The achicago LaTeX package
Chicago Manual A-style (but author-date) citations

Matt Swift <swift@alum.mit.edu>

Version: 1 Date: 1995/11/27
Documentation revision: 1999/03/01

Abstract

A bibliography style based on *The Chicago Manual of Style*. The style is mostly style A, but borrows the author-date reference system from style B. Section 16.25 of the manual permits this combination. Requires accompanying BibTeX bibliography style, `achicago`.

Contents

Part I
Discussion

1 General

Warning: *This documentation is spotty because I want to release this in time to get on TEXLive 4. Sorry! Despite appearances in this file, this package really does work. My testbed has been a long document with about 350 citations and a 12-page bibliography.*

Warning: *Annotation was working once, but I haven't toyed with it in a long time.*

This package is designed for use with the `achicago` BibTeX bibliography style. For the moment, you pretty much 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.

2 Notes on the future of this package and bibstyle

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

Some of this dox is spotty, some redundant. Here are the basic inline citations that you can conjure with this package:

```
%   Citation format:
%             (author-last-name year)
```

```
%             (author-last-name and author-last-name year)
%             (author-last-name et al. year)
%             (author-last-name)
%             author-last-name
%             author-last-name (year)
%             (author-last-name and author-last-name)
%             (author-last-name et al.)
%             (year) or (year, year)
%             year or year, year
%
```

Here are the various citing commands, and what they look like. Presently the variable which controls the number of names before switching to "et al." in the abbreviated author list is 2, but this can be made 3 or more easily with a simple change to a constant in 'format.short.label.names'. This unfortunately cannot be controlled by TEX. I don't think spacing after the period in "et al." is handled correctly: FIX. Also, 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.

`\cite{key}`	full author list and year	(Brown 1978; Jarke, Turner, and Stohl 1985)
`\citeNP{key}`	full author list and year, no parentheses	Brown 1978; Jarke, Turner, and Stohl 1985
`\citeA{key}`	full author list	(Brown; Jarke, Turner, and Stohl)
`\citeANP{key}`	full author list, without parentheses	Brown; Jarke, Turner, and Stohl
`\citeN{key}`[1]	full author list and year with only the year in parentheses, i.e., in the form of a noun	Brown (1978) states that . . .
`\shortcite{key}`	abbreviated author list and year	(Brown 1978; Jarke et al. 1985)
`\shortciteNP{key}`	abbreviated author list and year, no parentheses	Brown 1978; Jarke et al. 1985
`\shortciteA{key}`	abbreviated author list	(Brown; Jarke et al.)
`\shortciteANP{key}`	abbreviated author list, no parentheses	Brown; Jarke et. al
`\shortciteN{key}`[1]	abbreviated author list and year, in form of a noun	Jarke et al. (1985) state that . . .
`\citeyear{key}`	year in parentheses	(1978, 1985)
`\citeyearNP{key}`	year no parentheses	1978, 1985

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

4 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.

This BibTeX style has support for abbreviated author lists and for year-only citations. This is done by having the citations actually look like:

```
\citeauthoryear{full-author-info}{abbrev-author-info}{year}
```

5 Credits

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

6 Version control

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

These definitions must be the first ones in the file.

```
\def\fileinfo{Chicago Manual A-style (but author-date) citations}
\def\DoXPackageS {achicago}
\def\initelyHavECitationS {}
\def\fileversion{v1}
\def\filedate{1995/11/27}
\def\docdate{1999/03/01}
\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{moredefs,titles,blkcntrl}

\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?

```
\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` The argument will contain the actual 'long.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.

FIX: OK, we're trying to do it so that we print the argument if it's the first `\SCduplicate` on the page, using the *afterpage* package. Problem is, we process sometimes one or two more `\bibitem`'s than make it onto the page. This is going to be next to impossible, even if you change the `\output` routine. So comment it out, including requiring *afterpage*.

Instead, we define `\topitem`, which can be inserted manually into the `bbl` file right before the `\bibitem` that occurs with an `\SCduplicate` at the top of a final page. The `\topitem` command causes the immediately following (and only that one) `\SCduplicate` to print its argument.

```
%\RequirePackage{afterpage}
%\newboolean{@sc@duplicatedash@}
%\@sc@duplicatedash@true
%\SaveCS\bibitem
%\newcommand\sc@bibitemfirst {%
%  \typeout{in bibitemfirst  [\the\foo]}%
%  \afterpage {%
%    \typeout{in afterpage [\the\foo]}%
%    \global\@sc@duplicatedash@false
%    \global\let\bibitem\sc@bibitemfirst
%  }%
%  \let\bibitem\MDSavedbibitem
%  \bibitem
%}
%\let\bibitem\sc@bibitemfirst

\newcommand*\sc@duplicate@dash [1] {%
%  \if@sc@duplicatedash@
    \sc@mmmdash
%  \else
%    #1%
%  \fi
```

```
%  \@sc@duplicatedash@true
}

\newcommand\sc@duplicate@arg [1] {%
  #1%
  \let\SCduplicate\sc@duplicate@dash
}
\newlet\SCduplicate\sc@duplicate@dash
\newcommand\topitem {%
  \let\SCduplicate\sc@duplicate@arg
}
```

\sc@mmmdash A 3em-dash.

```
\newcommand\sc@mmmdash {%
  \rule[.6ex]{3em}{.03ex}%
}
```

\PreAnnotation
\sc@begingobble
\sc@endgobble
SCannotation

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

To do: *These should be* comment *and* endcomment*, but we have to be careful how we define them. An lrbox required balanced text inside.*

```
\newcommand\sc@begingobble {%
  \relax\begin{lrbox}{\sc@box@a}%
}
\newcommand\sc@endgobble {%
  \relax\end{lrbox}\sbox\sc@box@a{}%
}
```

To do: *I shouldn't define annote in terms of quotation, we should copy a standard one here.*

```
\newenvironment{SCannotation} {%
    \ifAnnotate
      \let\PreQuotation\PreAnnotation
      \relax\begin{quotation}%
    \else
      \sc@begingobble
    \fi
  }{%
  \ifAnnotate
    \relax\end{quotation}%
  \else
    \sc@endgobble
  \fi
}
```

\SCcite
\sc@one@three
\sc@two@three

\SCcite is what achicago produces. Its args are 'long.label', 'short.label', and 'year.label'.

```
\ReserveCS\SCcite

\newcommand\sc@one@three [3] {#1}
\newcommand\sc@two@three [3] {#2}
```

This spits out arguments #1 and #3 with #2 intervening iff \@tempswatrue.

```
  FIX -- what's going on
% \newcommand\sc@citeformat [3]
%  {\def\@cite#1#2{%
%    #1\if@tempswa#2\fi#3}
%
```

\PreCite
\PostCite

Should have one single command be what each of the following executes. Its arguments determine what all the other stupid things do. hunh? FIX

```
\newcommand\PreCite {%
  (%
}
\newcommand\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 (\\sc@cite@sc) or commas (\\sc@cite@comma), and these become argument #1 for \\@cite.

\PreCite1[, 2]\PostCite \sc@citeformat{\PreCite##1}{, ##2}{\PostCite}

```
\def\cite {%
  \defcommand*\@cite [2] {%
    \PreCite ##1\if@tempswa , ##2\fi\PostCite
  }%
  \defcommand*\SCcite [3] {%
    ##1\ ##3%
  }%
  \sc@cite@sc
}
% 1[, 2]    \sc@citeformat{##1}{, ##2}{}
\newcommand*\citeNP {%
  \defcommand*\@cite [2] {%
    ##1\if@tempswa , ##2\fi
  }%
  \defcommand*\SCcite [3] {%
    ##1\ ##3%
  }%
  \sc@cite@sc
}
% 1[, 2])  \sc@citeformat{##1}{, ##2}{)}
% \newcommand*\citeN {%
%   \defcommand*\@cite [2] {%
%     ##1\if@tempswa , ##2\fi
%   }%
%   \defcommand*\SCcite [3] {%
%     ##1\ \PreCite##3\PostCite
%   }%
%   \sc@cite@comma
% }
```

'long.label' ('year', optarg)

```
\newcommand*\citeN {%
  \defcommand*\@cite [2] {%
```

```
    ##1\if@tempswa , ##2\fi\PostCite
  }%
  \defcommand*\SCcite [3] {%
    ##1\ \PreCite ##3%
  }%
  \sc@cite@comma
}
\newlet\UnexpandableProtect\@unexpandable@protect
% \PreCite1[, 2]\PostCite  \sc@citeformat{\PreCite ##1}{, ##2}{\PostCite}
\newcommand*\citeA {%
  \defcommand*\@cite [2] {%
    \PreCite ##1\if@tempswa , ##2\fi\PostCite
  }%
  \let\SCcite\sc@one@three
  \sc@cite@sc
}
% 1[, 2]    \sc@citeformat{##1}{, ##2}{}
\newcommand*\citeANP {%
  \defcommand*\@cite [2] {%
    ##1\if@tempswa , ##2\fi
  }%
  \let\SCcite\sc@one@three\sc@cite@sc
}
```

\shortcite
\shortciteNP
\shortciteN
\shortciteA
\shortciteANP

```
\newcommand*\shortcite {%
  \defcommand*\@cite [2] {%
    \PreCite ##1\if@tempswa , ##2\fi\PostCite}%
  \defcommand*\SCcite [3] {%
    ##2\ ##3%
  }%
  \sc@cite@sc
}
\newcommand*\shortciteNP {%
  \defcommand*\@cite [2] {%
    ##1\if@tempswa , ##2\fi
  }%
  \defcommand*\SCcite [3] {%
    ##2\ ##3%
  }%
  \sc@cite@sc
}
\newcommand*\shortciteN {%
  \defcommand*\@cite [2] {%
    ##1\if@tempswa , ##2\PostCite\else\PostCite\fi}%
  \defcommand*\SCcite [3] {%
    ##2\ \PreCite ##3%
  }%
  \sc@cite@comma
}
\newcommand*\shortciteA {%
  \defcommand*\@cite [2] {%
    \PreCite ##1\if@tempswa , ##2\fi\PostCite
  }%
```

```
  \let\SCcite\sc@two@three
  \sc@cite@sc
}
\newcommand*\shortciteANP {%
  \defcommand*\@cite [2] {%
    ##1\if@tempswa , ##2\fi
  }%
  \let\SCcite\sc@two@three\sc@cite@sc
}
```

\citeyear
\citeyearNP

```
% \PreCite1[, 2]\PostCite  \sc@citeformat{\PreCite ##1}{, ##2}{\PostCite}
\newcommand*\citeyear {%
  \defcommand*\@cite [2] {%
    \PreCite ##1\if@tempswa , ##2\fi\PostCite
  }%
  \defcommand*\SCcite [3] {%
    ##3%
  }%
  \sc@cite@comma
}
% 1[, 2]    \sc@citeformat{##1}{, ##2}{}
\newcommand*\citeyearNP {%
  \defcommand*\@cite [2] {%
    ##1\if@tempswa , ##2\fi
  }%
  \defcommand*\SCcite [3] {%
    ##3%
  }%
  \sc@cite@comma
}
```

\sc@citesep
\sc@cite@sc
\sc@cite@comma

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

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

```
}
```

To do: *handle reserving names*

```
\providecommand\@writeaux {%
  \immediate\write\@auxout
}
\NewName*{sc@@cite} {[#1]#2} {% args: [optarg] label % optarg MANDATORY
  \if@filesw
    \@writeaux{\string\citation{#2}}%
  \fi
  \@cite{%
    \InitCS\sc@t@a
    \@for\sc@label:=#2\do {%
      \sc@t@a
%      \let\sc@t@a\sc@citesep
      \def\sc@t@a {\sc@citesep\ }% add space
      \@ifundefined{b@\sc@label} {%
          {\bfseries ?}%
          \@warning{Citation `\sc@label' on page \thepage\space undefined}%
        }{% ELSE
        \@nameuse{b@\sc@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\sc@defbib [2] {%
  \renewenvironment*{thebibliography} [1] {%
      #1*{#2\@mkboth{#2}{#2}}% FIX didn't have this comment char here, OK?
      \relax\begin{list}{}{%
                          \leftmargin\z@
                          \advance\leftmargin\labelsep
                          \advance\leftmargin\bibindent
                          \itemindent -\bibindent
                          \listparindent \itemindent
                          \parsep \z@}%
```

The Chicago Manual (University of Chicago Press 1993) 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\end{list}%
  }%
}
\@ifclassloaded{article} {%
    \sc@defbib{\section}{\refname}%
  }{% ELSE
  \sc@defbib{\chapter}{\bibname}%
}
\InitCS\sc@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.