acmsmall Author Submission Guide: Setting Up Your LaTeX 2ε Files

DONALD E. KNUTH, Stanford University
LESLIE LAMPORT, Microsoft Corporation

The LaTeX 2ε acmsmall document class formats articles in the style of the ACM small size journals and transactions. Users who have prepared their document with LaTeX 2ε can, with very little effort, produce camera-ready copy for these journals.

Categories and Subject Descriptors: D.2.7 [**Software Engineering**]: Distribution and Maintenance—*documentation*; H.4.0 [**Information Systems Applications**]: General; I.7.2 [**Text Processing**]: Document Preparation—*languages*; *photocomposition*

General Terms: Documentation, Languages

Additional Key Words and Phrases: Document preparation, publications, typesetting

ACM Reference Format:
Knuth, D. E. and Lamport, L. 2010. acmsmall Author submission guide: setting up your LaTeX 2ε files. ACM Comput. Surv. 2, 3, Article 1 (July 2010), 12 pages.
DOI = 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

1. INTRODUCTION

This article is a description of the LaTeX 2ε acmsmall document class for typesetting articles in the format of the ACM small size transactions and journals—*Transactions on Programming Languages and Systems*, *Journal of the ACM*, etc. It has, of course, been typeset using this document class, so it is a self-illustrating article. The reader is assumed to be familiar with LaTeX, as described by Lamport [1986].

This document also describes the acmsmall bibliography style.

LaTeX 2ε is a document preparation system implemented as a macro package in Donald Knuth's TeX typesetting system [Knuth 1984]. It is based upon the premise that the user should describe the logical structure of his document and not how the document is to be formatted. Formatting is under the direction of a *document class* chosen by the user. The user can dramatically change the way the document is formatted by simply choosing a different document class. The idea of separating the logical structure from the formatting comes from Brian Reid's *Scribe* system [Reid 1980].

It is impossible to provide predefined logical structures to handle all situations that may arise in a document, so users must sometimes make their own formatting decisions. LaTeX 2ε provides a number of features to assist in this task and, if necessary, the user can call upon the full power of TeX, which is probably the most powerful typesetting system currently available. However, very little user formatting is necessary for the majority of documents that appear in ACM journals and transactions. Consequently, it is quite easy to convert an existing LaTeX 2ε input file to the acmsmall style.

Author's addresses: D. E. Knuth, Computer Science Department, Gates Building 4B, Stanford University, Stanford, CA 94305-9045; L. Lamport, Microsoft Corporation, 1065 La Avenida, Mountain View, CA 94043.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000

2. THE TITLE PAGE

2.1. The Title, Author(s), and Abstract

Following order is mandatory to generate a correct title page:

```
\documentclass{acmsmall}
            %\acmVolume{V}
            %\acmNumber{N}
            %\acmArticle{A}
            %\acmYear{YYYY}
            %\acmMonth{O}
            \markboth{}{}
            \title{}
            \author{...
            \affil{...}}
            \begin{abstract}
            ...
            \end{abstract}
            \category{}{}{}
            \terms{}
            \keywords{}
            \acmformat{}
            \begin{document}
            \begin{bottomstuff}
            ...
            \end{bottomstuff}
            \maketitle
```

The `\documentclass[prodmode,journalName]{acmsmall}` takes as option "prodmode," which represents — Production Mode, i.e., employing similar fonts that will be used at the production stage. This option will give the user a fair idea of total pages that the article will produce at the typeset stage. Next option is the specific transaction/journal, one is preparing. The transactions/journals currently supported are as follows:

option name	journal
acmcie	Computers in Entertainment
acmcsur	Computing Surveys
acmjacm	Journal of the ACM
acmjea	Journal of Experimental Algorithmics
acmjetc	Journal on Emerging Technologies in Computing Systems
acmtaas	Transactions on Autonomous and Adaptive Systems
acmtaco	Transactions on Architecture and Code Optimization
acmtalg	Transactions on Algorithms
acmtalip	Transactions on Asian Language Information Processing
acmtecs	Transactions on Embedded Computing Systems
acmtissec	Transactions on Information and System Security
acmtkdd	Transactions on Knowledge Discovery from Data
acmtochi	Transactions on Computer-Human Interaction
acmtocl	Transactions on Computational Logic
acmtocs	Transactions on Computer Systems
acmtodaes	Transactions on Design Automation of Electronic Systems

option name	journal
acmtods	Transactions on Database Systems
acmtois	Transactions on Information Systems
acmtoit	Transactions on Internet Technology
acmtomacs	Transactions on Modeling and Computer Simulation
acmtoms	Transactions on Mathematical Software
acmtoplas	Transactions on Programming Languages and Systems
acmtos	Transactions on Storage
acmtosem	Transactions on Software Engineering and Methodology
acmtosn	Transactions on Sensor Networks
acmtrets	Transactions on Reconfigurable Technology and Systems
acmtslp	Transactions on Speech and Language Processing
acmtweb	Transactions on the Web
acmtmis	Transactions on Management Information Systems
acmtiis	Transactions on Interactive Intelligent Systems
acmtist	Transactions on Intelligent Systems and Technology
acmtoct	Transactions on Computation Theory
acmjdiq	Journal of Data and Information quality
acmtaccess	Transactions on Accessible Computing
acmtoce	Transactions on Computing Education

For example, to prepare a manuscript for the Transactions on Computational Logic the file should begin with

```
\documentclass[prodmode,acmtocl]{acmsmall}
```

The five commands

```
%\acmVolume{V}
%\acmNumber{N}
%\acmArticle{A}
%\acmYear{YYYY}
%\acmMonth{O}
```

are needed to generate footer and copyright information. The commands store the following information: volume number, issue number, article number, year of publication, and month number respectively. The appropriate values will be communicated by the Editor-in-Chief upon acceptance of the final version of the paper.

2.1.1. Title and Author. The LaTeX `\title` and `\author` declarations and the `\maketitle` command are employed as usual. However, the user must format the author a little differently to match the ACM standard. The following example [Archer, Jr. et al. 1984] illustrates most features:

```
\author{JAMES E. ARCHER, JR. \affil{Rational Machines}
        RICHARD CONWAY and FRED B. SCHNEIDER
             \affil{Cornell University}}
```

Note that authors' names are in uppercase letters, their affiliations are coded inside `\affil{...}` command, and sucessive authors with the same affiliation are separated by "and" (or commas and "and" if there are more than two).

In both the title and the author, you may have to insert `\\` commands if lines need to be broken.

2.1.2. Abstract. The abstract is typed as usual with the `abstract` environment. However, this environment must come before the `\maketitle` command.

2.2. Content Indicators and Keywords

The content indicators and keywords are entered with LaTeX declarations. The CR categories are indicated with `\category` declarations. The first CR category of this article, appearing right below the abstract, was entered with the following command:

```
\category{D.2.7}{Software Engineering}{Distribution and
  Maintenance}[Documentation]
```

Note that the last argument (which contains the subject descriptors) is optional, since some categories have none. Multiple subject descriptors are separated by `\and` commands, as in the last category of this article:

```
\category{I.7.2}{Text Processing}{Document Preparation}
         [Languages \and Photocomposition]
```

Use a separate `\category` declaration for each CR category; they will be listed in the order that the commands appear. The `\category` commands must precede the `\maketitle` command.

The General Terms are declared with a (single) `\terms` command as in the one for this article:

```
\terms{Documentation, Languages}
```

The `\terms` declaration must come before the `\maketitle` command. The terms *must* be chosen from the following list:

> Algorithms; Design; Documentation; Economics; Experimentation; Human factors; Languages; Legal aspects; Management; Measurement; Performance; Reliability; Security; Standardization; Theory; Verification;

The general terms are orthogonal to the Categories, at least theoretically, and so may be applied to any elements of the classification tree.

Think of them as 'perspectives' from which any topic may be approached. Thus you could use *Theory* or *Performance* for an article about *C.2.1 Distributed Networks*. However, some of these general terms actually slide over into content areas. Thus *Legal aspects* is a general term applicable to any category, but also an entire node in the tree, *K.5*, devoted to *Legal aspects of computing,* with many sub-topics.

So, though perhaps not perfect, the General Terms are most useful in online searches when used in combination with categories.

The "Additional Keywords and Phrases" item on the title page is provided by the `\keywords` declaration, **listed alphabetically**. For this article, they were produced by the following command:

```
\keywords{Document preparation, publications, typesetting}
```

There is no prescribed list of "additional keywords;" use any that you want.

The "ACM Reference Format" depicts the reference format of the article being processed. The reference is coded using `\acmformat` command. Specific abbreviations of Journal/Transactions names, their volume number, issue number, article number, publishing date and total number of pages are auto-generated (the DOI information will be available only at the publishing stage). Rest of the details need to be supplied by the user.

Since the total number of pages are auto calculated, the correct page numbers will not be shown when the file runs through LaTeX for the first time, or if the number of the first or last page has changed since the last update.

2.3. The Bottom of the Title Page

The bottom of the article's title page contains acknowledgment of support, the author(s) address(es), a "permission to copy" statement, and a line containing a copyright symbol (©) along with the copyright information.. This is all entered with a `bottomstuff` environment; there must be no blank line after the `\begin{bottomstuff}` command.

2.4. The Page Headers

`\markboth{}{}` generates the left and right-page headers. The first argument is the author's name(s):

— If there is one author, then use author's full name (ex. Leslie Lamport);
— If there are two authors, then abbreviate each author's first name (L. Lamport and D. E. Knuth);
— If there are more than two authors, then the format is Leslie Lamport et al.

The second argument of `markboth` is the title; if the title is too long, contract it by omitting subtitles and phrases, not by abbreviating words.

3. ORDINARY TEXT

Most of the body of the text is typed just as in an ordinary document. This section lists the differences.

3.1. Lists

3.1.1. Enumeration and Itemization. Let's begin with enumeration.

(1) The ACM style has two different formats for itemized lists, which I will call the *long* and *short* formats. The long format is generally used when the individual items are more than two or three lines long, but ACM has been inconsistent in their choice of format, sometimes using the long format for lists whose items are all one or two lines long and the short format for lists of long items. This list is an example of the long format.
(2) The ordinary `enumerate` environment produces the short format. For the long format, use the `longenum` environment.
(a) This inner enumeration uses the short format.
(b) It was produced using LaTeX's ordinary `enumerate` environment.
(c) ACM has no standard for enumerations nested more than two levels deep, so the `acmsmall` style does not handle them well.

Itemized lists are similar to enumerated ones.

— As with enumerations, there is a long and a short format for itemized lists. This list is in the long format.
— The long format is produced by the `longitem` environment. The ordinary `itemize` environment uses the short format.
— This is an itemized list using the short format.
— It was produced with the `itemize` environment that is used in ordinary LaTeX input.

It is interesting to observe that the style of tick mark used for an itemization changed around 1985 from an en dash (–) to an em dash (—).

3.1.2. Descriptions. A list is a sequence of displayed text elements, called items, each composed of the following two elements:

label: A marker that identifies or sets off the item. It is a number in an enumerated list and a tick mark in an itemized list.

item body: The text of the item. It is usually ordinary prose, but sometimes consists of an equation, a program statement, etc.

Or another paragraph, which will be indented like normal paragraphs.

When the labels of a list are names rather than numbers or tick marks, the list is called a *description* list. The ACM style has both long and short description lists. The above list is a short description list; the bodies of all the items are indented enough to accommodate the widest label. The following list is a long description list. The `acmsmall` style provides both kinds of description lists:

Short. The `describe` environment takes an argument, which should be the same as the argument of the `\item` command that produces the widest label. Thus, the above description list was begun with the command

```
\begin{describe}{{\em item body\/}:}
```

A description label is often emphasized in some way; in this example I used the LaTeX `\em` command, italicized the label. The ACM appears to have no standard convention for formatting the labels of a description list, so the `describe` environment leaves the label formatting up to you. An `\hfill` command can be used to produce a label like "*gnu* –" where *gnu* is flush left against the margin and the "–" is aligned flush right next to the item body.

Long. The standard LaTeX `description` environment produces a long description list. It italicizes the labels, and puts a period after them, which seems to be what is done in the ACM transactions/journals.

3.2. Theorems, Etc.

LaTeX provides a single class of theorem-like environments, which are defined with the `\newtheorem` command. The ACM style divides this class into two subclasses that are formatted differently. The first class includes theorems, corollaries, lemmas, and propositions. It is produced with the `\newtheorem` command. Such a theorem-like environment is often followed by a proof, for which the `acmsmall` style provides a `proof` environment.

THEOREM 3.2.1. *Notice that theorems are numbered inside the nearest section subsection.*

When listing within the theorem environment, this style will now produce roman parantheses.

PROOF. This theorem is an instance of `subtheorem`, theorems nested in subsections. □

Please use this type of set of definitions (don't use the ones depicted here as they have already been incorporated in `acmsmall` style), if you want to have more such environments:

```
\newtheorem{theorem}{Theorem}[section]
\newtheorem{conjecture}[theorem]{Conjecture}
\newtheorem{corollary}[theorem]{Corollary}
\newtheorem{proposition}[theorem]{Proposition}
\newtheorem{lemma}[theorem]{Lemma}
```

```
\newdef{definition}[theorem]{Definition}
\newdef{remark}[theorem]{Remark}
```

The second subclass of theorem-like environments includes ones for definitions, examples, and remarks. These environments are defined with the `\newdef` command, (used just above) which works the same as `\newtheorem`. Here is an example of such an environment.

Definition 3.2.2. This is an example of a Definition, typed with an `subexample` environment defined with `\newdef`. As you can see theorems are italicized and definitions are not.

Sometimes theorem-like environments are numbered in unusual ways, or are identified by a name. Consider the following example from Nielson [1985].

PROPERTY 3.2.3 (Ca). *Let syn* $\in$ *Syn, occ* $\in$ *Occ be maximal and sta* $\in$ *Sta. Then Tcol*[[*syn*]] *occ sta*$\downarrow 1 = Tsto$[[*syn*]] *sta.*

PROOF OF PROPERTY Ca. By straightforward structural induction, and is omitted. □

It was obtained by giving optional arguments to the `property` environment (defined with `\newtheorem`) and the `proof` environment and was typed as follows.

```
\begin{subproperty}[{\rm Ca}] Let ...  \end{subproperty}
\begin{proof}[of Property {\rm Ca}]  By straightforward ...
```

Notice that the optional argument to the `property` environment suppresses the automatic numbering. If a null optional argument were given to this environment by typing "`[]`", then it would have produced the label "PROPERTY." This is how unnumbered theorems, etc. are produced.

3.3. Overfull hbox - Stretching/filling one horizontal line

To solve a line break due to "Overfull `\hbox`", here is a plain TEX solution; here `\hsize` is the default setting of acmsmall.cls:

```
\hbox to \hsize{line sentence to be stretched}
```

This can be used in a list environment as well but `\hsize` declared to a reduce dimension:

```
\hbox{\vbox{\hsize = less than the default setting
\hbox to \hsize{line sentence to be stretched}}}
```

3.4. Algorithms

ACM recommends ruled style algorithms in `\small` point size. Users can use any of the available standard LATEX packages, to produce algorithm-like structures, e.g. Christophe Fiorio's `algorithm2e.sty`, Szasz Janos's `algorithmicx.sty` or Peter Williams' and Rogério Brito's `algorithmic.sty`, provided that the output is compatible with ACM style. You can also refer to `sample.tex`, where `algorithm2e.sty` is used to generate the ouput of an "Algorithm" environment.

3.5. Programs

Good formatting of programs requires a knowledge of their semantics, and is beyond the scope of a document production system. While "pretty printers" are useful for handling the many pages of a real program, the short examples that are published in

```
type date =
  record day: 1..31;
         month: 1..12;
         year: integer
  end
var mybirth, today : date;
var myage : integer;
```

Fig. 1. An example of a program centered in a figure

	$\bot$	F	T
$\bot$	$\bot$	$\bot$	T
F	$\bot$	F	T
T	$\bot$	T	T

Fig. 2. The truth table for the parallel-or.

articles should be formatted by hand to improve their clarity. The LaTeX `tabbing` environment makes the formatting of programs relatively easy, especially if the user defines commands for his particular language constructs. One may also use the `verbatim` environment.

The ACM style requires that programs be formatted with different size fonts, depending upon whether they appear in the text or in a figure, but that is handled by the figure macro which automatically sets the correct font size. Moreover, programs in running text should be indented two picas on each side (as provided by the `quote` environment), and programs in regular figures should be centered. (Programs in "narrow figures" (q.v.) are left or right justified automatically).

Here is an example of a program:

```
type date =
  record day: 1..31;
         month: 1..12;
         year: integer
  end
var mybirth, today : date;
var myage : integer;
```

Figure 1 shows how the same program looks in a figure.

In addition to formatting programs, the `tabbing` environment may be used for similar displayed material such as BNF syntax specifications and rewrite rules.

4. FIGURES AND TABLES

4.1. Figures

The ordinary LaTeX `figure` environment works as usual. Figure 2, which is Figure 6 of Nielson [1985], a bogus reference, was produced in this way. Note that figures should never appear in the text or at the bottom of a page. (If you use the figure placement optional argument, use only `t` or `p` or both; do not use `h` or `b`).

Some figures (and tables) have no caption except for the figure number. For such figures (and tables), one uses a `\nocaption` command, which has no argument, instead of the `\caption` command.

In addition to this method of formatting figures, the ACM class also uses figures with side captions, as in Figure 3. Such a figure is produced with the `narrowfig` environment. This environment has a single mandatory argument, which is the width of

```
type date =
  record day: 1..31;
         month: 1..12;
         year: integer
  end
var mybirth, today : date;
var myage : integer;
```

Fig. 3. An example of a program displayed in a figure.

Table I. This is an Example of Table Caption

First head[a]	Second head	Third head	$V_M(r)$
Left	Word entries	0.2	10.55
Left	Word entries	0.15	33.12
Left	Word entries	10.58	45.10
Left	Word entries	43.9	12.34
Left	Word entries	0.15	60.50

Source: This is a table sourcenote. This is a table sourcenote. This is a table sourcenote.

Note: This is a table footnote.

[a]This is a table footnote. This is a table footnote. This is a table footnote.

the figure. Note that if the figure is generated by `tabbing` or `tabular`, one can safely overestimate the size. It works just like the ordinary `figure` environment, except it must contain only one `\caption` or `\nocaption` command, which must come after the figure itself.

The `narrowfig` environment should obviously not be used unless the figure is narrow enough to leave a reasonable amount of space beside it for the caption. The ACM seems to have no consistent policy for choosing which style of figure to employ.

4.2. Tables

The standard LaTeX `table` environment can be used to create a table, but the user should add formatting commands to match with the ACM style. `acmsmall` provides a command called `\tbl{}{}`, which should be used inside the `table` environment. The first argument of `\tbl` command is the caption and the second argument is the table body coded inside standard LaTeX `tabular` environment. This command automatically calculates the width of the table and fits the caption and table notes accordingly.

Table notes should be added inside `tabnote` environment; they can be further differentiated using `\Note{}` and `\tabnoteentry{}{}` commands. Please refer `sample.tex` file for the coding.

Users can also apply `acmtable` environment, which was introduced in the older version of the ACM style file. This environment has a compulsory argument that equals the width of the table—more precisely, it specifies the width of the rules above and below the table. There should be only one `\caption` or `\nocaption` command, which should appear after the text of the table. (Even though the table caption is printed above the table, the `\caption` command comes after the table in the input file.)

5. THE END OF THE DOCUMENT

5.1. Appendix

The appendix (if the article has one) should precede the acknowledgments (if any) and bibliography. If the appendix isn't broken into separate sections, then you should add the following commands after the `\appendix` command:

```
\section*{APPENDIX}
\setcounter{section}{1}
```

Setting the counter is necessary so that numbered subsections and theorems will have the names "A.*N*" in the text.

For an article with multiple appendices, one begins the appendix with an `\appendix` followed by `\section*{APPENDIX}`, and then starts each appendix with an ordinary `\section` command.

Information about electronic appendices is given in Section 7 and in the Appendix.

5.2. Acknowledgments

An optional acknowledgments section follows all the text of the article, including any appendices. It is produced with the `acks` environment. Sometimes, there is just a single acknowledgment. This may be given using the `ack` or `acknowledgment` environment.

5.3. Bibliography

The bibliography follows the acknowledgments, and is the last significant body of text in the article. It is produced by the usual LaTeX commands.

The user is encouraged to let LaTeX produce his bibliography with the `\bibliography` command, letting BibTeX handle the formatting of the entries. The `acmsmall` bibliography style file generates citations in this format [Nielson 1985]. Put

```
\bibliographystyle{acmsmall}
```

between the `\begin{document}` and the `\end{document}`.

When submitting the document source (.tex) file to external parties, it is strongly recommended that the BibTeX .bbl file be manually copied into the document (within the traditional LaTeX bibliography environment) so as not to depend on external files to generate the bibliography and to prevent the possibility of changes occurring therein.

The conventional `\cite` command will generate citations as usual in LaTeX. Note that the style file automatically omits repeating author names [Nielson 1985; Knuth 1981]. If you mention the work explicitly in your prose, you should use `\citeN` command. This command generates for example, Nielson [1985] discusses denotational program transformations. Or, you use `\citeyear` and say that Nielson [1985] discusses them. The command `\shortcite` is an alias for `\citeyear`. Either command may be used in cases where one refers to multiple works (of the same authors!). For example, `Nielson~\shortcite{7:3:359,test}` generates Nielson [1985; 1981].

More variations of `\cite` are discussed in comments in the `acmtrans.sty` file. Here are some **examples** on how to get

(1) Appel [1996] → using either `\citeN` or `\citeyear`
(2) [Kempe 1879] → `\cite{kempe79}`
(3) Appel [1995; 1996] → `\shortcite{ref1-key,ref2-key}`
(4) Filé [1981a; 1981b] → `Fil\'{e}~\shortcite{engelfriet/file:81sweep, engelfriet/file:81passes}` or simply `\shortcite{ref1-key, ref2-key}`
(5) [Appel and Shao 1992; Shao and Appel 1994] → `\cite{appel-zhong-lsc92, shao94:clo}`
(6) Chow and Harrison [1992; 1994] → Chow and Harrison [`\citeyearNP{CH-popl92}; \citeyearNP{CH-iccl94}`]
(7) [Chow and Harrison 1992; 1994; Cousot and Cousot 1984] → `[\citeNP{CH-popl92}; \citeyearNP{CH-iccl94}; \citeNP{CC-apct77}]`
(8) [Cytron et al. 1991] → `\cite{cytron-et-al-toplas91}`

(9) Briggs et al. [1994] → `\citeN{briggs-cooper-torczon-toplas94}` or
(10) Duri et al. [1993] → `Duri~et~al.~\citeyear{DBDS-sigsoft93}`
(11) [Chaitin 1982; Chaitin et al. 1981] → `\cite{chaitin-pldi82,chaitin-et-al-cl81}`
(12) [Alblas 1991; Deransart et al. 1988; Knuth 1868] → `\cite{alblas:91intro, deransart/jourdan/lorho:88ag,knuth:68semantics}`
(13) [Gary and Johnson 1979] → `\cite{garey-johnson-bk79}`
(14) [Brand and Zafiropulo 1983; Gouda et al. 1984;1987] → `[\citeNP{brand83}; \citeNP{gouda84};\citeyearNP{gouda87}]`

The list will be updated as we find unique cases.

5.4. Received Date

The article should end by the following lines:

```
\received{Month Year}{Month Year}{Month Year}
```

The three values required are, respectively, 'received, revised and accepted dates'. The actual dates will be supplied by the Editor-in-Chief.

6. RUNNING HEADS AND FEET

The running foot of all but the title page of the article is declared with the `\runningfoot` command. It contains the name of the journal, volume, number, article number and date. The foot for the title page also contains the same information but it is declared with the `\firstfoot` command.

The default page style for the `acmsmall` style is `headings`. Thus, a `\markboth` command is used to set the running heads. The left head contains the author's name (or authors' names) and the right head contains the title. For long titles, some contraction of the title is used.

7. ELECTRONIC APPENDICES

Because of severe constraints on how many pages it can print, some ACM journals accept some articles with *electronic appendices:* appendices in Postscript format that will not appear in the printed article but will be available separately. If your article is accepted with an electronic appendix, you should put an appendix header where the appendix normally belongs (before the "acknowledgments"). The body of the electronic appendix should be given after the references. The `appendixhead` command is given as:

`\appendixhead{`*URLend*`}`

where *URLend* will be determined by the Editor (it is usually the last name of the first author).

In case your paper will have an electronic appendix, the part of the paper that will appear in print should LaTeX correctly, i.e. in this part no LaTeX references (`\ref`) should be made to the electronic appendix.

The result of `\appendixhead` looks like this:[1]

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

ACKNOWLEDGMENTS

This is an example of Acknowledgments environment, which should be coded after electronic appendix header (if any).

[1]See the end of this document for the remainder of the explanation of electronic appendices

REFERENCES

ARCHER, JR., J. E., CONWAY, R., AND SCHNEIDER, F. B. 1984. User recovery and reversal in interactive systems. *ACM Trans. Program. Lang. Syst. 6,* 1, 1–19.

KNUTH, D. E. 1981. *Seminumerical Algorithms* 2nd Ed. The Art of Computer Programming Series, vol. 2. Addison-Wesley, Reading, MA.

KNUTH, D. E. 1984. *The TEXbook*. Addison-Wesley, Reading, MA.

LAMPORT, L. 1986. *LATEX: A Document Preparation System*. Addison-Wesley, Reading, MA.

NIELSON, F. 1985. Program transformations in a denotational setting. *ACM Trans. Program. Lang. Syst. 7,* 3, 359–379.

REID, B. K. 1980. A high-level approach to computer document formatting. In *Proceedings of the 7th Annual Symposium on Principles of Programming Languages*. ACM, New York, 24–31.

Received February 2010; revised October 2010; accepted December 2010

Online Appendix to:
acmsmall Author Submission Guide: Setting Up Your LaTeX 2ε Files

DONALD E. KNUTH, Stanford University
LESLIE LAMPORT, Microsoft Corporation

The contents of the electronic appendix is written after the references and the "received" environment. The electronic appendix is started by an \elecappendix command:

```
\elecappendix
```

A. SPLITTING OFF THE ELECTRONIC APPENDIX

If you have an electronic appendix, only the main body of the article, up through and including the description of how to obtain the electronic appendix, will be printed in the journal.

It will be necessary to split your dvi or Postscript file into two parts: one to be printed, the other to be available by FTP. Please split your Postscript into two separate postscript files using dvipages, pslpr or psselect and email them separately to the Editor.

Note that the pages of the appendix are numbered App-1, App-2, etc. so as not to interfere with the normal journal pagination.

B. SINGLE APPENDIX

When an article has a single electronic appendix, then after the \elecappendix command, type the following.

```
\setcounter{section}{1}
```

If the text starts immediately, add a \medskip to set off the text from the horizontal rule created by \elecappendix.

C. MULTIPLE APPENDICES

For an article with multiple electronic appendices, one begins the appendix with an \elecappendix command, then starts each appendix with an ordinary \section command. Lower levels of sectioning are produced by the ordinary sectioning commands.

DOI 10.1145/0000000.0000000 http://doi.acm.org/10.1145/0000000.0000000