The **achicago** LaTeX package
Chicago Manual author-date citations

Matt Swift `<swift@alum.mit.edu>`

Version: 1.2 Date: 2001/08/31
Documentation revision: 2001/08/31

Abstract

Achicago provides a documentation style for LaTeX that aims for compliance with the *Chicago Manual of Style*. It uses author-date citations (per ch. 16), but bibliography entries contain unabbreviated information (per ch. 15). Requires accompanying BibTeX bibliography style, `achicago`.

Contents

Part I

Discussion

1 General

This package is a companion to the `achicago` BibTeX bibliography style. The set of citation commands offered by *achicago* is unfortunately shared only with an early implementation of a *Chicago Manual*-compliant documentation style, *achicago* and `achicago`. FIX: any others? In the future, I hope to make *achicago* compatible with the most common

have to commit to using this bibstyle-package combination when you write your sources. This needn't be true, and one day I am going to create a series of mappings from other common bibstyles that are conceptually similar, such as the `harvard` styles. The user commands are slightly different, but it should be the case that either set of user commands can be the front end for either bibstyle.

Here in this documentation you will read about the commands you will use in your LaTeX source file to make citations, and what the citations look like. Documentation of the `achicago` bibstyle itself is in the file `achicago-bst.dvi`. There you can read about what the References section, the actual book list, looks like. There are also some new fields recognized and other information you might want to know that relates to your BibTeX bibliography database file (`bib` file). You may also wish to look at the *titles* package (also in the `Frankenstein` bundle), which can be very helpful in typesetting various styles of titles properly, even when nested. The *achicago* package already requires the *titles* package, so those commands are always there if you want to use them.

Warning: *This documentation is sparse but should be accurrate. I will improve it in the future.*

2 Notes on the future of this package and bibstyle

§16.25 permits this combination.

In the future I hope to document more closely *Chicago Manual*'s principles wrt each entry type, the many choices given by *Chicago Manual*.

Untested and indeed hardly testable nature of the subject, with all its many permutations and difficult special bibliographic cases. Feedback is very welcome, especially with citations from *Chicago Manual*.

FUTURE One thing I realize now is that I've kept the same user interface I inherited, which I don't think corresponds with any other popular style. Besides refinement, this is the next major step in the future of this bibstyle-package.

3 Pros and cons of this style

FIX: terminology: reference style, citation style, bibstyle, bibliography style

When you are required to use a certain reference style, your judgment is not called for. In the cases where it is, here are some considerations.

citations are verbose Sometimes this information is useful to the reader, sometimes not. If in most citations it is not useful, consider a style with briefer citations.

citation style and quality of writing FIX cite btxdoc Oren Patashnik argues that “encourages the passive voice and vague writing.” I’m not sure. Even if it’s true, an author can surely resist this “encouragement.” To encourage is not to require. Do consider how well *you* write using different citation styles. If the document is already written, realize that if you change the citation style to one different than the author had in mind while writing, you may make reading it more awkward. Then again, maybe you will improve it.

ease of locating references The entry for (Benson et al. 1980) will appear any number of entries *after* the entry for (Benson et al. 1999) when the former was written by Benson, Zymursky, Wheeler, and Flynn and the latter by Benson (i.e., the same Benson), Floyd, Wheeler, and Flynn, since Zymursky compares greater than Floyd.

This is an issue to consider if you have a large number of entries with the same initial author (or editor) and often with different sets of 3 or more subsequent authors. This is, in general, very unlikely.

The entry for (Grüber 1990b) may not follow the entry for (Grüber 1990a) immediately. When (the same) Grüber has authored Any number of entries may intervene, but their principals will all be exactly Grüber. Entries will intervene only in the case when Grüber both authored and edited a work in the same year, and authored one a work with a later date.

4 Usage

Here are the various citing commands, and examples of the citations they produce.

To do: *What about the situation when you end a sentence with something ending with “et al.”—in this case you don’t want to reproduce the period.*

Distinct examples are distinguished this way.

`\cite` {⟨*key*⟩}	parenthesized list of up to 3 principals or one "et al." plus a year label	(Brown 1978) (Jarke, Turner, and Stohl 1985)
`\cite` [⟨*spec*⟩]{⟨*key1*⟩}		(Brown 1978, 17) Jarke, Turner, and Stohl 1985, §3.3)
`\cite` {⟨*key1,key2*⟩}		(Brown 1978; Jarke, Turner, and Stohl
`\cite` [⟨*spec*⟩]{⟨*key1,key2*⟩} FIX: how does this work?		(Brown 1978; Jarke, Turner, and Stohl
`\citeNP` ⟨*key*⟩	as `\cite` but without enclosing parentheses	Brown 1978 Jarke, Turner, and Stohl 1985
`\citeA` {⟨*key*⟩}	as `\cite` but without year label(s)	(Brown) (Jarke, Turner, and Stohl)
`\citeANP` {⟨*key*⟩}	as `\citeA` but without enclosing parentheses	Brown Jarke, Turner, and Stohl
`\citeyear` {⟨*key*⟩}	as `\cite` but without principal list	(1978 1985
`\citeyearNP` {⟨*key*⟩}	as `\citeyear` but without enclosing parentheses	1978 1985
`\citeN` {⟨*key*⟩}[1]	principal list and parenthesized year label (i.e., a noun phrase)	FIX: referring to author instead of paper? need good example

To do: *Should I use a warning in case more than one key is given to a command that shouldn't have them?*

4.1 Short citation labels

Previous versions of *achicago* offered a parallel set of citation commands with the prefix `short` that created citations with abbreviated labels (`\shortcite`, `\shortciteNP`, `\shortciteA`, `\shortciteANP`, `\shortciteN`). *Achicago* now offers only one kind of label, which is abbreviated according to principles in the *Chicago Manual* as much as possible. For backwards compatibility, the `short` citation commands still function, but they are identical to their non-`short`, and will produce a warning that this syntax is deprecated. Do not use the `short` commands in new documents.

5 Some technical notes

To do: *See* §*16.14 for issues to do with multiple citations.*

---

[1] This command should only be used for one key.

6 History

I wrote this package incorporating `chicago\{,a\}.bst` and filechicago.sty (v4, 92/8). By now I've almost totally rewritten it. It was a great mess because it had been based on `newapa.bst` which had been based on several other bibstyles, and each author had really just hacked up what was there in the previous bibstyle apparently without fully understanding it. Now, of course, it is a shining example of clarity and efficiency.

Part II

Implementation

7 Version control

`\fileinfo`
`\DoXUsepackagE`
`\HaveECitationS`
`\fileversion`
`\filedate`
`\docdate`
`\PPOptArg`

These definitions must be the first ones in the file.

```
\def\fileinfo{Chicago Manual author-date citations}
\def\DoXPackageS {achicago}
\def\initelyHavECitationS {}
\def\fileversion{v1.2}
\def\filedate{2001/08/31}
\def\docdate{2001/08/31}
\edef\PPOptArg {%
  \filedate\space \fileversion\space \fileinfo
}
```

If we're loading this file from a `\ProcessDTXFile` command (see the *compsci* package), then `\JusTLoaDInformatioN` will be defined; othewise we assume it is not (that's why the FunkY NamE).

If we're loading from `\ProcessDTXFile`, we want to load the packages listed in `\DoXPackageS` (needed to typeset the documentation for this file) and then bail out. Otherwise, we're using this file in a normal way as a package, so do nothing. `\DoXPackageS`, if there are any, are declared in the `dtx` file, and, if you're reading the typeset documentation of this package, would appear just above. (It's OK to call `\usepackage` with an empty argument or `\relax`, by the way.)

```
\makeatletter% A special comment to help create bst files.  Don't change!
\@ifundefined{JusTLoaDInformatioN} {%
  }{% ELSE (we know the compsci package is already loaded, too)
  \UndefineCS\JusTLoaDInformatioN
  \SaveDoXVarS
  \eExpand\csname DoXPackageS\endcsname\In {%use \csname in case it's undefined
    \usepackage{#1}%
  }%
  \RestoreDoXVarS
  \makeatother
  \endinput
}% A special comment to help create bst files.  Don't change!
```

Now we check for LaTeX2e and declare the LaTeX package.

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{achicago}[\PPOptArg]
```

```
\RequirePackage{blkcntrl,moredefs,slemph,titles,verbatim}
```

```
\newboolean{Annotate}
\newcommand\annotate {%
  \Annotatetrue
}
\newcommand\noannotate {%
  \Annotatefalse
}
```

`\citework` is supposed to be a general command for citing things declared with `\newwork` in the *abbrevs* package. It has one optional and one required argument so that it is parallel with the other citing commands, but I cannot see any use for it without the optional argument. Environments can exert complete control over how this macro looks by resetting the three parameters. The default will look good outside all environments, in running text.

The second argument is expected to be something defined with `\newwork`.

Needs modification to handle the empty optional arg. Watch interfering with things surrounding macros might have set. `\relax`'s are intentionally left out to let constructions like `\csname` . . . `\endcsname` [eh? FIX] work on the arguments.

`\PreCiteWork`
`\PostCiteWork`

```
\providesavebox\sc@box@a
\newcommand\PreCiteWork {%
  (\csname%
}
\newcommand\PostCiteWork {%
  \end{lrbox}\usebox{\sc@box@a})%
}
```

We don't want to be unbreakable here, but we want a high penalty. We absolutely do not want to break the number range, so we put it in an lrbox.

I think comma is better, even though it might seem fussy, because it is better parallel with the way `\cite` works with an optional page argument: the convention is that page numbers come after commas.

```
\newcommand\MidCiteWork {%
  \endcsname,\penalty9000\ \begin{lrbox}{\sc@box@a}%
}
\newcommand\citework [2] {%
  \PreCiteWork #2\MidCiteWork #1\PostCiteWork
}
```

```
% The {} fools abbrevs.dtx into not adding an extra space
% \newcommand\MidCiteWork {%
%   \endcsname{}\penalty9000\ \begin{lrbox}{\sc@box@a}%
% }
%
```

To do: *Make citework* with no parentheses, or other alternative.*

FIX: When the ? is placed there, there are two left parens, one right.

We want the remaining macros in this section to be available in their own piece.

To do: *is `\PreChunk` the only dependence on blkcntrl? Should make this not necessary if so.*

```
\newcommand\PreAnnotation {%
  \PreChunk
}
```

This will make the definition of the `thebibliography` environment in `classes.dtx` do the right thing. FIX: not defined in letter class?

```
\defcommand\@openbib@code {%
```

```
  \advance\leftmargin\bibindent
  \itemindent -\bibindent
  \listparindent \itemindent
  \parsep \z@
}
\let\newblock\relax
```

This doesn't work at the beginning, for some reason. The auxfiles are not set up right? URK: don't do this. Confuses users and also prevents anyone from using achicago.sty with another bibstyle, such as a modified achicago.bst. Is there a way I can provide a useful warning message for those who might have been using this before, without a `\bibliographystyle`

```
%\AtEndDocument {%
%  \bibliographystyle{achicago}%
%}
```

The `achicago` bibliography style will insert some macros that are not defined by LaTeX, and some that must have new meanings. They are: `\citeN`, `\SCcite`, `\SCduplicate`, `\begin{SCannotation}`, `\end{SCannotation}`.

Some of these commands should properly have `@` in their names, but `@`-commands cannot appear in the `bbl` file. As a compromise, the names have the prefix `SC`.

`\SCduplicate`
`\ac@mmmdash`

The argument will contain the 'label' that is a duplicate, in case it might ever be of use. But for now, we just want to replace duplicates with 3-em dashes.

To do: *provide option to spell out the duplicate when it is the first entry on a page (oneside) or verso page (twosided)*

A 3em-dash.

```
\newcommand\ac@mmmdash {%
  \rule[.6ex]{3em}{.03ex}%
}
\newcommand*\SCduplicate [1] {%
  \ac@mmmdash
}
```

`\PreAnnotation`
`\ac@begingobble`
`\ac@endgobble`
`SCannotation`

This sets up the `SCannotation` environment. When the boolean `\IfAnnotate` is false, we gobble everything between `\begin{SCannotation}` and `\end{SCannotation}`.

We require the *verbatim* package to do this. I used to put the text into an `lrbox` and just never use the box. This required balanced text inside (not a bad thing), but it also would process any `\cite`-like commands that appeared in the gobbled text, which could lead to perpetual warnings about unresolved references. There were in fact no unresolved references, but the warnings were annoying.

```
\newlet\ac@begingobble\comment
\newlet\ac@endgobble\endcomment
```

To do: *I shouldn't define annote in terms of quotation, we should copy a standard one here; What is the point of the `\relax`? I ended up removing them before the `\ac@begingobble` cases because I had to use the `\expandafter`.*

```
\newenvironment{SCannotation} {%
    \ifAnnotate
      \let\PreQuotation\PreAnnotation
      \relax\quotation
    \else
```

```
      \expandafter\ac@begingobble
    \fi
  }{%
  \ifAnnotate
    \relax\endquotation
  \else
    \expandafter\ac@endgobble
  \fi
}
```

`\SCcite`
`\ac@firstoftwo`

`\SCcite` is what `achicago` produces. Its args are ‘label’, and ‘year.label’. FIX: aak, plus tag.

```
\ReserveCS\SCcite
\newlet\UnexpandableProtect\@unexpandable@protect

\newcommand*\ac@firstoftwo [2] {#1}
\newcommand*\ac@secondoftwo [2] {#2}
\newcommand*\ac@onespacetwo [2] {#1\ #2}
\newcommand*\ac@onespacepretwo [2] {#1\ \PreCite #2}

\newcommand*\ac@cite@preonecommatwopost [2] {%
  \PreCite #1\if@tempswa , #2\fi\PostCite
}
\newcommand*\ac@cite@onecommatwo [2] {%
  #1\if@tempswa , #2\fi
}
\newcommand*\ac@cite@onecommatwopost [2] {%
  #1\if@tempswa , #2\fi \PostCite
}

```

`\PreCite`
`\PostCite`

```
\newlet\PreCite  (
\newlet\PostCite )
```

`\cite`
`\citeNP`
`\citeA`
`\citeN`
`\citeANP`

The way this works is: `\\@cite` is called once for each citing command, and `\SCcite` is called once for each key. The results are spaced by either semicolons (`\\ac@cite@sc`) or commas (`\\ac@cite@comma`), and these become argument `#1` for `\\@cite`.

```
\def\cite {%
  \let\@cite\ac@cite@preonecommatwopost
  \let\SCcite\ac@onespacetwo
  \ac@cite@sc
}
\newcommand*\citeNP {%
  \let\@cite\ac@cite@onecommatwo
  \let\SCcite\ac@onespacetwo
  \ac@cite@sc
}
\newcommand*\citeN {%
  \let\@cite\ac@cite@onecommatwopost
  \let\SCcite\ac@onespacepretwo
  \ac@cite@comma
}
\newcommand*\citeA {%
  \let\@cite\ac@cite@preonecommatwopost
  \let\SCcite\ac@firstoftwo
  \ac@cite@sc
}
\newcommand*\citeANP {%
  \let\@cite\ac@cite@onecommatwo
  \let\SCcite\ac@firstoftwo
  \ac@cite@sc
}
```

`\shortcite`
`\shortciteNP`
`\shortciteN`
`\shortciteA`
`\shortciteANP`

```
\newlet\shortcite\cite
\newlet\shortciteNP\citeNP
\newlet\shortciteN\citeN
\newlet\shortciteA\citeA
\newlet\shortciteANP\citeANP
```

`\citeyear`
`\citeyearNP`

```
\newcommand*\citeyear {%
  \let\@cite\ac@cite@preonecommatwopost
  \let\SCcite\ac@secondoftwo
  \ac@cite@comma
}
\newcommand*\citeyearNP {%
  \let\@cite\ac@cite@onecommatwo
  \let\SCcite\ac@secondoftwo
  \ac@cite@comma
}
```

`\ac@citesep`
`\ac@cite@sc`
`\ac@cite@comma`

```
\ReserveCS\ac@citesep
\newcommand\ac@cite@sc {%
  \let\ac@citesep ;%
  \ac@cite
}
\newcommand*\ac@cite@comma {%
  \let\ac@citesep ,%
  \ac@cite
}
```

`\ac@cite` This command executes `\\b@foo` for every `\foo` in the list of cited labels, and separates them by arg `#1`.

There has got to be a more elegant solution to this whole thing. FIX

```
\newcommand*\ac@cite {%
  \@ifnextchar [ {%
      \@tempswatrue
      \ac@@cite
    }{% ELSE
    \@tempswafalse
    \ac@@cite[]%
  }%
}
```

To do: *handle reserving names*

```
\providecommand\@writeaux {%
  \immediate\write\@auxout
}
\NewName*{ac@@cite} {[#1]#2} {% args: [optarg] label % optarg MANDATORY
  \if@filesw
    \@writeaux{\string\citation{#2}}%
  \fi
  \@cite{%
    \InitCS\sc@t@a
    \@for\ac@label:=#2\do {%
      \sc@t@a
%      \let\sc@t@a\ac@citesep
      \def\sc@t@a {\ac@citesep\ }% add space
      \@ifundefined{b@\ac@label} {%
          {\bfseries ?}%
          \@warning{Citation ‘\ac@label’ on page \thepage\space undefined}%
        }{% ELSE
        \@nameuse{b@\ac@label}%
      }%
    }%
  }{#1}%
}
```

`\bibindent` Indent second and subsequent lines of bibliographic entries.

```
\setlength\bibindent{1.5em}
```

`thebibliography` There is no `openbib` option. The definitions of `\newblock` and `\@biblabel` are kept local in case something else weird is going on.

```
\newcommand\ac@defbib [2] {%
  \renewenvironment*{thebibliography} [1] {%
      #1*{#2\@mkboth{#2}{#2}}%
      \list{}{%
                          \leftmargin\z@
                          \advance\leftmargin\labelsep
                          \advance\leftmargin\bibindent
                          \itemindent -\bibindent
                          \listparindent \itemindent
                          \parsep \z@}%
```

Chicago Manual does not acknowledge different spacings after different marks of punctuation, distinguish interword from intersentence space, or give rules about where to break a line near an ellipsis. So we are on our own in the bibliography. I have chosen to leave things as they are done in the standard bibliography styles, because I haven't yet given it my close consideration. That is, we leave all the punctuation the same except for the period, which we set to 1000, I forget now whether that's a lower or upper case letter. Extending the space after a period when appropriate seems to be the purpose of using `\newblock`, in this bibstyle.

```
      \sfcode‘\.=\@m
      \def\newblock {%
        \hskip .11em \@plus.33em \@minus.07em%
      }%
      \let\@biblabel\Gobble
      \sloppy
      \clubpenalty4000\widowpenalty4000%
    }{%
    \def\@noitemerr {%
      \@latex@warning{Empty ‘thebibliography’ environment}%
    }%
    \relax\endlist
  }%
}
\@ifclassloaded{article} {%
    \ac@defbib{\section}{\refname}%
  }{% ELSE
  \ac@defbib{\chapter}{\bibname}%
}
\InitCS\ac@defbib  % FIX -- where else can I do this?
```

References

University of Chicago Press. 1993. *The Chicago Manual of Style*. 14th ed. Chicago: University of Chicago Press.

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.