

Preparing Articles with the ECEASST \LaTeX Document Class

Benjamin Braatz¹, Author Two^{2*} and Author Three^{3†}

¹ bbraatz@cs.tu-berlin.de, <http://www.bbraatz.eu/>

Institut für Softwaretechnik und Theoretische Informatik
Technische Universität Berlin, Germany

² two@someuniversity.xyz

³ three@three.org

Institut de l'Éternité Divine
Some University, Some City, Some Country

Abstract: This document instructs authors in the usage of the ECEASST document class. It also gives an overview over orthographic, bibliographic and typographic conventions for ECEASST articles.

Moreover, the document itself is written using the class and is, therefore, an example document, which can be inspected by authors.

Keywords: ECEASST, \LaTeX , document class

1 Introduction

The ECEASST \LaTeX document class is used to prepare articles, which should be published in the Electronic Communications of the EASST. It is derived from the standard \LaTeX `article` class. In the following sections instructions for authors using this class will be given.¹ The source file of this document itself uses the class and may serve as an example.

In order to use this class the files `eceasst.cls` and `easst.pdf` or `easst.eps` have to be in the current directory (or somewhere \LaTeX can find them). An article using the class starts with `\documentclass{eceasst}`. It does not have any options. Note, that the following packages are required for the class and therefore do not have to be included in the document again: `mathptmx`, `helvet`, `courier`, `graphicx`, `color`, `amsmath`, `amsthm`², `amsfonts`, `amssymb`, `hyperref`, `hypcap`. These should be available in all contemporary \LaTeX distributions, such as \TeX Live and $\text{MiK}\text{\TeX}$.

We suggest to compile your publication using `pdflatex`, because it results in better PDF output, especially in connection with the `hyperref` package. It should, however, be also possible to use `latex` and `dvipdfm`. Attention should be paid to the warnings issued by \LaTeX

* This author is sponsored by the Foo Society under Grant Nr. 42-23.

† This author is partly sponsored by the “Weird \LaTeX ” project of the Bar Research Council.

¹ Standard features of \LaTeX and the `article` class, which are not documented here, such as footnotes, itemisations and enumerations, may be used if appropriate.

² The pre-defined theorem styles and environments are treated in [Subsection 3.2](#).

Table 1: Article frontmatter commands

<code>\title{Article Title}</code>	Required
<code>\short{Short Title}</code>	Optional, but Recommended
<code>\author{Author(s)}</code>	Required
<code>\institute{Institute(s)}</code>	Required
<code>\abstract{Abstract}</code>	Required
<code>\keywords{Keywords}</code>	Required

during compilation. While underfull hboxes and vboxes may be not that bad, overfull boxes should always be avoided.

The `hyperref` package not only enables different kinds of hyperlinks to online resources and internal links (references to sections, figures, citations, etc.), but also creates metadata in the PDF document properties and a table of contents in the Adobe Reader “Bookmarks” tab. Therefore, line breaks, mathematical formulae and other non-text characters and commands are not allowed in section headings and the `\title`, `\author` and `\keywords` frontmatter commands. To circumvent this the `\texorpdfstring` command of `hyperref` may be used. Its first argument is the disallowed L^AT_EX construction, while the second is the pure text equivalent for use in the PDF metadata. For example, a line break can be included by `\texorpdfstring{\ \ \ }{ }` or a mathematical formula by `\texorpdfstring{$a+b$}{a + b}`.

2 Frontmatter

The title, authors, institutes, abstract and keywords on the first page are created by the `\maketitle` command which should be the first command issued after `\begin{document}`. Volume editors creating a preface for their volume should use the `\prefacetitle` command instead, where the article frontmatter commands are completely ignored and just the title “Preface” is used on and the first page automatically.

Acknowledgements for sponsoring organisations can be inserted in the article frontmatter. Acknowledgements for reviewers and other people should not be included in the frontmatter, but rather be inserted in an `acknowledge` environment at the end of the last section just before the bibliography.

The commands that have to be provided by article authors themselves are summarised in [Table 1](#). The title of the article is provided by `\title` and `\short`, where the former is the title printed on the cover page and the top of the first content page and the latter is used for the header of even pages. If `\short` is omitted the long title is used, which will nearly always be too long for the header. Please capitalise the title except for prepositions, conjunctions, articles and auxiliaries. The abstract and keywords are given by the commands `\abstract` and `\keywords`, respectively.

The authors are given as parameter of the `\author` command and their affiliations in the `\institute` command. The standard L^AT_EX `\and` command is *not* provided, please decide for yourself which enumeration scheme (“one, two and three”, “one, two, and three”, “one, two,

three”) you want to use for multiple authors. In articles with multiple authors the `\autoref` and `\autolabel` commands can be used to establish links between the individual authors and their addresses, where the reference should follow the last name of the author immediately without a space and the label should be placed in front of the (e-mail) address of the author separated by a space. If acknowledgements for sponsoring organisations should be included a `\sponsor` command can be used right after the author’s name (or after her `\autoref`). This will typeset a footnote on the first page (but not on the cover).

The institute information should start with online addresses, where the commands `\email` and `\url` should be used for e-mail and home page addresses, respectively. The general information should consist of two lines, where the first is the department, institute or faculty (whatever you deem appropriate) and the second contains the university (or company), city and country. The names of institutes, universities and cities should be given in native language, where the \LaTeX `inputenc` package or commands like `\’` and `\”` can be used to include special characters. (For languages with totally different alphabets, e. g., asian, greek or kyrillian, however, please use a transcription to the latin alphabet.) Multiple institutes should be separated by a paragraph break (empty line or `\par`).

3 Article Structure

In this section, we describe the methods for structuring an article, namely sections and subsections in [Subsection 3.1](#), theorems and similar environments in [Subsection 3.2](#) and figures and tables in [Subsection 3.3](#).

3.1 Sectioning

For sectioning of an article the standard \LaTeX structuring commands, such as `\section` or `\subsection` may be used. As in the `article` class, `\chapter` is not available. The `\part` command should not be used in normal articles, but could be permissible in larger contributions, such as tutorial overviews or surveys.

References to sections should have capitalised names as in “Section 3” and “Subsection 3.1” (created by `Section~\ref{sect}` and `Subsection~\ref{subsect}`, respectively). A non-breaking space (`~`) should be put between the name and the number to avoid line breaks.

The `\autoref` command from the `hyperref` package can be used to generate complete references including the correct title before the number. Additionally, this command is useful, because in the resulting reference both the name and the number are hyperlinked and, therefore, constitute a larger target for readers. Thus, `\autoref{sect}` results in “[Section 3](#)” and `\autoref{subsect}` in “[Subsection 3.1](#)”.

When referring to a section or subsection without explicitly giving its number, “section” is not capitalised. The referring text should, however, provide a hyperlink as in “[this section](#)” (typeset by `\hyperref[sect]{this section}`) and “[this subsection](#)” (typeset by `\hyperref[subsect]{this subsection}`).

Additionally, you can use `\nameref` to refer using the name of the section as in “[Article Structure](#)” (typeset by `\nameref{sect}`). As usual in \LaTeX , it is also possible to use



`\pageref` to refer to the page, but this is rather superfluous in short articles.

In the following, examples for the substructures not used in this introduction are given below. Note, that it is not possible to refer to paragraphs and subparagraphs using `\ref` and `\autoref`, but only using `\nameref` and possibly `\pageref`.

3.1.1 Example Subsubsection

Example text in the Subsubsection [3.1.1](#) or better [Subsubsection 3.1.1](#).

Example Paragraph Example text in the Paragraph “[Example Paragraph](#)”.

Example Subparagraph Example text in the Subparagraph “[Example Subparagraph](#)”.

3.2 Theorems and Similar Environments

The following theorem-like environments are available. They are all numbered separately. All environments get the name as an optional argument. These names should be capitalised for the first six environments, while they should be left uncapitalised for examples and remarks, since these only provide additional information.

For the definition of these environments the `amsthm` package is used. See the documentation for this package for additional information on how you can create your own environments. The documentation should be available inside your \TeX distribution in the directory `doc/latex/amscls/amsthdoc.pdf`. Use one of the three theorem styles mentioned below for your own environments.

The first four environments are used for proper theorems, i. e., facts that are proven to be correct. They use the theorem style `ectheorem`, which is set in italics.

Theorem 1 (Some Theorem) *For theorems we have the `theorem` environment.*

Proof. Proofs can be inserted using the `proof` environment. □

Proposition 1 (Some Proposition) *The `proposition` environment may be used for propositions. The distinction between theorems and propositions is not precisely defined and left to the author.*

Proof Sketch. The `proof` environment gets an optional argument, which can be used to provide a custom title such as “Proof Sketch”. □

Lemma 1 (Some Lemma) *Lemmata can be created by the `lemma` environment. Lemmata are facts that are not results for themselves but are rather used in more important theorems and propositions.*

Proof Idea. “Proof Idea” is also a possible idea for a proof title. □

Table 2: References to theorem-like environments

Manual Code	Result	\autoref Code	Result
Theorem~\ref{the}	Theorem 1	\autoref{the}	Theorem 1
Proposition~\ref{prop}	Proposition 1	\autoref{prop}	Proposition 1
Lemma~\ref{lem}	Lemma 1	\autoref{lem}	Lemma 1
Corollary~\ref{cor}	Corollary 1	\autoref{cor}	Corollary 1
Definition~\ref{def}	Definition 1	\autoref{def}	Definition 1
Algorithm~\ref{alg}	Algorithm 1	\autoref{alg}	Algorithm 1
Remark~\ref{rem}	Remark 1	\autoref{rem}	Remark 1
Example~\ref{ex}	Example 1	\autoref{ex}	Example 1

Corollary 1 (Some Corollary) *Corollaries can be created by the `corollary` environment and are propositions following immediately from previous theorems, propositions and lemmata.*

The next two environments are deemed for definitions and algorithms. They use the theorem style `ecdefinition` using a roman body font.

Definition 1 (Some Definition) Definitions are created by the `definition` environment.

Algorithm 1 (Some Algorithm) The `algorithm` environment can be used for algorithms, which should be referencable in the same manner as the other theorem-like constructs.

The last two environments are used for auxiliary information. More precisely, remarks and examples fall into this category. The theorem style `ecremark` with a slanted body font is used for these environments.

Remark 1 (Remark concerning something) *Remarks are obtained by the `remark` environment.*

Example 1 (An example) *Examples are created by the `example` environment.*

All of these environments should be referenced using capitalised names. To achieve this the `\autoref` command may be used. Table 2 summarises references to the example environments above.

3.3 Figures and Tables

Figures and tables should be included using the appropriate floating environments and *not* inserted directly into the text. Both should be centered by `\centering` as the first command inside the environment. The `\label` for the float should be inserted after the `\caption`, where the caption should be set below figures and above tables. Please capitalise the first word of a caption, but not the rest (except for names).

When referencing specific figures and tables “Figure” and “Table” should be capitalised as in “Figure 1” (typeset by `Figure~\ref{fig}`). Please use a non-breaking space (~) between the word and the `\ref` command to avoid a line break. Do *not* abbreviate “Figure” to

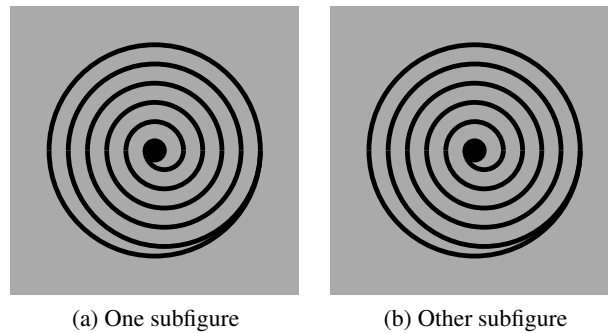


Figure 1: Example figure

“Fig.”. You may also use the `hyperref` command `\autoref` to create a link, where “Figure” or “Table” are inserted automatically and are also hyperlinked as in “[Figure 1](#)” (typeset by `\autoref{fig}`).

To save space and group related figures and tables you can use the `subfig` package with its `\subfloat` command as in our example figure. References to subfigures should also use the names “Figure” and “Table” as in “Figure 1a” (typeset by `Figure~\ref{subfig}`). The `\autoref` command can also be used as in “[Figure 1a](#)” (typeset by `\autoref{subfig}`). (Note that the `subfigure` package recommended in the previous version of the ECEASST class is deprecated.)

4 Bibliographies

In this final section we will give the bibliographical conventions for ECEASST.

The Bib_T_EX style `eceasst.bst` is distributed together with the ECEASST document class. It can be used to create bibliographies automatically. The scheme for citation references is derived from the standard `alpha` style, which uses the initials of authors rather than numbers.

Additionally to the fields available in the standard Bib_T_EX styles, this style allows a `doi` and a `url` field for Digital Object Identifiers³ and URLs, respectively. Please include this information if possible.

If you cite multiple articles from one journal issue, collection or proceedings or multiple volumes in a series, you can use the `crossref` field to save space.

The example bibliography at the end of this document is generated from the accompanying `example.bib` file, which is an example distributed together with Bib_T_EX, but enhanced by `doi` and `url` examples.

Acknowledgements: Here, we may acknowledge colleagues, referees, family, pets, ...

³ See <http://www.doi.org/> for more details

Bibliography

- [Aam86a] L. A. Aamport. The Gnats and Gnus Document Preparation System. *G-Animal's Journal*, 1986.
- [Aam86b] L. A. Aamport. The Gnats and Gnus Document Preparation System. *G-Animal's Journal* 41(7):73+, July 1986. This is a full ARTICLE entry.
[doi:10.9999/example-doi](https://doi.org/10.9999/example-doi)
<http://www.example.org/gnats-and-gnus/>
- [Aam86c] L. A. Aamport. The Gnats and Gnus Document Preparation System. In [GAJ86], pp. 73+. This is a cross-referencing ARTICLE entry.
- [GAJ86] *G-Animal's Journal* 41(7), July 1986. The entire issue is devoted to gnats and gnus (this entry is a cross-referenced ARTICLE (journal)).
- [Knu68] D. E. Knuth. *The Art of Computer Programming*. Four volumes. Addison-Wesley, 1968–90. Seven volumes planned (this is a cross-referenced set of BOOKs).
<http://www-cs-faculty.stanford.edu/~knuth/taocp.html>
- [Knu73a] D. E. Knuth. *Fundamental Algorithms*. Chapter 1.2. Addison-Wesley, 1973.
- [Knu73b] D. E. Knuth. *Fundamental Algorithms*. The Art of Computer Programming 1, section 1.2, pp. 10–119. Addison-Wesley, Reading, Massachusetts, second edition, 10 Jan. 1973. This is a full INBOOK entry.
- [Knu73c] D. E. Knuth. *Fundamental Algorithms*. Volume 1 in [Knu68], section 1.2, second edition, 1973. This is a cross-referencing INBOOK entry.
<http://www-cs-faculty.stanford.edu/~knuth/taocp.html>
- [Knu81a] D. E. Knuth. *Seminumerical Algorithms*. Addison-Wesley, 1981.
- [Knu81b] D. E. Knuth. *Seminumerical Algorithms*. The Art of Computer Programming 2. Addison-Wesley, Reading, Massachusetts, second edition, 10 Jan. 1981. This is a full BOOK entry.
- [Knu81c] D. E. Knuth. *Seminumerical Algorithms*. Volume 2 in [Knu68], second edition, 1981. This is a cross-referencing BOOK entry.
<http://www-cs-faculty.stanford.edu/~knuth/taocp.html>
- [Knv] The Programming of Computer Art.
- [Knv88] J. C. Knvth. The Programming of Computer Art. Vernier Art Center, Stanford, California, Feb. 1988. This is a full BOOKLET entry.
<http://www.somewhere.org/>
- [Lin77a] D. D. Lincoll. Semigroups of Recurrences. In *High Speed Computer and Algorithm Organization*. Academic Press, 1977.

- [Lin77b] D. D. Lincoll. Semigroups of Recurrences. In Lipcoll et al. (eds.), *High Speed Computer and Algorithm Organization*. Fast Computers 23, part 3, pp. 179–183. Academic Press, New York, third edition, Sept. 1977. This is a full INCOLLECTION entry.
[doi:10.9999/another-example-doi](https://doi.org/10.9999/another-example-doi)
- [Lin77c] D. D. Lincoll. Semigroups of Recurrences. Pp. 179–183 in [LLS77]. This is a cross-referencing INCOLLECTION entry.
- [LLS77] D. J. Lipcoll, D. H. Lawrie, A. H. Sameh (eds.). *High Speed Computer and Algorithm Organization*. Fast Computers 23. Academic Press, New York, third edition, Sept. 1977. This is a cross-referenced BOOK (collection) entry.
- [Man] The Definitive Computer Manual.
- [Man86] L. Manmaker. The Definitive Computer Manual. Chips-R-Us, Silicon Valley, silver edition, Apr.-May 1986. This is a full MANUAL entry.
<http://www.manmaker.com/manual/>
- [Mas88a] É. Masterly. Mastering Thesis Writing. Master’s thesis, Stanford University, 1988.
- [Mas88b] É. Masterly. Mastering Thesis Writing. Master’s project, Stanford University, English Department, June-Aug. 1988. This is a full MASTERSTHESIS entry.
<http://www.stanford.edu/masterstheses/masterly.pdf>
- [Mis] This is a minimal MISC entry.
- [Mis84] J.-B. Missilany. Handing out random pamphlets in airports. Handed out at O’Hare, Oct. 1984. This is a full MISC entry.
<http://www.missilany.org/pamphlet/ohare.pdf>
- [OUI83a] A. V. Oaho, J. D. Ullman, M. Yannakakis. On Notions of Information Transfer in VLSI Circuits. In *Proc. Fifteenth Annual ACM Symposium on the Theory of Computing*. 1983.
- [OUI83b] A. V. Oaho, J. D. Ullman, M. Yannakakis. On Notions of Information Transfer in VLSI Circuits. In Oz and Yannakakis (eds.), *Proc. Fifteenth Annual ACM Symposium on the Theory of Computing*. All ACM Conferences 17, pp. 133–139. Academic Press, Boston, Mar. 1983. This is a full INPROCEEDINGS entry.
[doi:10.9999/example.proceedings-article.STOC](https://doi.org/10.9999/example.proceedings-article.STOC)
<http://www.not-the-acm.org/stoc83/>
- [OUI83c] A. V. Oaho, J. D. Ullman, M. Yannakakis. On Notions of Information Transfer in VLSI Circuits. Pp. 133–139 in [OXstoc83a]. This is a cross-referencing INPROCEEDINGS entry.
- [OXstoc83a] *Proc. Fifteenth Annual Symposium on the Theory of Computing*. Boston, 1983. This is a cross-referenced PROCEEDINGS.

- [OXstoc83b] *Proc. Fifteenth Annual Symposium on the Theory of Computing*. 1983.
- [OY83] W. V. Oz, M. Yannakakis (eds.). *Proc. Fifteenth Annual Symposium on the Theory of Computing*. All ACM Conferences 17. Academic Press, Boston, Mar. 1983. This is a full PROCEEDINGS entry.
<http://www.not-the-acm.org/stoc83/>
- [PB88a] F. P. Phony-Baloney. *Fighting Fire with Fire: Festooning French Phrases*. PhD thesis, Fanstord University, 1988.
- [PB88b] F. P. Phony-Baloney. *Fighting Fire with Fire: Festooning French Phrases*. PhD dissertation, Fanstord University, Department of French, June-Aug. 1988. This is a full PHDTHESIS entry.
<http://www.fanstord.edu/phd/phony.pdf>
- [Ter88a] T. Terrific. An $O(n \log n / \log \log n)$ Sorting Algorithm. Technical report, Fanstord University, 1988.
- [Tér88b] T. Tèrrific. An $O(n \log n / \log \log n)$ Sorting Algorithm. Wishful research result 7, Fanstord University, Computer Science Department, Fanstord, California, Oct. 1988. This is a full TECHREPORT entry.
<http://www.fanstord.edu/techreports/1988/Number7.pdf>
- [ÜÑP̄] U. Ünderwood, N. Ñet, P. P̄ot. Lower Bounds for Wishful Research Results. Talk at Fanstord University (this is a minimal UNPUBLISHED entry).
- [ÜÑP̄88] U. Ünderwood, N. Ñet, P. P̄ot. Lower Bounds for Wishful Research Results. Nov., Dec. 1988. Talk at Fanstord University (this is a full UNPUBLISHED entry).
<http://www.underwood.net/talks/1988fanstord/>
- [Vol] Volume 2 is listed under Knuth [Knu81b].