automatically added to the description below the figure. Clicking the hyperlink will take you to the LIST OF ITEMS in the end of the document that holds an overview of all included files in the report, their full absolute or relative paths, and modification date.

## 19.2. Figure Arrays

Figures from multiple image files can be combined into a single PharmTeX figure with subfigures:

```
\pmxfigure
{ % filenames: & = new column, | = new row
residual1.png & residual2.png |
residual3.png & residual4.png
}
{ % reference keys: first one for whole figure, then for each part
fig:residuals
[fig:residual1] & [fig:residual2] |
[fig:residual3] & [fig:residual4]
}
{Residual plots} % caption
{Residual plots for final model.} % description
{ % figure options for each part
[scale=0.9] & [scale=0.9] |
[scale=0.9] & [scale=0.9]
}
```

**Figure 4. Residual plots**



Residual plots for final model.

Note that individual descriptions for each subfigure (subcaptions) are not supported in PharmTeX in order to save space between figure lines (it is possible to implement them). The figure array has an item number for each file place under the respective figure. They are cross reference keys for the figure as a whole and for each subfigure that can be used in different ways as shown below:

```
See \cref{fig:residuals} vs. \cref{fig:residual1} vs. \subref{fig:residual1}.
```

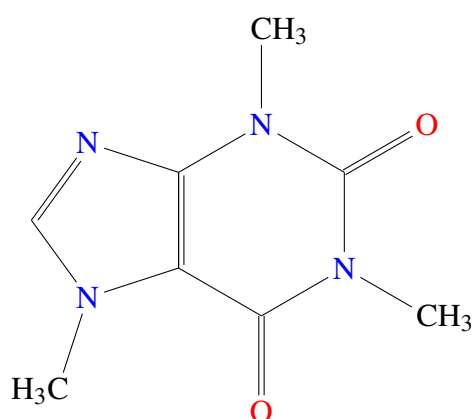
See [Figures 4](#) vs. [Figure 4a](#) vs. [\(a\)](#).

### 19.3. Inline-Produced Figures

An example of an inline figure, in this case produced using the [chemfig](#) LaTeX package is shown below:

```
\pmxtexfigure
{fig:chem} % cross reference key
{Example of inline figure} % caption
{Figure description.} % description
{ % LaTeX figure commands, here code from the chemfig package
\chemfig{
  H_3C-[:72]{\color{blue}N}
    *5(-
      *6(-({\color{red}O})-{\color{blue}N}(-CH_3)-({\color{red}O}
        )-{\color{blue}N}(-CH_3)-=)
      --{\color{blue}N}=-) }
}
```

**Figure 5. Example of inline figure**



Inline figure Figure description.

Note, for the above example to work, the chemfig package should be loaded before `\begin{document}` using the following command:

```
\usepackage{chemfig}
```

## 20. TABLES

### 20.1. Tables from Delimiter Files

Tables can be loaded directly from delimiter-separated files, e.g. using comma as the separator (CSV file). In this case, the file `table.csv` contains 4 rows and 5 columns and it can be loaded using the code below:

```
\pmxtable
[1] % optional: number of first rows to include as header
{table.csv} % filename
{comma} % delimiter
{C C C C C} % justification for each column
{tab:table} % cross reference key
{Table from file} % caption
{Table description.} % description
{} % replace header with a new one
```

**Table 6. Table from file**

a	b	c	d	e
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

Item 11. Table description.

The first argument [1] tells PharmTeX to use the first row of the file as the header. The delimiter is chosen as comma. Other supported delimiters are semicolon, tab, and pipe (|). Classical LaTeX-style tables using & and \can be loaded from delimiter files by simply using tex as the delimiter specification (see Section 20.2 on how to create such tables).

The justification is chosen per column as C (center) for all columns in this case, where the capital letter tells PharmTeX to share the column width equally between the columns to fill the page width. Other supported justifications are L (left) and R (right). The final argument, here left empty, allows the user to change the header of the table compared to what is in the table file, see example below:

```
\pmxtable
[1] % optional: number of first rows to include as header
{table.csv} % filename
{comma} % delimiter
{C C C C C} % justification for each column
{tab:table1} % cross reference key
{Table from file with header substitution} % caption
{Table description.} % description
{A & B & C & D & E} % replace header with a new one
```

**Table 7. Table from file with header substitution**

A	B	C	D	E
1	2	3	4	5
6	7	8	9	10

**Table 7. Table from file with header substitution**

A	B	C	D	E
11	12	13	14	15

[Item 12.](#) Line 1 substituted. Table description.

The header is inserted by separating each column by the symbol &. Note that upon substitution, PharmTeX automatically prints in the table description that the first line of the CSV file has been substituted.

If the column justification argument is left empty, PharmTeX will include all columns in the table file as left justified (L) as shown below:

```
\pmxtable
[1] % optional: number of first rows to include as header
{table.csv} % filename
{comma} % delimiter
{} % justification for each column, here left blank
{tab:table1a} % cross reference key
{Table from file with default justification} % caption
{Table description.} % description
{} % replace header with a new one
```

**Table 8. Table from file with default justification**

a	b	c	d	e
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

[Item 13.](#) Table description.

The column widths of the table can be scaled relative to each other by putting a scaler in curly brackets for the column(s) that should be scaled:

```
\pmxtable
[1] % optional: number of first rows to include as header
{table.csv} % filename
{comma} % delimiter
{C{2} C C{0.5} C c} % justification for each column
{tab:table2} % cross reference key
{Table from file with column width scaling} % caption
{Table description.} % description
{} % replace header with a new one
```



**Table 9. Table from file with column width scaling**

a	b	c	d	e
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

Item 14. Table description.

Notice that for the last column, a lowercase `c` was used. This tells PharmTeX to make the column as "tight" as possible around the text in that column. Be cautious when using lowercase justification, since if a lot of text is present in the column, it may end up extending beyond the page width. As opposed to this, uppercase justification letters will wrap the text to match the column width. Coefficients for the scaling can also contain units, e.g. `C{1in}` (inches), `C{2cm}` (centimeters), or `C{5mm}` [millimeters). All units supported by LaTeX for this purpose are supported by PharmTeX.

Columns from files can also be swapped or omitted using PharmTeX code. A specific column from the delimiter file can be put in at a specific point using square brackets as shown below:

```
\pmxtable
[1] % optional: number of first rows to include as header
{table.csv} % filename
{comma} % delimiter
{C C[5] C C} % justification for each column
{tab:table3} % cross reference key
{Table from file with column swapping} % caption
{Table description.} % description
{} % replace header with a new one
```

**Table 10. Table from file with column swapping**

a	e	c	d
1	5	3	4
6	10	8	9
11	15	13	14

Item 15. Columns [1 5 3 4] out of 5. Table description.

In this case, only four columns are included, and the second column in the table is the fifth column from the delimiter file. PharmTeX automatically prints which columns have been imported in the description of the table. Scalers and column swaps can be combined in either order, i.e. as for instance `C[5]{2}` or `C{2}[5]` which are equivalent for any given column.

LaTeX/PharmTeX code can also be interpreted from delimiter files. In this case, more advanced tables can be created, by putting the term `tex` before the delimiter specification as shown below:

```
\pmxtable
[1] % optional: number of first rows to include as header
{parameters.csv} % filename
{texcomma} % delimiter
{L{2} L L L} % justification for each column
{tab:parameters} % cross reference key
{Table from file with LaTeX code} % caption
{Parameter overview created from the results.} % description
{} % replace header with a new one
```

**Table 11. Table from file with LaTeX code**

Parameter	Expected	95 % CI	%RSE%
$\theta_{KD}$ [min <sup>-1</sup> ]	0.00526	[0.00501; 0.00551]	2.4 %
$\theta_{KA}$ [min <sup>-1</sup> ]	0.0154	[0.0146; 0.0163]	2.7 %
$\theta_{KE}$ [min <sup>-1</sup> ]	0.0987	[0.0930; 0.104]	2.9 %
$\theta_V$ [L]	15.3	[14.6; 15.9]	2.2 %

Item 16. Parameter overview created from the results.

The input file `parameters.csv` is a text file that contains the following:

**Listing 1. The input file parameters.csv**

```
Parameter, Expected, 95 \% \gls{CI}, \gls{RSE}\%
$\theta\_text{KD}$ [min$^{-1}$], 0.00526, [0.00501; 0.00551], 2.4 \%
$\theta\_text{KA}$ [min$^{-1}$], 0.0154, [0.0146; 0.0163], 2.7 \%
$\theta\_text{KE}$ [min$^{-1}$], 0.0987, [0.0930; 0.104], 2.9 \%
$\theta\_text{V}$ [L], 15.3, [14.6; 15.9], 2.2 \%
```

If `texcomma` was not used, but rather just `comma`, the LaTeX code would not have been compiled and instead just printed raw as shown below:

```
\pmxtable
[1] % optional: number of first rows to include as header
{parameters.csv} % filename
{comma} % delimiter
{L{2} L L L} % justification for each column
{tab:raw} % cross reference key
{Table from file with LaTeX code (no compilation)} % caption
{Parameter overview created from the results (LaTeX code not
compile).} % description
{} % replace header with a new one
```

**Table 12. Table from file with LaTeX code (no compilation)**

Parameter	Expected	95 \% \gls {CI}	\gls {RSE}\%
$\theta\_text {KD}$ \$ [min\$^{-1}\$]	0.00526	[0.00501; 0.00551]	2.4 \%

**Table 12. Table from file with LaTeX code (no compilation)**

Parameter	Expected	95 \% \gls {CI}	\gls {RSE}\%
$\theta_{\text{KA}}$ [min <sup>-1</sup> ]	0.0154	[0.0146; 0.0163]	2.7%
$\theta_{\text{KE}}$ [min <sup>-1</sup> ]	0.0987	[0.0930; 0.104]	2.9%
$\theta_{\text{V}}$ [L]	15.3	[14.6; 15.9]	2.2%

Item 17. Parameter overview created from the results (LaTeX code not compile).

## 20.2. Inline Tables

A table can be created inside the TEX document as shown below using the `\pmxtable` command:

```
\pmxtable
[1] % optional: number of first rows to include as header
{C C C C C} % justification for each column
{tab:inline} % cross reference key
{Inline table} % caption
{Table created in document.} % description
{ % table code
a & b & c & d & e \\
1 & 2 & 3 & 4 & 5 \\
6 & 7 & 8 & 9 & 10 \\
11 & 12 & 13 & 14 & 15
}
```

**Table 13. Inline table**

a	b	c	d	e
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

Inline table. Table created in document.

## 20.3. Multicolumn and Multirow Tables

Below is an example of an advanced mixed multicolumn and multirow table, i.e. a table which cells spanning multiple columns of rows. We will not go through the details of these types of table, but will instead refer the user to the extensive online tutorials available.

NOTE. The option `{=}` for the `\multicolumn` and `\multirow` commands is native to PharmTeX and essentially means that the commands should respect the width and height settings from the main table, in this case `{L{4} L{2.5} L{1} L{2.5} L{1}}`. It is strongly recommended to always use this option instead of alternatives like `{*}` that you will find online.

```
[2]
{L{4} L{2.5} L{1} L{2.5} L{1}}
{tab:multitable}
{Multicolum/row table} % caption
{Multicolum/row table.} % description
{
\multirow[t]{2}{=}{\textbf{Parameter}} & \multicolumn{2}{=}{\textbf{Final Parameter Estimate}} & \multicolumn{2}{=}{\textbf{Magnitude of Variability}} \\
& \textbf{\%RSE} & \textbf{Final Estimate} & \textbf{\%RSE} \\
Ka (1/hr) & 0.587 & 2.60 & 21.5 \%CV & 19.4 \\
CL (L/hr) & 22.2 & 0.892 & 17.6 \%CV & 7.49 \\
Vc (L) & 52.6 & 4.12 & 27.2 \%CV & 21.2 \\
Q (L/hr) & 43.2 & 1.92 & 36.4 \%CV & 9.16 \\
Vp (L) & 438 & 1.20 & 22.8 \%CV & 8.28 \\
RV CV & 0.0149 & 2.04 & \multirow[t]{2}{=}{29300 - 12.5 \%CV\newline F [0.01 - 100]} & \multirow[t]{2}{=}{NA} \\
RV Add & 8.57 & 37.4 \\
\multicolumn{5}{=}{Minimum Value of the Objective Function = 70246.07}
}
```

**Table 14. Multicolum/row table**

Parameter	Final Parameter Estimate		Magnitude of Variability	
	Population Mean	%RSE	Final Estimate	%RSE
Ka (1/hr)	0.587	2.60	21.5 %CV	19.4
CL (L/hr)	22.2	0.892	17.6 %CV	7.49
Vc (L)	52.6	4.12	27.2 %CV	21.2
Q (L/hr)	43.2	1.92	36.4 %CV	9.16
Vp (L)	438	1.20	22.8 %CV	8.28
RV CV	0.0149	2.04	29300 - 12.5 %CV F [0.01 - 100]	NA
RV Add	8.57	37.4		
Minimum Value of the Objective Function = 70246.07				

Inline table. Multicolum/row table.

## 20.4. Table Notes

Table notes, i.e. notes that lead to the table description (similar to footnotes), can be inserted into a table using the code below. The use of `\tref` will produce an alphanumeric superscript reference and place the note text where `\tnote` is used in the description.

```
\pmxtextable
[1] % optional: number of first rows to include as header
{C C C C C} % justification for each column
{tab:tnote} % cross reference key
{Inline table with table note} % caption
{Table created in document. \tnote{note:mytnote}{This is a table
note.}.} % description
{ % table code
a & b & c\tref{note:mytnote} & d & e \\
1 & 2 & 3 & 4 & 5 \\
6 & 7 & 8 & 9 & 10 \\
11 & 12 & 13 & 14 & 15
}
```

**Table 15. Inline table with table note**

a	b	c	d	e
1	2	3	4	5
6	7	8 <sup>a</sup>	9	10
11	12	13	14	15

Inline table. Table created in document. <sup>a</sup> This is a table note.

## 20.5. Tables Based on Image Files

A table can be created from an image file, e.g. PDF, PNG, JPG, and GIF, as shown below using the `\pmxfigtable` command. The syntax is similar to `\pmxfigure`:

```
\pmxfigtable
{table.pdf} % filename
{tab:figtable} % cross reference key
{Table created from a PDF file} % caption
{Table created from a PDF file.} % description
{scale=0.9} % options, e.g. scaling of the table figure
```

**Table 16. Table created from an image file**

a	b	c	d	e
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15

[Item 18](#). Table created from an image file.

## 20.6. Excel2LaTeX Tables

In the PharmTeX software bundle for Windows, a tool called [Excel2LaTeX](#) is included. It is located in the bundle folder, which on Windows 10/11 is

C:\Users\USERNAME\pharmtex\excel2latex\Excel2LaTeX.xla. To start using it, double-click this file and then click "Enable Macros" in the warning that comes up. Then close Excel. From this point on, an "Add-ins" pane should appear in Excel. If not,

try to open your specific Excel table file while having

C:\Users\USERNAME\pharmtex\excel2latex\Excel2LaTeX.xla open at the same time the first few times you use it.

You can create an Excel table or open one and then select/highlight the table you want to convert (relevant cells only). Then click the first button in the "Add-ins" pane (if you hover the mouse over it, it will say "Excel2LaTeX: Convert Table to LaTeX") and a panel will appear showing the LaTeX code for your table. This code can be copied-and-pasted into a `\pmxtable` command as shown below, where an example table was created.

```
\pmxtable
[1] % optional: number of first rows to include as header
{C C C C C} % justification for each column
{tab:excel} % cross reference key
{Excel2LaTeX table} % caption
{Excel2LaTeX table description.} % description
{ % table code
% Table generated by Excel2LaTeX from sheet 'Sheet1'
\begin{table}[htbp]
  \centering
  \caption{Add caption}
  \begin{tabular}{ccccc}
    \textbf{a} & \textbf{b} & \textbf{c} & \textbf{d} & \textbf{e} \\
    \\
    1 & 2 & 3 & 4 & 5 \\
    6 & 7 & \multicolumn{2}{c}{\multirow{2}[0]{*}{8}} & 10 \\
    \\
    11 & 12 & \multicolumn{2}{c}{} & 15 \\
    16 & 17 & 18 & 19 & 20 \\
  \end{tabular}%
  \label{tab:addlabel}%
\end{table}%
}
```

**Table 17. Excel2LaTeX table**

<b>a</b>	<b>b</b>	<b>c</b>	<b>d</b>	<b>e</b>
1	2	3	4	5
6	7	8		10
11	12			15
16	17	18	19	20

Inline table. Excel2LaTeX table description.

You can copy the entire output from the Excel2LaTeX tool, since PharmTeX is set up to ignore the bits it does not need. In this case, a complex table with headers in bold and merged columns and rows was created, and Excel2LaTeX takes care of using the otherwise complex LaTeX packages `multicol` and `multirow` in combination.

Before you export your table, please ensure that you have justified all columns as you

prefer, particularly if you have merged cells. This will set the `\multicolumn` and `\multirow` commands correctly when exporting. You still need to set the justification for the `\pmxtable` command to your preference as well.

NOTE. You can consider replacing the `{*}` option with `{=}`, a native PharmTeX option that makes `\multicolumn` and `\multirow` respect the width and height settings from the main table, see [Section 20.3](#).

## 20.7. Font Size and Whitespace

You can change the font size for a given table by putting the following code right before the `\pmxtable` or `\pmxtable` command:

```
\pmxfont{\scriptsize} % 8 pt (default for code listings)
```

Or:

```
\pmxfont{\footnotesize} % 10 pt (default for tables)
```

Also, issuing the following right before a table will reduce the whitespace around the table somewhat:

```
\pmxtight
```

Note. The above changes applies to the current table only.

## 21. INCLUDING A CHILD DOCUMENT

You can include the contents of one PharmTeX document into another using the `\pmxinclude` command. This is useful when working as multiple authors on a document. Everything between `\begin{document}` and `\references` in the child document will be included in the parent document at the point where `\pmxinclude` is used.

```
\pmxinclude{child.tex}
```

This is text from the child document, including a bit of math mode  $\sqrt{x}$ .

You should remember to copy-and-paste all the child document's references into the database of the parent document. Use the handy tool in JabRef under "Search" -> "Find duplicates" or "Resolve duplicate BiBTeX keys" to resolve duplicate article/key issues.

## 22. INCLUDING A TEXT FILE INLINE

If you have a key result from an analysis, it can be printed into a text file and included using the `\pmxinput` command as shown below;

```
The result is \pmxinput{result.txt}.
```

The result is 42 .

When this is done, an item number is shown in a footnote for the text file in question and the value in the text file is included at the point where the `\pmxinput` command is used.

### 23. R KNITR SUPPORT

R code can be run from within a PharmTeX document using the built-in `knitr` support. Just add the following line to the very first line of your TEX document, changing the path to match your local R installation:

```
%rpath=C:\Program Files\R\R-X.X.X\bin\x64
```

Then `\Sexpr` can be used to evaluate e.g.  $\pi$ :

```
The value of  $\pi$  is \Sexpr{pi}.
```

The value of  $\pi$  is 3.1415927.

An R code chunk can be executed (remember to give it a new ID for each chunk, e.g. `<<Run1>>`, `<<Run2>>`, etc.):

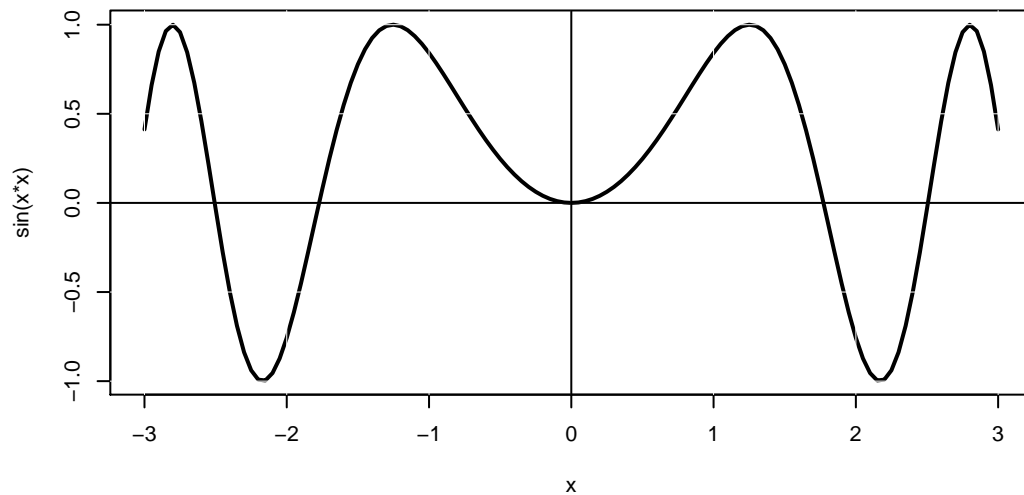
```
<<Run1>>=  
xpts <- seq(-3, 3, .05)  
pdf("myplot.pdf", width=5.6, height=2.8, pointsize=8)  
par(mai=c(.6, .6, .2, .2))  
plot(xpts, sin(xpts*xpts), type="l", lwd=2, xlab="x", ylab="sin(x*  
x) ")  
grid(col="white", lty=1, lwd=.2)  
abline(h=0, v=0)  
@
```

The chunk prints a figure that can then be included using:

```
\pmxfigure  
{myplot.pdf}  
{fig:myfigure}  
{R figure}  
{Figure generated from R code within this document.}  
{}
```



**Figure 6. R figure**



[Item 20](#). Figure generated from R code within this document.

## 24. DEFINING CUSTOM COMMANDS

In LaTeX, it is possible to define custom commands. Below is an example of defining a command without any input arguments:

```
\newcommand{\pmx}{pharmacometrics\xspace}
```

It can be used, e.g. as:

```
By using \pmx we can quantify pharmacology.
```

By using pharmacometrics we can quantify pharmacology.

The `\xspace` ensures that a space is present after `\pmx` when needed, i.e. as above. The space will not be present if e.g. a period (.) is used right after `\pmx`.

Custom commands can also be created to take input arguments. An example is shown below:

```
\newcommand{\printstuff}[2]{#1 is #2\xspace}
```

It can be used as shown below:

```
I think \printstuff{pharmacometrics}{awesome}.
```

I think pharmacometrics is awesome.

The `[2]` defines the number of input arguments, and they are used as `#1` and `#2`.

## 25. RUNNING PERL CODE IN PHARMTEX

PharmTeX is based on perltex, meaning that Perl code can be executed as part of the LaTeX compilation steps. This is handled in the background for the custom PharmTeX commands, but LaTeX/PharmTeX users can access this capability from within a document by invoking `\perlnewcommand`. An example of this is shown below:

```
\perlnewcommand{\add}{2}{  
$v0 = $_[0];  
$v1 = $_[1];  
$s = $v0 + $v1;  
return "The sum of $v0 and $v1 equals $s";  
}
```

The following code would then yield the output shown below:

```
\add{1}{2}
```

The sum of 1 and 2 equals 3

Perl obviously has much more advanced capabilities than simple algebra, for instance advanced text parsing, something that is heavily utilized in the background in PharmTeX. The native LaTeX command `\newcommand` also allows for definition of custom macros, but is very limited in its capabilities compared to `\perlnewcommand`.

## 26. CONVERSION TO OR FROM MICROSOFT WORD FORMAT

### 26.1. Conversion to Microsoft Word from the Finalized PDF

For various reasons, it may be needed to convert the PharmTeX document to Microsoft Word format. It could be that elements of the PharmTeX report will be incorporated into e.g. and Investigator's Brochure and that the authors are not using PharmTeX for that document. In this case, it is the experience of the author of this document that conversion of a PharmTeX report to Word format is done most accurately by converting the finalized PDF document using Adobe Acrobat <https://acrobat.adobe.com/us/en/acrobat.html>. The current version, Adobe Acrobat DC, has in the author's experience produced Word conversions that are for all practical purposes identical to the PDF output from PharmTeX. The conversion is done by opening the PDF in Adobe Acrobat DC and then clicking "File" -> "Export To" -> "Microsoft Word" -> "Word Document" (not the "Word 97-2003 Document").

### 26.2. Conversion of Advanced Equations and Mathematical Symbols

If advanced equations or inline math that go beyond simply greek letters and subscript/superscript have been used in PharmTeX, this conversion can be further stabilized by converting **all** equations and inline math to images using the PharmTeX options "eqnimg" and "mathimg", respectively. If you only want to convert **some** select troublesome inline math element(s) you can use the "convimg" option instead of "mathimg" and use the command `\convimg{some troublesome math}` instead of

$\$$ some troublesome math $\$$  for such math elements. The option "conving" can also be used reversely to convert **everything but** the inline math contained in `\conving`. In this case the options "noconving,mathimg" would be used in conjunction.

Use of one of these options will replace the equations (eqnimg) and/or inline math (all for "mathimg", math inside `\conving` command only for "conving") with images compiled separately using PharmTeX, ensuring that the Word conversion will not misrepresent the mathematical symbols if they are unknown to Word. **These options will only be applied by PharmTeX when using the F12 button to finalize** the document and the syntax is as follows:

```
\documentclass[eqnimg]{PharmTeX}
```

In the above case two finalized documents will be produced by PharmTeX. The formal finalized PDF with no equation or math conversions named according to the `\docpdfname` command and another finalized PDF with the addition of "-word" to the filename where equations (from environments such as `\begin{align}`, `\end{align}`) and/or inline math elements have been converted to images and included in place of each equation. Equation numbers will be unchanged and still referable and most LaTeX equation environments. When "eqnimg" and "mathimg"/"conving" are used together, only one combined "-word" PDF document will be produced. If you want to convert only some of the inline math and not other, you can alternate between  $\$$  which is converted and `\ensuremath` which is not converted, or you can combine the options "mathimg" and "noconving" as described above.

It should be noted that for each instance of an equation environment (for "eqnimg") and inline math environment (for "mathimg" and "conving") a separate PharmTeX compilation is triggered in the background. This means that if you have many instances of either equations or inline math, the overall compilation process can be significantly longer than when not using these options. It is, however, the experience of the author of this user guide that one can usually get away with using only the "eqnimg" option, potentially combined with the "conving" option for a few inline math instances, since it is in the equation environments that the most advanced symbols and math, included fractions and roots, appear. The "-word" document will have its own log file.

### 26.3. Other Conversion Tools

There are several tools that allow for conversion from LaTeX source code or PDF documents to Word format or the other way. One popular generic tool is Pandoc (<https://pandoc.org>), the use of which will be described here. The author of this user guide has had some success with using GrindEQ (<https://grindeq.com>) to convert LaTeX source code directly to Word, or converting Word documents to LaTeX source code. This includes equations and mathematical symbols. The process is by no means perfect, but it can be used along with strict user control and manual editing of conversion errors to save a lot of time.

## 27. COMPILING THE SYNOPSIS AS A SEPARATE DOCUMENT

In some cases it is practical to have the synopsis elements of the document as a separate document and it is for this purpose the the "synonly" option for PharmTeX has been created. When invoked using the syntax below the synopsis part of the document is compiled into a separate PDF file with the addition of "-synopsis" to the filename defined by the `\docpdfname` command will be created. The normal combined document, including both the main body and the synopsis, is created as usual. The synopsis document will have its own References and List of Items depending on the citations and included items in the synopsis part, respectively. No Table of Contents, List of Tables, or List of Figures are included the separate synopsis document, but it will have its own glossary.

When used in conjunction with either one or both of the "eqning" and "mathimg" options, four separate PDF will be compiled, containing 1) the combined document with no equation or math conversions, 2) the "-word" version with any equation or math conversions included, 3) the "-synopsis" version with the synopsis only and no equation or math conversions, and 4) the "-synopsis-word" version with the synopsis only with any equation or math conversions included. The synopsis document will also have its own log file.

## 28. INCLUDING NLMIXR OUTPUT IN PHARMTEX

It is possible to include figures and tables automatically from an `nlmixr` [5] run automatically into PharmTeX. The software `nlmixr` ([Nonlinear Mixed Effects Models in Population Pharmacokinetics and Pharmacodynamics](#)) is an open-source R package for nonlinear mixed effects models in population PK and pharmacodynamic (PD), see <http://nlmixr.org> and <https://nlmixrdevelopment.github.io> for more info.

The following PharmTeX syntax will read all output in the `runX` folder containing `nlmixr` output tables and figures:

```
\nlmixr{runX} % print the file nlmixr-path-to-runX.tex  
\pmxinclude{nlmixr-path-to-run1.tex} % include the above file
```

Here, `runX` should be replaced by the relative or absolute path to the Nonlinear Mixed Effects Models in Population Pharmacokinetics and Pharmacodynamics (`nlmixr`) run folder(s) from which you want to include output tables and figures. One example could be that you have your PharmTeX report folder such that the `nlmixr` "analysis" folder containing the "run1", "run2", ..., individual run folders are one folder up from the PharmTeX report folder. In this case, one could invoke:

```
\nlmixr{../analysis/run1} % print file nlmixr-analysis-run1.tex  
\pmxinclude{nlmixr-analysis-run1.tex} % include the above file
```

This would create PharmTeX code for all tables and figures in the `../analysis/run1` folder and print the code into the file `nlmixr-analysis-run1.tex` into the PharmTeX report folder. The command `\pmxinclude{nlmixr-analysis-run1.tex}` simply

includes this files into the report to be compiled. **Note.** If the user wants to edit the generated TEX file, the `nlmixr` command should be removed or commented out as the generated file will be overwritten when recompiled. Alternative, the desired code from the generated TEX file can be copied-and-pasted into the main document and compiled from there.

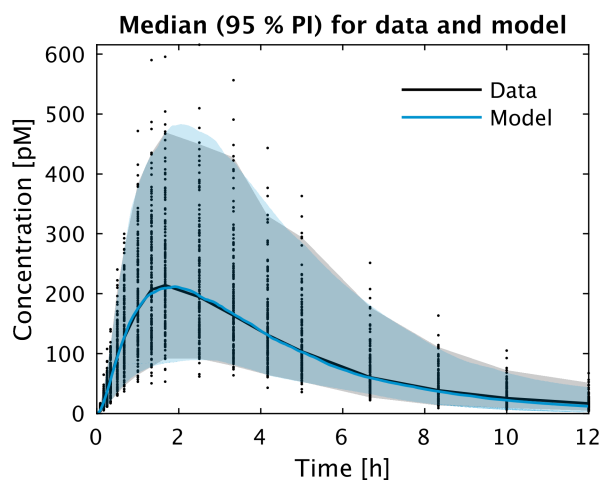
## 29. INTEGRATING EXTERNAL FILE SYSTEMS WITH PHARMTEX

PharmTeX supports the inclusion of files that are stored on a remote file system/server and that are identified with unique file ID. The files are denoted artifacts and are identified by PharmTeX by putting in front of the file ID (a number) the letters "RA", e.g. "RA12345678". For such files, PharmTeX will in the first compilation cycle create comma-separated list of this IDs in the file `getartifacts.txt` that is printed to the report folder. The files then need to be downloaded to the same folder as the report by inputting the proper custom commands at the point denoted `## Download artifacts at this point` in the `runlatex.pl` file as shown below. Moreover, the next five lines of code need to be uncommented (by removing a single `#` symbol in front of each line). The naming convention for artifact files is RA followed by the file ID number, possibly followed by an underscore `_` and then any file name, e.g. the one used on the remote server. The same name (but not ID) can be used for multiple files. An example used below is `RA12345678_vpc.png`.

```
## Download artifacts at this point
# if ( -e 'dotwice' ) {
#   system("$perltext -fmt=$fmtfile -jobname=\"$name\" -shell
# -escape -interaction=batchmode -draftmode \"$nametex\");
#   ## Download artifacts at this point
#   unlink 'dotwice';
# }
```

Once implemented, one can input a remote file system file using standard PharmTeX commands, e.g. for a figure:

**Figure 7. Remote file system artifact**



[Item 21](#), artifact ID RA12345678. Remote file system artifact.

An item number is still assigned to the figure, but with the exception that the artifact ID is also printed in the description of the figure and in the LIST OF ITEMS artifacts a denoted by their ID and not the path to the file. The time of update is taken from the file on the computer compiling the report, i.e. after the download of the from the remove file system.

**Note.** Downloaded artifacts are cleared when clearing auxiliary files using F4 and when finalizing the document using F12 in Texstudio.

### **30. COMMON WARNINGS AND ERRORS**

When working on a PharmTeX report, warnings are bound to be thrown by LaTeX at some point. In Texstudio, these appear in the "Log" pane in the box below the editor window (see [Figure 2](#)). Here are a few common ones:

**Table 18. Common warnings and errors**

Warning	Description	Solution
<code>Overfull \hbox (...pt too wide) in paragraph at lines ...</code>	This is a common error for tables with many columns that barely fit in the page. It means that the text is protruding outside one or more of the cells.	Switch table to a landscape page, use <code>\pmxtight</code> and/or <code>\pmxfont{\scriptsize}</code> just before the table.
<code>! I can't find file 'document.pipe'</code>	This is a Windows (10/11?) related error. It happens every once in a while, but goes away if you rerun with F1.	Rerun with F1.

Inline table. Common warnings and errors thrown by LaTeX.

## 31. REFERENCES

- 1 Albus M, Hubmann W, Scherer J, Dreikorn B, Hecht S, Sobizack N, and Mohr F, 2002. A prospective 2-year follow-up study of neurocognitive functioning in patients with first-episode schizophrenia. *Eur Arch Psychiatry Clin Neurosci* **vol. 252**(6): 262–267.  
URL <http://dx.doi.org/10.1007/s00406-002-0391-4>
- 2 Hughes C, Kumari V, Soni W, Das M, Binneman B, Drozd S, O’Neil S, Mathew V, and Sharma T, 2003. Longitudinal study of symptoms and cognitive function in chronic schizophrenia. *Schizophr Res* **vol. 59**(2): 137–146.
- 3 Javitt DC and Laruelle M, 2006. Textbook of Schizophrenia, chap. Neurochemical Theories, 85–116. American Psychiatric Publishing, Washington DC.
- 4 Braak H, Del Tredici K, Rüb U, de Vos RA, Steur ENJ, and Braak E, 2003. Staging of brain pathology related to sporadic parkinson’s disease. *Neurobiol Aging* **vol. 24**(2): 197–211.
- 5 Fidler M, Xiong Y, Schoemaker R, Wilkins J, Trame M, Post T, Leary R, Wang W, Wickham H, Eddelbuettel D, Ardia D, and Mullen K, 2018. nlmixr package for r. Website, accessed 18 SEP 2018.  
URL <https://CRAN.R-project.org/package=nlmixr>



## Appendix 1. INCLUDING AN EXTERNAL PDF

You can include an external PDF file using the code below. The bookmarks and hyperlinks of the external PDF will function correctly inside the parent document.

```
\pmxpdf  
{external.pdf} % filename  
{c} % page number location in the bottom of the page: l, r, c  
{pdf:pdffile} % cross reference key  
{External PDF file.} % description
```

The next 3 pages are an external document with [Item 22](#). External PDF file.

<b>Modeling Analysis Report</b>	
Report Title:	Population pharmacokinetic modeling of DRUG-X
Date Issued:	DD MMM YYYY
Sponsor:	COMPANY
Product:	DRUG-X
Analysis Provider:	ANOTHER COMPANY
Name Surname	Name Surname
Title	Title
Company	Company
City, Country	City, Country

This document contains confidential information belonging to COMPANY. Except as may be otherwise agreed to in writing, by accepting or reviewing these materials, you agree to hold such information in confidence and not to disclose it to others (except where required by applicable law), nor to use it for unauthorized purposes. In the event of actual or suspected breach of this obligation, COMPANY should be promptly notified.

## TABLE OF CONTENTS

1. INTRODUCTION .....	3
2. OBJECTIVE(S) .....	3
3. STUDY OVERVIEW .....	3
3.1. Study Design .....	3
3.2. Study Assessments .....	3
4. DATA FOR ANALYSIS .....	3
4.1. Analysis Data Files .....	3
4.2. Data Exclusions .....	3
4.3. Missing Data and Imputations .....	3
5. METHODS .....	3
5.1. Prior Knowledge and Modeling Experience .....	3
5.2. Modeling: Software and Strategy .....	3
5.3. Base Model Description .....	3
5.4. Random Effects Model Development .....	3
5.5. Inclusion of Covariates and Full Model Development .....	3
5.6. Final Model Development .....	3
5.7. Assessment of Model Adequacy (Goodness of Fit) .....	3
5.8. Assessment of Model Predictive Performance (Validation) .....	3
6. RESULTS .....	3
6.1. Observed Data .....	3
6.2. Deviations from the Population Modeling Analysis Plan (PMAP) .....	3
6.3. Base Model Results .....	3
6.4. Full Model Results .....	3
6.5. Final Model Results .....	3
6.6. Parameter Estimate Results .....	3
6.7. Graphical Summaries of Covariate Effects .....	3
6.8. Final Model Predictive Performance .....	3
7. DISCUSSION .....	3
8. CONCLUSION(S) .....	3

**1. INTRODUCTION**

**2. OBJECTIVE(S)**

**3. STUDY OVERVIEW**

**3.1. Study Design**

**3.2. Study Assessments**

**4. DATA FOR ANALYSIS**

**4.1. Analysis Data Files**

**4.2. Data Exclusions**

**4.3. Missing Data and Imputations**

**5. METHODS**

**5.1. Prior Knowledge and Modeling Experience**

**5.2. Modeling: Software and Strategy**

**5.3. Base Model Description**

**5.4. Random Effects Model Development**

**5.5. Inclusion of Covariates and Full Model Development**

**5.6. Final Model Development**

**5.7. Assessment of Model Adequacy (Goodness of Fit)**

**5.8. Assessment of Model Predictive Performance (Validation)**

**6. RESULTS**

**6.1. Observed Data**

**6.2. Deviations from the Population Modeling Analysis Plan (PMAP)**

**6.3. Base Model Results**

**6.4. Full Model Results**

**6.5. Final Model Results**

**6.6. Parameter Estimate Results**

**6.7. Graphical Summaries of Covariate Effects**

**6.8. Final Model Predictive Performance**

**7. DISCUSSION**

**8. CONCLUSION(S)**

## Appendix 2. INCLUDING CODE

```
\pmxlisting
{nonmem.mod} % input code file
{nonmem} % syntax highlighting
{lst:nonmem} % cross reference key
{Final model control stream} % caption
{\gls{NONMEM} control stream for final model.} % description
```

### Listing A2.1. Final model control stream

```
1 $PROBLEM run1.mod
2
3 $INPUT C ID TIME DV AMT EVID MDV WGT DOS WDOS SUB STUD
4
5 $DATA data.csv IGNORE=C
6
7 $SUBROUTINES ADVAN5 TRANS1
8
9 $MODEL
10 COMP = (SLOW, DEFDOSE)
11 COMP = (FAST)
12 COMP = (CENTRAL, DEFOBS)
13
14 $PK
15 ; Structural parameters
16 MU_1 = LOG(THETA(1))
17 KD = EXP(MU_1 + ETA(1))
18
19 MU_2 = LOG(THETA(2))
20 KA = EXP(MU_2 + ETA(2))
21
22 MU_3 = LOG(THETA(3))
23 KE = EXP(MU_3 + ETA(3))
24
25 MU_4 = LOG(THETA(4))
26 V = EXP(MU_4 + ETA(4))*WGT/75
27
28 ; Input to ADVAN5
29 K12 = KD
30 K23 = KA
31 K30 = KE
32
33 $ERROR
34 IPRED = F/V*1E3
35 Y = IPRED + IPRED*ERR(1) + ERR(2)
36 ERRP = SIGMA(1,1)
37 ERRC = SIGMA(2,2)
38
39 $THETA
40 (0,0.007) ; KD ; 1
41 (0,0.013) ; KA ; 2
42 (0,0.11) ; KE ; 3
43 (0,14) ; V ; 4
44
45 $OMEGA
46 0.1 ; KD ; 1
47 0.1 ; KA ; 2
48 0.1 ; KE ; 3
49 0.1 ; V ; 4
50
51 $SIGMA
52 0.01 ; ERR(1)
53 1 ; ERR(2)
54
55 $EST METHOD=1 INTERACTION NSIG=3 SIGL=10 NOABORT MAXEVAL=9999 PRINT=1 FILE=run6.ext
```

[Item 23](#). NONlinear Mixed Effects Modeling (NONMEM) control stream for final model.

## LIST OF ITEMS

---

ITEM	PATH	UPDATE (GMT)
<a href="#">Item 1</a>	PharmTeX.png	2018-02-19 12:54
<a href="#">Item 2</a>	references.bib	2018-09-27 14:22
<a href="#">Item 3</a>	glossary.tex	2021-06-29 16:04
<a href="#">Item 4</a>	folder.png	2023-10-04 11:53
<a href="#">Item 5</a>	texstudio.png	2023-09-27 10:20
<a href="#">Item 6</a>	figure.png	2016-07-28 15:10
<a href="#">Item 7</a>	residual1.png	2018-02-20 10:50
<a href="#">Item 8</a>	residual2.png	2018-02-20 10:50
<a href="#">Item 9</a>	residual3.png	2018-02-20 10:50
<a href="#">Item 10</a>	residual4.png	2018-02-20 10:51
<a href="#">Item 11</a>	table.csv	2018-02-27 08:57
<a href="#">Item 12</a>	table.csv	2018-02-27 08:57
<a href="#">Item 13</a>	table.csv	2018-02-27 08:57
<a href="#">Item 14</a>	table.csv	2018-02-27 08:57
<a href="#">Item 15</a>	table.csv	2018-02-27 08:57
<a href="#">Item 16</a>	parameters.csv	2017-12-22 13:34
<a href="#">Item 17</a>	parameters.csv	2017-12-22 13:34
<a href="#">Item 18</a>	table.pdf	2019-03-28 12:26
<a href="#">Item 19</a>	child.tex	2018-02-27 20:33
<a href="#">Item 20</a>	myplot.pdf	2018-02-27 21:06
<a href="#">Item 21</a>	RA12345678	2016-07-28 15:10
<a href="#">Item 22</a>	external.pdf	2018-03-07 10:53
<a href="#">Item 23</a>	nonmem.mod	2018-03-01 10:31

---