The amsrefs LaTeX package and the amsxport BibTeX style

Michael Downes
American Mathematical Society
mjd@ams.org

Introduction

When the bibliography entries in a LaTeX document are written in amsrefs form, they have rich internal structure and high-level markup close to what is traditionally found in BibTeX database files. Among other things, this raises the information quality of the LaTeX document if it is intended to serve as (or to yield via automatic translation) a self-contained archival version. Using amsrefs markup also means that the style of the bibliography can be specified completely in a LaTeX documentclass file, instead of being handled partly by LaTeX and partly by the BibTeX style file, in two different style languages, each of them more idiosyncratic than the other.

It has always been possible to write short bibliographies by hand without going through BibTeX, but those who do so usually put in as many ad hoc formatting commands as BibTeX would. An author who uses the amsrefs package when writing a bibliography by hand will be better off if it becomes necessary at some later time to change the style of the bibliography: the interior structural markup means that the same information can be reformatted in different ways by simple changes in the LaTeX setup rather than by explicitly changing formatting commands in each item.

Some of the sources that I consulted during the design of the amsrefs package are given in the bibliography. The bibliographies of [1] and [7] are recommended. The chief goal of my design efforts was to find a way of representing and citing bibliography entries at a level of markup abstract enough to support automatic formatting for a wide range of known bibliography styles without any change in the data; the secondary goal was to find a syntax for this representation that was natural in a LaTeX context and as easy as possible for authors to use. What I ended up with was a design that would leave any experienced TeXnician aghast at the implementation difficulties. Fortunately, however, many of the hardest bits could be dealt with using known methods; for example, Donald Arseneau's cite package established long ago that sorting and compressing lists of cite numbers is not horrendously impossible to do in TeX, and the technique used in amsrefs for combining multiple author names is adapted from some code of mine written a while back for the amsart documentclass.

Contents of a bibliography entry

Bibliography entries are done with a \bib command, not \bibitem, and look like this:

```
\bib{BW}{article}{
 author={Bertram, A.},
 author={Wentworth, R.},
  title={Gromov invariants for holomorphic
         maps on Riemann surfaces},
   date={1996},
journal={jams},
 volume={9},
 number={2},
  pages={529\ndash 571},
}
```

This will be recognizable as very similar to the data format used in BibTeX database files. There are, however, some key differences. If you use the amsxport BibTeX style and export from a BibTeX database, the differences will be automatically attended to, but if you write a short bibliography by hand, you should bear in mind the following points:

braces Always use braces to enclose the value of each field, never quotes (BibTeX allows both quotes and braces).

repeated fields Certain fields can be repeated (and should be, where applicable). As shown in this example, when there is more than one author, each author name is given separately. The task of combining the names as needed for the current publication is handled by LaTeX. The fields that are defined to be repeatable by the amsrefs package are author, editor, isbn, review, and translator.

inverted names It is recommended to give author and editor names uniformly in *Last, First* order. This is the form that provides the most flexibility with the least extra markup. When printed, the names will be automatically uninverted in whatever manner is specified by the bibliography style in use. (Some styles have the first author's

name inverted and the remaining names not inverted.) For suffixes such as III or Jr., write `Smith, John Q., III` or `Smith, John Q., Jr.` as the inverted form. For Chinese names and others where the surname comes first, if you write *Last First* without a comma the parts will never be transposed.

abbreviations In certain fields abbreviations may be used. In the example above the journal name `jams` will be expanded by LaTeX to J. Amer. Math. Soc. Abbreviations for journals and publishers can be defined with two commands provided for this purpose:

`\DefineJournal` takes four arguments: the abbreviation, the ISSN, a short form of the journal name, and the full journal name. For example,

```
\DefineJournal{mpcps}
  {0305-0041}
  {Math. Proc. Cambridge Philos. Soc.}
  {Mathematical Proceedings of the
   Cambridge Philosophical Society}
```

For `\DefinePublisher` the four arguments are: abbreviation, short form, full publisher name, and location, e.g.,

```
\DefinePublisher{ucp}
  {Univ. Chicago Press}
  {University of Chicago Press}
  {Chicago}
```

Beware of variation among the books of a single publisher in the cities of publication.

If the `amsrefs` package is invoked with the `jpa` option, it will automatically load an auxiliary package `amsjpa` that contains `\DefineJournal` and `\DefinePublisher` statements for more than a hundred of the journals and publishers mentioned most often in AMS bibliographies. To get a different set of abbreviations, you can put your own definitions in a `.sty` file and load them with `\usepackage`.

capitalization Proper nouns do not need to be written with extra braces: it is sufficient to write `Riemann` instead of `{R}iemann` or `{Riemann}` (as required by BibTeX). Do not capitalize words that are not proper nouns (unless you're writing in German, of course). Capitalization for English-language titles will be applied automatically where specified by the bibliography style. See **Capitalization of English titles**, below.

ndash Using `\ndash` instead of `--` for en-dashes is recommended. (And `\mdash` instead of `---`, for that matter.) See the remarks on `textcmds` in the **Auxiliary Packages** section.

date It is recommended to use a `date` field to give the year of publication; although `year` is also accepted, `date` is more general. If the date includes a month or month and day, using month numbers in ISO 8601 form is recommended, e.g., 1987-12 (or 1987-12-30 if a day is present). This allows month names to be printed in full or abbreviated (or left as numeric), at the behest of the current bibliography style, without changing the contents of the bibliography. For "Winter", "Spring", "Summer", "Fall", either use month numbers of 13, 14, 15, 16 (respectively), or just put in the text before the year:

```
date={Summer 1987},
```

The first mandatory argument of `\bib` is the citation key to be used with `\cite`. Like `\bibitem`, `\bib` also takes an optional argument to be used as the item label in the bibliography and as the printed output of the `\cite` command.

The second mandatory argument of `\bib` is the entry type. The recognized entry types are just about the same as the ones commonly recognized in BibTeX style files, except that `phdthesis` and `masterthesis` are subsumed under a single `thesis` category. By default an entry of this type is treated as a master's thesis; to indicate some other kind, one can write, e.g., `type={Ph.D. thesis}` or `type={Diplomarbeit}`. As a special case the abbreviations `type={phd}` and `type={masters}` are also recognized.

Restricted key-value scanning

Although the fields within a `\bib` command are given in standard LaTeX key-value notation, the parser used to scan the keys and their values is not the standard one from LaTeX's `keyval` package but one written especially for the `amsrefs` package in order to provide some refinements in the error checking. It is embodied as a separate package, `rkeyval`.

Missing commas If a comma is missing in a `\bib` command, you get an error message that tells you exactly what the problem is and where. With the standard key-value parsing provided by LaTeX this would not happen — a missing comma would lead to one erroneous value and one lost value without any warning to the user. For example, consider what happens with

```
\rotatebox[x=9pt y=9pt]{180}{UPSIDE DOWN}
```

This results in a value for `x` of "`9pt y`", while the second `9pt` is discarded and `y` retains its default value. Consequently the rotated box is positioned incorrectly and a spurious letter y is printed on the

page when LaTeX attempts to use the value of x. And there is no warning message to alert the user that something went wrong.

To be sure, when key-value notation is used for options that specify *how* some material should be printed, the values tend to be short and simple, they are normally written without braces, and missing commas are not all that frequent. But when the values contain textual material, and each key-value pair is given in braces on a separate line, missing comma errors are very easy to make and occur quite often in practice. It was the sinking feeling of this realization that drove me in the end to write an alternative parser, with great reluctance, after using the standard parser for nearly the whole development period of the amsrefs package.

Mandatory use of braces for values Requiring braces all the time is better for key-value pairs if the values are printable text. Because, of course, braces are the only completely reliable way to avoid the problems that afflict delimited arguments if any kind of nesting is involved or if the argument may legitimately contain the delimiter string.

If the braces are inadvertently left out, the error message looks like this (more or less):

```
! Package rkeyval Error:
         Missing open brace for key value.
 ...
l.10    year=1985,

?
```

In most cases LaTeX will be able to carry on after such an error and produce something close to the intended output.

Capitalization of English titles

The capitalization recommended for English titles is *sentence case*: all lowercase except for proper nouns (i.e., the same as is normally used anyway for all other languages with a Latin-1 character set). Initial caps can be applied on demand to a title written in this form, as explained in the discussion of \bibspec below.

The capitalization done by amsrefs succeeds rather well in following the rules given in the *Chicago Manual of Style*, as paraphrased here:

> Capitalize each word except for articles, coordinate conjunctions, and prepositions, or the word *to* in infinitives. Do capitalize pronouns and subordinate conjunctions. Always capitalize the first and last word of the title and the first and last word of any subtitles that it may contain. In a hyphenated compound, capitalize the second (or any later) word only if it is a noun or proper adjective, or it has equal force with the first word.

If some word is capitalized that should not be capitalized, putting braces around the word will prevent that. But I think it will be very seldom necessary in practice; in a test of some two hundred titles taken at random from AMS journal articles, all of the titles were capitalized correctly, even the ones containing more difficult fragments such as hyphenated compounds, math formulas, or \␣ and ~ for inter-word spaces.

Citations

The amsrefs package offers three primary citing commands: \cite, \citelist, \cites. Features include:

- sorting and range compression for numeric cite keys (like the excellent cite package)
- support for author-year citation schemes, with some additional commands \ycite, \ocite, \citeauthor, etc.
- back-reference capabilities (similar to those of the backref package that comes with hyperref)

The \cite command works just about the way the LaTeX book says it should, except that it supports an additional optional argument mechanism that avoids a pitfall associated with the standard optional argument syntax. Instead of \cite[Chapter 2]{xyz}, one writes

```
\cite{xyz}*{Chapter 2}
```

Using this * notation instead of the usual square brackets prevents the unexpected error messages that novice users meet if they incautiously attempt to use a \cite command with the square brackets inside another pair of optional argument brackets, for example,

```
\begin{thm}[\cite[Theorem 4.9]{xyz}]
```

The \citelist command takes one argument, which is simply a list of \cite commands, optionally separated by spaces.[1] Each \cite command may have its own optional argument.

```
\citelist{\cite{key1}
  \cite{key2}*{Chapter 2} \cite{key3}}
```

The \cites command is a straightforward variant of the \citelist command that can be used when none of the individual \cite commands have

[1] No commas or other inter-cite punctuation should be written between the \cite commands because that will all be supplied automatically and attempts to write it in by hand will only interfere.

an optional argument. Then one can give the list of cite keys without including a `\cite` command for each one:

```
\cites{key1,key2,...}
```

Use of multiple cite keys with `\cite` is deprecated, even though it must be supported for backward compatibility. Using `\cites` or `\citelist` is better because it clears up the semantically dubious old treatment of the optional argument.

Author-year citation schemes

In author-year citation schemes three main citing forms are required to cover the cases that in other schemes require only one form. The first form is used when the citation serves as a parenthetical annotation — i.e., it could be omitted without harming the grammatical structure of the sentence containing it. For example:

> The question first arose in systems theory (Rupp and Young, 1977).

The second form is like the first but is used when the author name is already present as a natural part of the sentence, and the cite therefore ought to supply only the year:

> Rupp and Young (1977) have investigated ...

The third form is used when the citation serves as a direct object or other inomissible noun-like object within its sentence. For instance, dropping the citation from the following sentence leaves a grammatically incomplete remainder:

> ... for further details, see Rupp and Young (1977).

The logic of requiring three forms is perhaps most clearly seen if we envision replacing the author-year citations by numerical citations. The type of text replaced by [14] is different in each case:

> ... arose in systems theory [14].
>
> Rupp and Young [14] have investigated ...
>
> ... for further details, see [14].

We delegate `\cite` to produce the primary parenthetical form (Author, Year) and provide `\ycite` ("year cite") and `\ocite` ("object cite") as the other forms. Plural forms `\ycites` and `\ocites` are also provided in parallel with `\cites`. And `\citeauthor` can be used to produce the list of author names without the year.

These correspond with command names from some other commonly used author-year packages as follows:

amsrefs	harvard	natbib
`\cite`, `\cites`	`\cite`	`\citep`
`\ycite`, `\ycites`	`\citeyear`	`\cite`
`\ocite`, `\ocites`	`\citeasnoun`	`\citet`
`\citeauthor`	(none)	`\citeauthor`
`\citeauthory`	(none)	`\citet`

Some people like to use `\citet` or `\citeasnoun` when the author name serves as the subject of a sentence. This seems to me a questionable blurring of the boundary between the essential text of the sentence and the bibliography pointer. But just in case I am mistaken (though you may gasp in disbelief at the thought), I have provided `\citeauthory{xyz}` as an abbreviation for

```
\citeauthor{xyz} \ycite{xyz}
```

I.e., the command name is `\citeauthor`+`\ycite` with the redundant second `cite` dropped.[2]

When an author-year scheme is in use, parens are normally added by `\cite`, `\citelist` and their variants — unless the character immediately following the command's argument is a closing paren. This simple rule of thumb suffices for almost all cases.

Starred forms `\cite*` and `\ocite*` print the full list of author names instead of an abbreviated list, when an (Author, Year) style of citation is in use. For other citation schemes they produce the same output as the unstarred forms. Some citation styles require the first cite to use the full list and subsequent ones to use the abbreviated version. With proper setup this can be done automatically, so that the starred forms of these commands should seldom be necessary in practice.

Bibliography style setup

With the `amsrefs` package all style changes can be done with LaTeX. You don't need to understand BibTeX's unnamed `bst` language. And because everything is handled from the LaTeX side, the bibliography style for a given document class can be specified completely in the class file instead of partly there and partly in a `.bst` file.

The overall style of the bibliography list is dictated by a `biblist` environment; it takes an optional argument which may contain overrides of list parameters and other style specs. This makes it easy to modify the style slightly for a particular document. Documentclasses that load the `amsrefs` package should supply their own definition of `biblist`, and doing so normally means that they can leave `thebibliography` environment unchanged, because

[2] Why not simply call it `\aycite`, you may ask? Well, I could hardly pass up the opportunity to get all six vowels in a single command name, could I?

amsrefs makes it a simple wrapper function that calls biblist:

```
\renewenvironment{thebibliography}[1]{%
  \bibsection
  \biblist[\resetbiblist{#1}]%
}{%
  \endbiblist
}
```

where \bibsection is normally defined as

```
\newcommand{\bibsection}{%
  \section*{\bibname}}%
```

When defining \bibsection, the amsrefs package uses \chapter if it is defined, otherwise \section. Of course you can always redefine \bibsection yourself if need be.

The interior formatting within entries is specified by \bibspec commands, one for each entry type. To illustrate, let's look at an example style spec for entries of type article:

```
\bibspec{article}{%
  +{}{\PrintAuthors}  {author}
  +{,}{ \textit}      {title}
  +{,}{ }             {journal}
  +{}{ \textbf}       {volume}
  +{}{ \parenthesize} {date}
  +{,}{ }             {pages}
  +{,}{ }             {note}
  +{.}{}              {transition}
  +{}{ }              {review}
}
```

It should be pretty obvious that each line specifies the formatting for a particular field. The fundamental model here is that after reading the data for a particular \bib command, LaTeX steps through the style spec and for each field listed, prints the field with the given formatting *if and only if the field has a nonempty value*. The + character at the beginning of each field spec must be followed by three arguments: the punctuation to be added if the field is nonempty; space and/or other material to be added after the punctuation; and the field name. It is permissible for the second part to end with a command that takes an argument, such as \textbf, in which case it will receive the field's value as its argument. By defining a suitable command and using it here you can place material after the field contents as well as before; \parenthesize is an example of this.

The reason that the punctuation and the following space are specified separately is that between them there is a crucial boundary for line breaks. If you put a \linebreak command at the end of a field value, the break point will actually be carried onward to a suitable point after the next bit of punctuation (whose actual value may vary depending on which of the following fields is the first to turn up with a nonempty value).

The meaning of the \parenthesize command, supplied by amsrefs, should be obvious. The meaning of the \PrintAuthors command is a different story. But I don't think it is all that hard to understand. If we have two or three author names which were given separately, and we need to combine them into a conventional name list using commas and the word "and", then it would be nice if we had a command which could take a list of names and Do The Right Thing. And that is just what \PrintAuthors is.

The rkeyval package allows keys to be defined as additive: if the key occurs more than once, each successive value will be concatenated to the previous value, along with a prefix. The setup done by amsrefs for the author field is

```
\DefineAdditiveKey{bib}{author}{\name}
```

This means that if two names are given, as in

```
author={Bertram, A.},
author={Wentworth, R.},
```

then the final value of the author field seen when LaTeX processes the style spec will be

```
\name{Bertram, A.}\name{Wentworth, R.}
```

The transition field in our bibspec example is a dummy field to be used when punctuation or other material must be added at a certain point in the bibliography without regard to the emptiness or non-emptiness of the fields after it. The transition field always tests as non-empty but has no printed content. So when you use it you always get the indicated punctuation and space at the indicated point in the list of fields. If it were the last thing in this bibspec example, it could serve just to put in the final period that is always wanted. But in AMS bibliographies, if a Mathematical Reviews reference is given, it is conventionally printed *after* the final period. Using the transition field as shown here ensures that the final period will be always printed, even when the review field is empty.

Miscellaneous commands provided by the amsrefs package Most of the following commands are helper commands for use in \bibspec statements. The others are intended for use in bibliography data.

\parenthesize This command adds parentheses around its argument. It is useful in \bibspec statements because there is no special provision for adding material after the field value.

`\bibquotes` This command is much like `\parenthesize` but it adds quotes around its argument and it has one other important difference: there are special arrangements to print the closing quote *after* a following comma or similar punctuation (unless the `amsrefs` package is invoked with the `logical-quotes` option, in which case `\bibquotes` puts the closing quote immediately after the quoted material).

`\voltext` The normal definition of this command is `vol.~` to supply the text that precedes a volume number.

`\editiontext` This command produces `ed.` following an edition number. See `\PrintEdition` for more information.

`\pptext` This command is similar in spirit to `\voltext` but more complicated in its implementation. It takes one argument which is expected to contain one or more page numbers or a range of page numbers. The argument is printed with a prefix of "p." if it seems to be a single page number, otherwise with a prefix of "pp.".

`\tsup, \tsub, \tprime` These are for text subscripts and superscripts, with `\tprime` producing a superscript prime symbol. Unlike the standard `\textsuperscript` and `\textsubscript` functions provided by LaTeX, these do not use math mode at all.[3]

`\nopunct` This command causes following punctuation to be omitted if it is added with the internal function `\@addpunct` (which is used throughout the `\bibspec` handling when appending a non-empty field).

`\PrintAuthors` This is a relatively complicated function that tests a list of author names in order to decide whether `\sameauthors` or `\aulist` should be called.

`\aulist` This takes a list of author names in the form

```
\name{Jones, Sam}\name{Smith, John}...
```

and prints them in standard series form with the names uninverted, e.g., "Sam Jones and John Smith", or "Sam Jones, John Smith, and James Taylor".

`\UninvertedNames` This is a lower-level function called by `\aulist`. For documentation see `amsrefs.dtx`.

`\sameauthors` This is a function of one argument. If you use the default set of `\bibspecs` from `amsrefs`, `\sameauthors` is applied to the author name for a given `\bib` command if it matches exactly the author name of the preceding `\bib` command. Change the definition of `\sameauthors` if you don't want to get a bysame dash.

`\bysame` This is a horizontal rule of length 3 em. The default definition of `\sameauthors` prints `\bysame` instead of the author names.

`\PrintEditorsA` This is similar to `\aulist` but adds `(ed.)` following the editor name (or `(eds.)` if applicable).

`\PrintEditorsB` This is like `\PrintEditorsA` but puts parentheses around the entire list of editor names.

`\Plural, \SingularPlural` These are helper functions for use with `\UninvertedNames` that allow you to conditionally print singular or plural forms such as `(ed.)` or `(eds.)` depending on the number of names in the current name list. The definition of `\PrintEditorsA` reads, in part,

```
... (ed\Plural{s}.) ...
```

`\ReviewList` This is similar to `\aulist` but is used for printing (possibly multiple) MR numbers given in the `review` field.

`\inicap` This command applies initial capitalization to its argument.

`\EnglishInitialCaps` This command will call `\inicap` if and only if the language of the current reference is English. If English is the default language, you need to specify

```
language={German},
```

(for example) for non-English references to ensure that nothing will be initial-capped.

`\BibField` This is for more complicated programming tasks such as may be necessary for some bib-specs. It takes one argument, a field name, and yields the contents of that field for the current `\bib` entry.

`\IfEmptyBibField` If one writes

```
\IfEmptyBibField{isbn}{A}{B}
```

then the commands in A will be executed if the `isbn` field is empty, otherwise the commands in B.

`\PrintEdition` If a bibliography entry has

```
edition={2}
```

and the bib-spec used `\PrintEdition` to handle this field, then the edition information will be printed as "2nd ed." — that is, the number is converted to cardinal form and "ed." is added (taken from `\editiontext`).

[3] There is one drawback: If you don't want to get the prime symbol for `\tprime` from the cmsy font, you will need to redefine `\tprime` in some suitable way.

`\CardinalNumeric` This provides the conversion to cardinal number form used by `\PrintEdition`.

`\PrintDate`, `\PrintYear` These functions convert a date in canonical form (ISO 8601) to the form required by the current bibliography style. You can get your preferred date form by redefining these functions or by changing your `\bibspec` statements to use another function of your own devising. The original definition of `\PrintDate` adds parens (as for the year of a journal article in normal AMS style), whereas the `\PrintYear` function simply prints the year without any additional material (as for a book's year of publication in normal AMS style).

`\SerialName` When a journal abbreviation is used as the content of the `journal` field, `\SerialName` will be called with three arguments from the results of the abbreviation lookup: ISSN number, short form, and long form. The default definition of `\SerialName` prints the short form; by changing the definition suitably you can arrange to get the long form instead.

`\PublisherName` This is like `\SerialName` but gets short form, long form, and city as its arguments; the default again is to print the short form.

`\mdash`, `\ndash` These are short forms for `\textemdash` and `\textendash`, recommended instead of the more usual `---` and `--` notation. From the `textcmds` package.

`\cite`, `\cites`, `\citelist`, `\ycite`, `\ocite`, etc. See the section on **Citations**.

Fields recognized by the `\bib` command

The following fields are supported out of the box by the `amsrefs` package. It should be emphasized that this list is not cast in stone. Additional fields can be supported by defining them with commands from the `rkeyval` package and modifying your `\bibspec` statements to take them into account.

address	issn	series
archive	journal	setup
author	label	status
booktitle	language	subtitle
conference	note	title
date	number	translator
doi	organization	type
edition	pages	volume
editor	part	xid
eprint	place	xref
ios	publisher	year
isbn	review	

For some of these a few explanatory remarks are in order.

archive The archive that holds the eprint listed in the `eprint` field.

author This field can be repeated. Multiple author names will be concatenated into a single field value. Names should be given in *Last, First* order.

date This is a generalization of the `year` and `month` fields. Its value should be written in ISO 8601 format, e.g., 1987-06-05; but the day and month are omissible, so this can be used instead of the `year` field.

doi Digital Object Identifier

edition For books. If the value of this field is a simple number, `\bib` will convert it to cardinal form and add "ed.".

editor Like `author`; can be repeated.

eprint Electronic preprint information such as for www.arXiv.org.

ios Institution, organization, or school. Replaces three different BibTeX field names.

isbn International Standard Book Number. Can be repeated.

issn International Standard Serial Number.

language Language of the work. The default language is English; however, this can be changed for an entire bibliography by redefining the following variable:

```
\renewcommand{\biblanguagedefault}{French}
```

The language name should be the printed form, not Babel-style language names, since in principle this field could contain more complicated remarks such as "Russian, with French abstract".

organization The BibTeX documentation says that `institution` should be used for technical reports and `organization` for other entry types, whereas `school` should be used for theses. Having three different field names for these strikes me as overkill, so I introduced `ios` as a substitute. But `organization` is retained as a synonym, for the sake of those who don't like overly cryptic short names.

part This is for a long journal article that is published in separate parts.

place A synonym for address. Or to put it another way, `address` is supported as a BibTeX-compatible synonym for `place`.

review A review number or similar pointer, e.g., for Mathematical Reviews or Zentralblatt. Can be repeated.

setup This is a special field that can be used to give arbitrary commands to be executed at the

beginning of the current `\bib` entry, after all the fields have been read. The idea is that one can alter the formatting of an individual entry through this field, to handle special cases.

status Typically used for notes such as "to appear" or "in preparation" with journal articles.

subtitle Typically used with a multipart journal article to give a subtitle for each part.

translator Like `author`; can be repeated.

url Universal Resource Locator.

xid This is used by a cross-referenced item to pass its identity to child entries that refer to it.

Miscellaneous features

Here are some miscellaneous features that might be of interest.

- Duplicate `\bib` keys are identified on the first LaTeX run and the line number is given. (With `\bibitem` you don't get a warning until the processing of the `.aux` file at the beginning of the second LaTeX run, nor any line number.)
- When a cite key is undefined, the cite command prints the key, not question marks.

Package options

The `amsrefs` package supports the following options. The options that are listed together are mutually exclusive. The options whose name begins with a star are relevant only for BibTeX use and when any such option is changed, the effects will not take hold until after the next BibTeX run.

`?` Information about the `amsrefs` package. If you use the `jpa` option as well, the most obvious effect of this option is that all the available journal and publisher abbreviations are shown on screen (and in the LaTeX log).

`traditional-quotes`, `logical-quotes` This option changes the action of the `\bibquotes` command. When a field is appended after a quoted field, the closing quote is moved if necessary to fall after a comma or similar punctuation instead of before. If the `logical-quotes` option is chosen, ending quotes are not moved.

`sorted-cites`, `non-sorted-cites` Relevant only when numeric cites are in use. Lists of two or more cites are sorted into numerical order.

`compressed-cites`, `non-compressed-cites` Relevant only when numeric cites are in use. Three or more consecutive cite numbers will be converted to range notation (using `\ndash`).

`short-journal-names`, `full-journal-names` These options only work for journals that are specified via abbreviations. Otherwise, of course, you have either the full journal name or the short-form journal name in your data and that's all you've got.

`short-month-names`, `full-month-names` This should be fairly easy to guess.

`initials` Convert authors' first and middle names to initials.

`jpa` Load the standard AMS journal and publisher abbreviations package.

`backrefs` Print back-reference page numbers in the bibliography.

`numeric`, `alphabetic`, `author-year` This option specifies the printed format to be used for cites.

`*sorted`, `*citation-order` This option is passed on to BibTeX (the `amsxport` style) and indicates that when producing the `.bbl` file the entries should be sorted or left in order of first citation, respectively. (The default is sorted.)

Auxiliary packages

The following components of the `amsrefs` package are written in package form for reasons of modularity or to facilitate using them elsewhere.

textcmds This package provides `\mdash`, `\ndash` and other commands to replace the TeX notations for ligatures that are "ligatures of convenience" rather than of esthetics — in effect, all the standard ligature combinations that consist of punctuation characters rather than letters. The fact that the ligatures of convenience lead to quite a bit of trouble in font substitutions and document conversion suggests that they are fundamentally flawed as a markup device. If you still like the convenience of typing two or three hyphens to get a dash instead of some longer sequence, my suggestion is to use the capabilities of your text editor to automatically convert `--` to `\ndash` as you write. You will find a copy of the Emacs setup that I use for this purpose in the `dtx` file for the `textcmds` package.

inicap This package provides the basic `\inicap` function which is called by `\EnglishInitialCaps`.

rkeyval This package provides the more restrictive key-value parser used in processing the contents of a `\bib` command.

ifoption This package provides a way of testing the presence or absence of particular options:

```
\IfOption{jpa}{
  \RequirePackage{amsjpa}[2000/02/02]
}{}
```

The \@ifpackagewith test that is already provided by LaTeX returns false for options that were only switched on with \ExecuteOptions, but not mentioned explicitly in the \usepackage call.

In order for \IfOption to work properly the options must be declared with \DeclareExclusiveOptions or \DeclareBooleanOption and processed with an additional statement \ProcessExclusiveOptions; see the package documentation for further information.

amsjpa This package simply contains a number of abbreviation definitions for the journals and publishers that are most frequently mentioned in AMS bibliographies.

BibTeX exporting — amsxport style

There are four chief functions of BibTeX: selecting items required by a particular document from a database, sorting them, discarding unwanted fields, and applying LaTeX formatting. If the formatting is handled instead entirely on the LaTeX side, and if the author uses editor facilities such as the fine Emacs RefTeX package to handle selecting and sorting, it is a short step to wonder if BibTeX might be approaching superfluity. Since my work on the amsxport BibTeX style involved a certain amount of beating my head against various limitations of BibTeX, I am indeed inclined to advocate abandoning it. (For example, there are only global variables (easy to clobber something inadvertently), and an extraordinarily low ceiling on the number of them; there's no way to get the actual ASCII length of a string (text.length$ counts {\foo} as one character, not six)[4]; and there is no way to call up a given cite individually, e.g., to examine a crossref'd item.) If you convert your existing .bib files to amsrefs format (using the amsxport style) you will never again have to worry about adding braces for proper nouns or extra braces around accented letters when adding material to your database. And if you let RefTeX build your bibliography incrementally while you write, you can get all your cites resolved on the first LaTeX run, not the fourth.

For most users of the amsrefs package it will remain necessary, however, to use BibTeX at least on occasion to convert .bib file data to amsrefs form with the amsxport filter. Therefore it will be of interest to mention a few of its features.

- The supported fields are the standard BibTeX fields plus the additional fields that amsrefs supports (see the list above).
- When sorting names, the amsxport style sorts "von Something" under "Something", not under "von".
- Options can be passed to the bst file; separate bst files are not needed to handle such variations as sorting or not sorting. For example, the *citation-order option of the amsrefs package works by writing \bibcite{[citation-order]} to the .aux file: cite keys that begin with a [character are specially interpreted as options by the amsxport style.
- In preamble strings, ^^M is recognized as an escape sequence to put in newlines at good places. With a BibTeX style that doesn't provide this feature (meaning every one except amsxport, at the present time, to the best of my knowledge), all of the text strings that you supply with @preamble{...} get mushed into a single "paragraph" by BibTeX; there is no good way that I know of to ensure that a real newline gets written to the .bbl file at any given point. (The bad way: put in percent characters or other junk to fill up 72-character lines.)

[4] Well, you can write a loop to chop off one character at a time until the remainder is empty, and I did so, but the fact that I had to do so is the limitation.

Availability

A beta version of the package is currently available at ftp://ftp.ams.org/pub/tex/.

References

[1] Pedro J. Aphalo, *A proposal for citation commands in LaTeX3*, *TUGboat* 18/4 (December 1997), 297–302.

[2] Nelson Beebe, xbtxbst.doc, http://www.math.utah.edu/pub/tex/bibtex/

[3] *Chicago Manual of Style*, 13th edition, 1982, University of Chicago Press.

[4] Dana Jacobsen, bp, *a Perl Bibliography Package* (version 0.2.97 beta, 19 December 1996), http://www.ecst.csuchico.edu/~jacobsd/bib/bp/index.html

[5] Michael Piotrowski, Jens Klöcker, and Jörg Knappen, *Is LaTeX 2ε markup sufficient for scientific articles?* EuroTeX'99 Proceedings (Giessen, Augsburg, 1999), ISBN 1438-9959.

[6] David Rhead, *The "operational requirement" for support of bibliographies*, *TUGboat* 14/4 (December 1993), 425–433.

[7] Reinhard Wonneberger and Frank Mittelbach, *BibTeX reconsidered*, *TUGboat* 12/1 (March 1991), 111–124.