Preparing Articles for the ACM Transactions with LaTeX

LESLIE LAMPORT
Digital Equipment Corporation
ANDREW W. APPEL
Princeton University
and
JOHN TANG BOYLAND
University of California, Berkeley

The LaTeX `acmtrans` document style formats articles in the style of the ACM transactions. Users who have prepared their document with LaTeX 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

1. INTRODUCTION

This article is a description of the LaTeX `acmtrans` document style for typesetting articles in the format of the ACM transactions—*Transactions on Programming Languages and Systems*, *Transactions on Database Systems*, etc. It has, of course, been typeset using this document style, 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 `acmtrans` bibliography style.

LaTeX 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 style* chosen by the user. The user can dramatically change the way the document is formatted by simply choosing a different document style. The idea of separating the logical structure from the formatting comes from Brian Reid's *Scribe*

Author's address: L. Lamport, System Research Center, Digital Equipment Corporation, 130 Lytton Ave., Palo Alto, CA 94301.
Start of a second footnote ...

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 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 journals such as the ACM transactions. Consequently, it is quite easy to convert an existing LaTeX input file to the `acmtrans` style.

2. THE TITLE PAGE

2.1 The Title, Author(s), and Abstract

Following order is mandatory to generate a correct title page:

```
\documentstyle{acmtrans}
\markboth{}{}
\title{}
\author{}
\begin{abstract} ...  \end{abstract}
\category{}{}{}
\terms{}
\keywords{}
\begin{document}
\begin{bottomstuff} ... \end{bottomstuff}
\maketitle
```

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.\\ Rational Machines
        \and RICHARD CONWAY and FRED B. SCHNEIDER \\
            Cornell University}
```

Note that authors' names are in uppercase letters, authors are separated from their affiliation by a \\ command, multiple authors with the same affiliation are separated by "and" (or commas and "and" if there are more than two), and authors with different affiliations are separated by an `\and` command. The following example [Korach et al. 1984] shows what to do if there are more than two affiliations:

```
\author{E. KORACH \\ IBM Israel \\
        D. ROTEM \\ University of Waterloo
        \and N. SANTORO \\ Carleton University}
```

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.

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 (©) and a mysterious number. This is all entered with a `bottomstuff` environment; there must be no blank line after the `\begin{bottomstuff}` command. The permission to copy statement is produced by the `\permission` 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 A. Appel);
—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 `acmtrans` 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 `acmtrans` 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.

3.2 Theorems, Etc.

LaTeX provides a single class of theorem-like environments, which are defined with the `\newtheorem` command. The ACM transactions 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 `acmtrans` 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. Thank you David Sands.

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

The second subclass of theorem-like environments includes ones for definitions, examples, and remarks. These environments are defined with the `\newdef` command, 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 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.

Environments defined by `\newdef` use the optional argument the same way as those defined by `\newtheorem`.

With this definition

```
\newtheorem{secthm}{Theorem}[section]
```

one can solve the counter problems and counter and label problems for theorems, lemmas and definitions etc.

With the above definition and with the following variations

```
\newdef{secdefn}[secthm]{Definition} or
\newdef{seclem}[secthm]{Lemma} or
\newdef{sectheo}{secthm]{Theorem} etc.
```

one can achieve

(1) correct counters in definitions, theorems etc. in `\section` environment. For example, to generate "Definition 4.1."; without the above two definitions, it produces "Definition 4.0.1." and

(2) correct counter and label associated with definitions, theorems etc. in any environment (section or subsection). For example,

```
\begin{secdefn}[\thesecthm\ (label)]
  text text text
\end{secdefn}
```

produces the correct definition counter + correct placement of the period as in "Definition 3.1 (label).".

```
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

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 acmtrans.sty:

```
\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 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 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 transactions 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 ems 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.

	$\perp$	F	T
$\perp$	$\perp$	$\perp$	T
F	$\perp$	F	T
T	$\perp$	T	T

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

```
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.

3.5 User-specified Formatting

If LaTeX does not provide a particular text structure, the user must define it himself and specify how it is to be formatted. This is most easily done by defining the new structure in terms of existing ones; the LaTeX `list` and `trivlist` environments are useful tools. However, it is occasionally necessary for the user to provide explicit formatting commands.

The best guide to how something should be formatted is what has been done in the ACM transactions. While horizontal spacing tends to depend strongly upon the particular text, there is a standard amount of vertical space used to set off text. The ordinary LaTeX `\medskip` command produces a vertical space of the appropriate size.

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 transactions 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 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.

Table I. The truth table for the parallel-or.

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

4.2 Tables

The ordinary LaTeX `table` environment can be used, but it requires the user to add formatting commands to match the ACM transactions style. This formatting is performed automatically if the `acmtable` environment is used instead, producing the result shown in Table I, which shows the same table displayed in Figure 2. This environment has a mandatory argument that equals the width of the table—more precisely, it specifies the width of the rules above and below the table. There must be only one `\caption` or `\nocaption` command, which must come 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 8 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. (Since I can never remember how many *e*'s there are in *acknowledgments*, it seemed like an abbreviation was in order for the environment name. One may also spell out the name as `acknowledgments`). 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.

In 1993 the ACM changed bibliography styles, from numerical citations [13] to author's name and year [Nielson 1985]. The user is encouraged to let LaTeX produce his bibliography with the `\bibliography` command, letting BibTeX handle the formatting of the entries. The `acmtrans.sty` bibliography style file now generates citations in this format. Put

`\bibliographystyle{acmtrans}`

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

The conventional `\cite` command will generate citations as usual in LaTeX. Note that the style file automatically omits repeating author names [Nielson 1985; Nielson 1986]. 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; 1986]].

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

An ACM transactions article ends with the dates that the article and its revised versions were received and the date it was accepted for publication. In the `acmtrans` document style, this information is produced with the `received` environment. Type the body of the environment just as it should appear in the printed version. (Note that only `Received` and the months should be capitalized, the entries should be separated by semicolons, and the whole "sentence" should *not* be terminated by a period.) You can get these dates from 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, and date. The foot for the title page contains this information plus the page numbers. It is declared with the `\firstfoot` command.

The `\pages` command prints the page numbers of the article, producing something like "123–132". It is implemented with the LaTeX `\pageref` command, so it will not produce the correct page numbers the first time the file is run through LaTeX, or if the number of the first or last page has changed since the last time.

The default page style for the `acmtrans` style is `myheadings`. 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.

At present, the ACM prefers to strip in their own running heads and feet, so it is unnecessary to worry about them when producing camera-ready copy.

7. INTERACTING WITH ACM'S PRODUCTION STAFF

After your article is accepted, format your document and send a Postscript file to the editor-in-chief. An ACM copy editor will mark up a paper copy and send it back to you for corrections. If there are just a few corrections, they may be sent to you by e-mail. Iterate this process until the copy editor is satisfied (it may take one or two iterations).

Then print your final camera-ready copy on a 1200-dpi phototypesetter (or a 600-dpi laser printer, if you must) and mail it to ACM.

If you are using a laser printer for your final camera-ready copy, you may enjoy using special paper designed for this purpose (it gives a sharper image on a whiter background). One such paper (manufactured by Hammermill Papers) is:

> Hammermill Laser Plus (Featuring Wax Holdout)
> $8\frac{1}{2}$x11–12M–S24/60
> 10450–5
> For Prepress Proofing and Camera-Ready Masters.

But the use of special paper is entirely optional; plain white laser-printer paper is fine.

7.1 ACM Copy Editors' Preferences

We wish to thank the authors who have mailed to us copies of their page proofs. From the page proofs (and explicit requests from ACM), here is a list of items most frequently marked up by the copy editors.

Center. Displayed items like figures, pieces of program codes, and equations. Equation numbers to appear right flushed. Remember to add a period if the displayed equation ends in a sentence.

acmcs	ACM Comput. Surv.	jlp	J. Logic Program.
acmlett	ACM Lett. Program. Lang. Syst.	jcss	J. Comput. Syst. Sci.
acta	Acta Inf.	jsmrp	J. Softw. Maint. Res. Pract.
al	Ada Lett.	jss	J. Syst. Softw.
acr	Adv. Comput. Res.	jlsc	J. Lisp Symb. Comput.
bit	Bit	lpls	Lett. Program. Lang. Syst.
cacm	Commun. ACM	mscs	Math. Struct. Comput. Sci.
cj	Comput. J.	mst	Math. Syst. Theor.
cn	Comput. J.	ngc	New Gen. Comput.
cl	Comput. Lang.	scp	Sci. Comput. Program.
ict	Inf. Contr.	sicomp	SIAM J. Comput.
ieebcs	IEE/BCS Softw. Eng. J.	spe	Softw. Pract. Exper.
ieees	IEEE Softw.	tocs	ACM Trans. Comput. Syst.
ieeese	IEEE Trans. Softw. Eng.	tods	ACM Trans. Database Syst.
ieeetc	IEEE Trans. Comput.	tog	ACM Trans. Graphics
ieeetpds	IEEE Trans. Parall. Distrib. Syst.	toms	ACM Trans. Math. Softw.
ieeetit	IEEE Trans. Inf. Theory	toois	ACM Trans. Office Inf. Syst.
ipl	Inf. Process. Lett.	toplas	ACM Trans. Program. Lang. Syst.
icp	Inf. Comput.	tcs	Theor. Comput. Sci.
ist	Inf. Softw. Tech.	tr	Tech. Rep.
ijsa	Int. J. Supercomput. Appl.	jf	J. Funct. Program.
ijpp	Int. J. Parallel Program.	jlc	J. Logic and Comput.

Fig. 4. Common journal abbreviations.

Fonts. Following items in regular Roman font:
—such words like et al., e.g., i.e., ad hoc, and bona fide, and
—the body of the definition environment.

Etc. No double period when a sentence ends with “etc.”

Figure. Spell out the word “figure” when using in a sentence, as in Figure 1. Use a period as in “Fig. 1.” and not a colon as in “Fig. 1:”

Contractions. Avoid the use of contractions; for example, use *do not* instead of *don’t.* (If you prefer to use contractions to avoid stuffiness, you’ll have to tell the copy editor explicitly after you get the marked-up proofs.)

Conference References. Please use *4th International Conf. on Document Formatting*, not *Fourth International Conference ...*; that is, do not spell out the number.

Journal References. Figure 4 gives common journal abbreviations; there is a duplicate list in the acmtrans.bst file for your convenience. Using the *abbreviation* feature of BibTeX for journal names (and months!) makes it easy to follow the rules.

The best method of reducing the number of copy editorial markups is by using a recent ACM journal as a guide to typesetting your article.

8. ELECTRONIC APPENDICES

Because of severe constraints on how many pages it can print, *ACM Transactions on Programming Languages and Systems* (TOPLAS) accepts 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

`\appendixhead`{*journal*}{*article-id*}

where *journal* is the name of the journal (e.g. `toplas` or `todf`), and *article-id* is a number given to you by the Editor. The result of `\appendixhead` looks like this:[1]

ONLINE-ONLY APPENDICES

A. SPLITTING OFF THE ELECTRONIC APPENDIX

B. SINGLE APPENDIX

C. MULTIPLE APPENDICES

Appendix A is available only online. You should be able to get the online-only appendix from the citation page for this article:

http://www.acm.org/toplas

Alternative instructions on how to obtain online-only appendices are given on the back inside cover of current issues of ACM TOPLAS or on the ACM TOPLAS web page:

http://www.acm.org/toplas

ACKNOWLEDGMENTS

We wish to thank Howard Trickey for providing the now-obsolete version of the `acmtrans` bibliography style, and for giving advice on creating the new one; Rebecca Davies for helping to adapt Glenn Paulley's "Chicago" bibliography style to fit the new ACM style; and Marilyn Salmansohn and George Criscione for providing information on the official ACM transactions style.

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 (Jan.), 1–19.

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

KORACH, E., ROTEM, D., AND SANTORO, N. 1984. Distributed algorithms for finding centers and medians in networks. *ACM Trans. Program. Lang. Syst. 6,* 3 (July), 380–401.

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

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

NIELSON, F. 1986. Program transformations in a denotational setting. *ACM Trans. Program. Lang. Syst. 7,* 3 (July), 359–379.

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

Received February 1986; November 1993; accepted January 1996

---

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

elecappendixhttp://www.acm.org/toplas1998ACM Transactions on Document Formatting, Vol. 8, No. 1, January 1999, Pages 111–123. © 1999 ACM 0164-0925/98/0100-0208A $5.00

The contents of the electronic appendix is written after the references and the “received” environment. The electronic appendix is started by an `\elecappendix` command that takes two required arguments:

`\elecappendix`{*year*}{*ref*}

where *year* is the year of appearance (or, if this not known, the year of acceptance), necessary for the copyright notice, and *ref* is a sentence provided by the Editor describing the publication reference information regarding your article.

D. 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` or `pslpr` and email them separately to the Editor.

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

E. 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`.

F. 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.