

Using the Foundations and Trends[®] L^AT_EX Class

Instructions for Creating an FnT Article

Suggested Citation: Alet Heezemans and Mike Casey (2018), "Using the Foundations and Trends[®] L^AT_EX Class", : Vol. xx, No. xx, pp 1–18. DOI: 10.1561/XXXXXXXXXX.

Alet Heezemans
now publishers, Inc.
alet.heezemans@nowpublishers.com

Mike Casey
now publishers, Inc.
mike.casey@nowpublishers.com

This article may be used only for the purpose of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval.

now
the essence of knowledge
Boston — Delft

Contents

1	The Distribution and Installation	2
1.1	Pre-requisites	2
1.2	The Distribution	2
1.3	Installation	3
2	Quick Start	5
2.1	<code>\documentclass</code>	5
2.2	<code>\issuesetup</code>	6
2.3	<code>\maintitleauthorlist</code>	6
2.4	<code>\author</code> and <code>\affil</code>	6
2.5	<code>\addbibresource</code>	6
3	Style Guidelines and \LaTeX Conventions	7
3.1	Abstract	7
3.2	Acknowledgements	7
3.3	References	7
3.4	Citations	8
3.5	Preface and Other Special Chapters	8
3.6	Long Chapter and Section Names	9
3.7	Internet Addresses	9

4	Compiling Your FnT Article	10
4.1	Compiling Your Article Prior to Submission	10
4.2	Preparing the Final Versions	11
4.3	Compiling The Final Versions	11
4.4	Wednesday	12
	Acknowledgements	14
	Appendices	15
A	Journal Codes	16
B	Files Produced During Compilation	18

Using the Foundations and Trends[®] L^AT_EX Class

Alet Heezemans¹ and Mike Casey²

¹*now publishers, Inc.; alet.heezemans@nowpublishers.com*

²*now publishers, Inc.; mike.casey@nowpublishers.com*

ABSTRACT

This document describes how to prepare a Foundations and Trends[®] article in L^AT_EX . The accompanying L^AT_EX source file FnTarticle.tex (that produces this output) is an example of such a file.

1

The Distribution and Installation

1.1 Pre-requisites

You will need a working LaTeX installation. We recomend using pdfflatex to process the files. You will also need biber.exe installed. This is distributed as part of the latest versions of LiveTex and MikTex. If you have problems, please let us know.

1.2 The Distribution

The distribution contains 2 folders: `nowfnt` and `nowfnttexmf`.

1.2.1 Folder `nowfnt`

This folder contains the following files using a flat stucture required to compile a FnT issue:

- `essence_logo.eps`
- `essence_logo.pdf`
- `now_logo.eps`
- `now_logo.pdf`

- nowfnt.cls
- nowfnt-biblatex.sty
- NOWFnT-data.tex

It also contains the following folders:

journaldata A set of data files containing the journal-specific data for each journal. There are three files per journal:

<jrnlcode>-editorialboard.tex

<jrnlcode>-journaldata.tex

<jrnlcode>-seriespage.tex

<jrnlcode> is the code given in Appendix A. You will need these three files to compile your article.

SampleArticle This folder contains this document as an example of an article typeset in our class file. The document is called `FnTarticle.tex`. It also contains this PDF file and the `.bib` file.

1.2.2 Folder `nowfnttexmf`

This folder contains all the files required in a `texmf` structure for easy installation.

1.3 Installation

If your \LaTeX installation uses a `localtexmf` folder, you can copy the `nowtexmf` folder to the `localtexmf` folder and make it known to your \TeX installation. You can now proceed to use the class file as normal.

If you prefer to use the flat files, you will need to copy all the required files each time into the folder in which you are compiling the article. Do not forget to copy the three data files for the specific journal from the folder `journaldata`.

You may need to configure your \TeX editor to be able to run the programs. If you have problems installing these files in your own system, please contact us. We use Computer Modern fonts for some of the

journals. You will need to make sure that these fonts are installed. Refer to your system documentation on how to do this.

2

Quick Start

The now-journal class file is designed in such a way that you should be able to use any commands you normally would. However, do **not** modify any class or style files included in our distribution. If you do so, we will reject your files.

The preamble contains a number of commands for use when making the final versions of your manuscript once it has been accepted and you have been instructed by our production team.

2.1 `\documentclass`

The options to this command enable you to choose the journal for which you producing content and to indicate the use of biber.

```
\documentclass[<jrnlcode>,biber]{nowjnt}.
```

<jrnlcode> is the pre-defined code identifying each journal. See Appendix A for the appropriate <jrnlcode>.

2.2 `\issuesetup`

These commands are only used in the final published version. Leave these as the default until our production team instructs you to change them.

2.3 `\maintitleauthorlist`

This is the authors list for the cover page. Use the name, affiliation and email address. Separate each line in the address by `\\`.

Separate authors by `\and`. Do not use verbatim or problematic symbols. `_` (underscore) in email address should be entered as `_`. Pay attention to long email addresses.

If your author list is too long to fit on a single page you can use double column. In this case, precede the `\maintitleauthorlist` command with the following:

```
\booltrue{authortwocolumn}
```

2.4 `\author` and `\affil`

These commands are used to typeset the authors and the affiliations on the abstract page of the article and in the bibliographic data.

`\author` uses an optional number to match the author with the affiliation. The author name is written `<surname>`, `<firstname>`.

`\affil` uses an optional number to match the author with the author name. The content is `<affiliation>; <email address>`.

2.5 `\addbibresource`

Use this to identify the name of the bib file to be used.

3

Style Guidelines and L^AT_EX Conventions

In this section, we outline guidelines for typesetting and using L^AT_EX that you should follow when preparing your document

3.1 Abstract

Ensure that the abstract is contained within the

```
\begin{abstract}
```

environment.

3.2 Acknowledgements

Ensure that the acknowledgements are contained within the

```
\begin{acknowledgements}
```

environment.

3.3 References

now publishers uses two main reference styles. One is numeric and the other is author/year. The style for this is pre-defined in the L^AT_EX

distribution and must not be altered. The style used for each journal is given in the table in Appendix A. Consult the sample-now.bib file for an example of different reference types.

The References section is generated by placing the following commands at the end of the file.

```
\backmatter
\printbibliography
```

3.4 Citations

Use standard `\cite`, `\citep` and `\citet` commands to generate citations.

Run biber on your file after compiling the article. This will automatically create the correct style and format for the References.

3.4.1 Example citations

This section cites some sample references for your convenience. These are in author/year format and the output is shown in the References at the end of this document.

Example output when using `citet`: **arvolumenumber** is a citation of reference 1 and **report** is a citation of reference 2.

Example output when using `citep`: (**beditorvolumenumber**) is a citation of reference 3 and (**inproceedings**) is a citation of reference 4.

3.5 Preface and Other Special Chapters

If you want to include a preface, it should be defined as follows:

```
\chapter*{Preface}
\markboth{\sffamily\slshape Preface}
{\sffamily\slshape Preface}
```

This ensures that the preface appears correctly in the running headings.

You can follow a similar procedure if you want to include additional unnumbered chapters (*e.g.*, a chapter on notation used in the paper), though all such chapters should precede Chapter 1.

Unnumbered chapters should not include numbered sections. If you want to break your preface into sections, use the starred versions of `section`, `subsection`, *etc.*

3.6 Long Chapter and Section Names

If you have a very long chapter or section name, it may not appear nicely in the table of contents, running heading, document body, or some subset of these. It is possible to have different text appear in all three places if needed using the following code:

```
\chapter[Table of Contents Name]{Body Text Name}  
\chaptermark{Running Heading Name}
```

Sections can be handled similarly using the `sectionmark` command instead of `chaptermark`.

For example, the full name should always appear in the table of contents, but may need a manual line break to look good. For the running heading, an abbreviated version of the title should be provided. The appearance of the long title in the body may look fine with L^AT_EX's default line breaking method or may need a manual line break somewhere, possibly in a different place from the contents listing.

Long titles for the article itself should be left as is, with no manual line breaks introduced. The article title is used automatically in a number of different places by the class file and manual line breaks will interfere with the output. If you have questions about how the title appears in the front matter, please contact us.

3.7 Internet Addresses

The class file includes the `url` package, so you should wrap email and web addresses with `\url{}`. This will also make these links clickable in the PDF.

4

Compiling Your FnT Article

During the first run using the class file, a number of new files will be created that are used to create the book and ebook versions during the final production stage. You can ignore these until preparing the final versions as described in Section 4.3. A complete list of the files produced are given in Appendix B.

4.1 Compiling Your Article Prior to Submission

To compile an article prior to submission proceed as follows:

Step 1: Compile the L^AT_EX file using pdf_latex.

Step 2: Run biber on your file.

Step 3: Compile again using pdf_LaT_EX. Repeat this step.

Step 4: Inspect the PDF for bad typesetting. The output PDF should be similar to FnTarticle.pdf. Work from the first page when making adjustments to resolve bad line breaks and bad page breaks. Re-run pdf_LaT_EX on the file to check the output after each change.

4.2 Preparing the Final Versions

If you choose the option to compile the final versions of your PDF for publication, you will receive a set of data from our production team upon final acceptance. With the exception of "lastpage", enter the data into the `\issuesetup` command in the preamble.

lastpage= This is the last page number in the sequential numbering of the journal volume. You will need to enter this once you have compiled the article once (see below).

4.3 Compiling The Final Versions

The final versions should be created once you received all the bibliographic data from our Production Team and you've entered it into the preamble. You will be creating a final online journal version pdf; a printed book version pdf; and an ebook version pdf.

Step 1: Compile the \LaTeX file using pdfLaTeX.

Step 2: Run biber on your file.

Step 3: Compile again using pdfLaTeX. Repeat this step.

Step 4: Inspect the PDF for bad typesetting. The output PDF should be similar to `FnTarticle.pdf`. Work from the first page when making adjustments to resolve bad line breaks and bad page breaks. Re-run pdfLaTeX on the file to check the output after each change.

Step 5: When you are happy with the output make a note of the last page number and enter this in `\issuesetup`.

Step 6: Compile the article again.

Step 7: Open the file `<YourFilename>-nowbook.tex`. This will generate the printed book version pdf.

Step 8: Compile the \LaTeX file using pdfflatex.

Step 9: Run biber on your file.

Step 10: Compile again using pdfLaTeX. Repeat this step.

Step 11: Repeat steps 7-10 on the file: <YourFilename>-nowebook.tex.
This will generate the ebook version pdf.

Step 12: Repeat steps 7-10 on the file: <YourFilename>-nowplain.tex.
This will generate a plain version pdf of your article. If you intend to post your article in an online repository, please use this version.

4.4 Wednesday

Proposition 4.1. G is hamiltonian implies that for any nonempty $S \subseteq V$, $G - S$ has at most $|S|$ components.

Theorem 4.1. G is Hamiltonian if for any vetices v, w such that $(v, w) \notin E$,

$$\deg(v) + \deg(w) \geq n$$

Knapsacle Problem

$$\begin{array}{ll} \max & \sum_{i=1}^n v_i u_i \\ \text{such that} & \sum_{i=1}^n w_i u_i \leq W \\ & u_i \in \{0, 1\}, i = 1, \dots, n \end{array}$$

This is the binary integer programming problem, which is NP-complete. We cannot find exact solution to this problem. There are 2^n possible solutions, which requires exponential time.

We can find some “pseudo”-polynomial algorithm. Assumption:

1. W is an integer.

State: remaining capacity, define as X_k .

Stage: item $1, \dots, n$.

$$J_N(x_N) = \begin{cases} v_N & \text{if } N \leq X_N \\ 0, & \text{if } w_N > X_N \end{cases}$$

$$J_N(x_{N-1}) = \begin{cases} 0 + J_N(X_N), & \text{if } w_{N-1} > X_{N-1}, \text{ Here } X_N = \\ \max\{v_{N-1} + J_N(x_N), 0 + J_N(x_N)\}, & \text{if } w_{N-1} \leq X_{N-1} \end{cases}$$

Here why the DP has “pesduo”-polynomial time, and the w_i should be integer?

Let's list states $\{x_1, \dots, x_N\}$. For each stage k , there would be $W+1$.

Each iteration at most 2 computation, and therefore we face $2(W+1) \cdot N$

4.4.1 Label Correcting Methods

Shortest Path. Back up some information.

The label refers to the intermediate information.

A^* -algorithm; Bellman-Ford Algorithm

Benchmark: MILP, CPLEX, CVX.

4.4.2 DP problems with perfect state information

Linear Quadratic System Let $x_k \in \mathbb{R}^1$ be the state; $u_k \in \mathbb{R}^n$ be the control; $\omega_k \in \mathbb{R}^n$ be the disturbance.

System Dynamics.

$$x_{k+1} = A_k x_k + B_k u_k + \omega_k$$

Stage cost:

$$x'_k Q_k x_k + u'_k R_k u_k$$

Here Q_k is required to be positive-semi-definite symmetric matrix; R_k to be positive-definite symmetric matrix.

$$J_{n-1}(x_{n-1}) = \min_{u_{n-1}} \mathbb{E}_\omega \{x'_{n-1} Q_{n-1} x_{n-1} + u'_{n-1} R_{n-1} u_{n-1} + J_n(x_n)\}$$

where

$$J_n(x_n) = (A_{n-1} x_{n-1} + B u_{n-1} + \omega)' Q_n (A_{n-1} x_{n-1} + B u_{n-1} + \omega)$$

Acknowledgements

The authors are grateful to Ulrike Fischer, who designed the style files, and Neal Parikh, who laid the groundwork for these style files.

Appendices

A

Journal Codes

The table below shows the journal codes to be used in `\documentclass`.
For Example: `\documentclass[ACC,biber]{nowfnt}`

Journal	<jrnocode>	Ref. Style
Annals of Corporate Governance	ACG	Author/Year
Annals of Science and Technology Policy	ASTP	Author/Year
FnT Accounting	ACC	Author/Year
FnT Comm. and Information Theory	CIT	Numeric
FnT Databases	DBS	Author/Year
FnT Econometrics	ECO	Author/Year
FnT Electronic Design Automation	EDA	Author/Year
FnT Electric Energy Systems	EES	Author/Year
FnT Entrepreneurship	ENT	Author/Year
FnT Finance	FIN	Author/Year
FnT Human-Computer Interaction	HCI	Author/Year
FnT Information Retrieval	INR	Author/Year
FnT Information Systems	ISY	Author/Year
FnT Machine Learning	MAL	Author/Year
FnT Management	MGT	Author/Year
FnT Marketing	MKT	Author/Year
FnT Networking	NET	Numeric
		<i>Continues</i>

Journal	<jrnocode>	Ref. Style
FnT Optimization	OPT	Numeric
FnT Programming Languages	PGL	Author/Year
FnT Robotics	ROB	Author/Year
FnT Privacy and Security	SEC	Author/Year
FnT Signal Processing	SIG	Numeric
FnT Systems and Control	SYS	Author/Year
FnT Theoretical Computer Science	TCS	Numeric
FnT Technology, Information and OM	TOM	Author/Year
FnT Web Science	WEB	Author/Year

B

Files Produced During Compilation

The files that are created during compilation are listed below. The additional *.tex files are used during the final production process only. See Section 4.3.

```
<YourFilename>-nowbook.tex  
<YourFilename>-nowchapter.tex  
<YourFilename>-nowebook.tex  
<YourFilename>-nowechapter.tex  
<YourFilename>-nowsample.tex  
<YourFilename>-nowplain.tex  
<YourFilename>.aux  
<YourFilename>.bbl  
<YourFilename>.bcf  
<YourFilename>.blg  
<YourFilename>.log  
<YourFilename>.out  
<YourFilename>.pdf  
<YourFilename>.run.xml  
<YourFilename>.synctex.gz  
<YourFilename>.tex  
<YourFilename>.toc
```