

# The `amsrefs` package

Michael Downes and David M. Jones  
American Mathematical Society

Version 2.12, 2013/01/16

## Contents

|        |   |    |
|--------|---|----|
| 1      | Introduction . . . . .  | 3  |
| 2      | Package options . . . . .   | 3  |
| 3      | More about the <code>\bib</code> command . . . . .                            | 3  |
| 3.1    | Field names for the <code>\bib</code> command . . . . .                       | 3  |
| 3.2    | Bibliography entry types . . . . .  | 4  |
| 4      | Customizing the bibliography style . . . . .                                  | 5  |
| 5      | Miscellaneous commands provided by the <code>amsrefs</code> package . . . . . | 6  |
| 6      | Implementation . . . . .  | 9  |
| 6.1    | Overview . . . . .  | 9  |
| 6.1.1  | Normal L <sup>A</sup> T <sub>E</sub> X processing of cites . . . . .          | 9  |
| 6.2    | How cites are processed by <code>amsrefs</code> . . . . .                     | 10 |
| 6.3    | Data structures . . . . .   | 11 |
| 6.4    | Preliminaries . . . . .   | 12 |
| 6.5    | Utilities . . . . .   | 13 |
| 6.6    | Declaring package options . . . . .   | 16 |
| 6.6.1  | The <code>?</code> option . . . . .   | 18 |
| 6.7    | Loading auxiliary packages . . . . .  | 18 |
| 6.7.1  | The <code>lite</code> option . . . . .  | 18 |
| 6.8    | Key-value setup . . . . .   | 18 |
| 6.8.1  | Standard field names (the <code>bib</code> group) . . . . .                   | 19 |
| 6.8.2  | Auxiliary properties (the <code>prop</code> group) . . . . .                  | 20 |
| 6.9    | Bibliography type specifications . . . . .                                    | 20 |
| 6.10   | The standard bibliography types . . . . .                                     | 23 |
| 6.11   | The <code>\biblist</code> environment . . . . .                               | 27 |
| 6.12   | Processing bibliography entries . . . . .                                     | 30 |
| 6.12.1 | <code>\@bibdef</code> Implementations . . . . .                               | 32 |
| 6.12.2 | <code>\bib@exec</code> Implementations . . . . .                              | 35 |
| 6.12.3 | Resolving cross-references . . . . .  | 37 |
| 6.12.4 | Bib field preprocessing . . . . .   | 40 |
| 6.12.5 | Date setup . . . . .  | 42 |
| 6.12.6 | Language setup . . . . .  | 43 |
| 6.12.7 | Citation label setup . . . . .  | 44 |

|         |   |     |
|---------|---|-----|
| 6.12.8  | Printing the bibliography . . . . .   | 47  |
| 6.13    | Name, journal and publisher abbreviations . . . . .                                       | 49  |
| 6.14    | Processing <code>.ltb</code> files . . . . .  | 49  |
| 6.15    | Citation processing . . . . .   | 53  |
| 6.15.1  | The <code>\citesel</code> structure . . . . .   | 53  |
| 6.15.2  | The basic <code>\cite</code> command . . . . .  | 53  |
| 6.15.3  | Fancier <code>\cite</code> commands . . . . .   | 59  |
| 6.15.4  | The <code>\nocite</code> command . . . . .  | 60  |
| 6.15.5  | Citation sorting . . . . .  | 61  |
| 6.15.6  | Range compression . . . . .   | 62  |
| 6.15.7  | Munging the <code>.aux</code> file . . . . .  | 65  |
| 6.15.8  | Back references . . . . .   | 67  |
| 6.15.9  | <code>hyperref</code> , <code>showkeys</code> and <code>shaderef</code> support . . . . . | 68  |
| 6.16    | Lexical structure of names . . . . .  | 69  |
| 6.16.1  | Text accents . . . . .  | 69  |
| 6.16.2  | Text symbols . . . . .  | 70  |
| 6.16.3  | <code>\edef</code> -like macros for names . . . . .                                       | 72  |
| 6.17    | Name parsing . . . . .  | 74  |
| 6.18    | Extracting initials . . . . .   | 78  |
| 6.18.1  | The algorithm . . . . .   | 79  |
| 6.18.2  | The implementation . . . . .  | 80  |
| 6.19    | Generating alphabetic labels . . . . .  | 86  |
| 6.19.1  | The algorithm . . . . .   | 86  |
| 6.19.2  | The implementation . . . . .  | 87  |
| 6.20    | Generating short alphabetic labels . . . . .  | 91  |
| 6.21    | Formatting series . . . . .   | 92  |
| 6.22    | Formatting names and series of names . . . . .  | 96  |
| 6.23    | The <code>partial</code> field . . . . .  | 102 |
| 6.24    | Special formatting for other fields . . . . .   | 102 |
| 6.25    | BIB <sub>TEX</sub> support . . . . .  | 108 |
| 6.26    | Implementing package options . . . . .  | 108 |
| 6.26.1  | The <code>alphabetic</code> option . . . . .  | 108 |
| 6.26.2  | The <code>shortalphabetic</code> option . . . . .   | 108 |
| 6.26.3  | The <code>backrefs</code> option . . . . .  | 108 |
| 6.26.4  | The <code>citation-order</code> option . . . . .  | 109 |
| 6.26.5  | The <code>initials</code> option . . . . .  | 109 |
| 6.26.6  | The <code>logical-quotes</code> option . . . . .  | 110 |
| 6.26.7  | The <code>non-compressed-cites</code> option . . . . .                                    | 110 |
| 6.26.8  | The <code>non-sorted-cites</code> option . . . . .  | 110 |
| 6.26.9  | The <code>short-journals</code> option . . . . .  | 110 |
| 6.26.10 | The <code>short-publishers</code> option . . . . .  | 110 |
| 6.26.11 | The <code>short-months</code> option . . . . .  | 110 |
| 6.26.12 | The <code>y2k</code> option . . . . .   | 111 |
| 6.26.13 | The <code>bibtex-style</code> option . . . . .  | 111 |
| 6.26.14 | The <code>msc-links</code> option . . . . .   | 111 |
| 6.26.15 | The <code>author-year</code> option . . . . .   | 112 |

|      |   |     |
|------|---|-----|
| 6.27 | The <code>amsbst</code> package . . . . . | 119 |
|------|---|-----|

## 1 Introduction

The `amsrefs` package is a  $\LaTeX$  package for bibliographies that provides an archival data format similar to the format of `BibTeX` database files, but adapted to make direct processing by  $\LaTeX$  easier. The package can be used either in conjunction with `BibTeX` or as a replacement for `BibTeX`.

This document is written for anyone who wants to implement a new bibliography style for `amsrefs` or who is just curious about how the package is implemented. The reader should be familiar with the contents of the “User’s Guide to the `amsrefs` Package” [1] (`amsrdoc.tex`).

For the publisher or implementor, the chief advantages of the `amsrefs` package are as follows:

**Preservation of structure** The internal structural information of the bibliography entries is not lost when they are imported from the database file into the  $\LaTeX$  document. This takes on its greatest significance when archiving documents in  $\LaTeX$  form or transmitting them to another user (such as a publisher).

**Deferred formatting** This means that the style of the bibliography can be readily changed without reimporting everything from the original database(s).

**Setup requires only  $\LaTeX$  knowledge** All bibliography setup can be done in  $\LaTeX$ ; learning another programming language (such as the one used in `BibTeX` `bst` files) is unnecessary.

## 2 Package options

In addition to the options documented in the user’s guide, there are a few additional options that were omitted either because they are obsolete or deprecated options included only for backwards compatability or because they are still considered experimental and not yet ready for widespread use.

? Informational option. This causes `amsrefs` to display a pointer to the User’s Guide on the terminal and in the log file. (In previous versions, it displayed much more material, including a summary of package options.)

**traditional-quotes, logical-quotes** With the *traditional quotes* option (default), quotation marks produced by `\bibquotes` (§5) fall outside of other punctuation, “like this,” whereas with the *logical quotes* option the order is reversed, “like this”.

## 3 More about the `\bib` command

### 3.1 Field names for the `\bib` command

In addition to the fields discussed in the user’s guide, the following fields are used internally:

**fulljournal** Used internally by `\DefineJournal`.

**name** Used internally by the **name** bibliography type and `\DefineName`.

**transition** A dummy field used inside `\BibSpecs` when we want to force an action unconditionally.

The following fields are included for backwards compatibility:

**institution, school** These are provided as aliases for **organization** for compatibility with `BIBTEX`.

**place** A synonym for **address**. In earlier versions of `amsrefs`, **place** was preferred and **address** was considered as an alias for **place**. However, this seemed like a gratuitous incompatibility with `BIBTEX` to me, so I have reinstated **address** as the primary field and **place** is now an undocumented alias.

The following fields are reserved for future use:

**doi** Digital Object Identifier

**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.

This is fully implemented, but I've been unable to think of any good examples of its use; so, I've decided to suppress it until such an example comes to light.

**url** Universal Resource Locator.

### 3.2 Bibliography entry types

The following additional entry types (or, really, pseudo-entry types) are used internally by `amsrefs`:

**collection.article**

**proceedings.article**

**partial**

**conference**

**innerbook**

**name**

**nameLE**

**nameBE**

**nameinverted**

**publisher**

The following are currently undocumented aliases for various of the standard types:

**miscellaneous**

**periodical**

## 4 Customizing the bibliography style

If you use the `amsrefs` package as is, the bibliography style you get is the kind of style customarily seen in AMS publications. The recommended way to get a different bibliography style is to write a  $\text{\LaTeX}$  package which loads the `amsrefs` package with `\RequirePackage` and then makes the desired changes by using suitable `\BibSpec` commands as explained below. Thus, the general form of the custom package will be

```
\ProvidesPackage{xyzbib}[2002/11/06 v1.28]

\RequirePackage{amsrefs}\relax

\BibSpec{article}{
  ...
}

\BibSpec{book}{
  ...
}
```

The interior formatting within entries is specified by `\BibSpec` commands, one for each entry type. To illustrate, let's look at an example style specification 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. After reading the data for a particular `\bib` command,  $\text{\LaTeX}$  steps through the style specification 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 specification 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 more 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  $\text{\LaTeX}$  processes the style specification 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.

## 5 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` This is used to format volume numbers. By default, it precedes the volume number by “vol.”
- `\issuetext` This is used to format issue numbers. By default, it precedes the volume number by “no.”
- `\editiontext` This command produces “ed.” following an edition number. See `\PrintEdition` for more information.
- `\DashPages` 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 L<sup>A</sup>T<sub>E</sub>X, these do not use math mode at all.<sup>1</sup>
- `\nopunct` This command causes following punctuation to be omitted if it is added with the internal function `\@addpunct`.
- `\PrintPrimary` This is a relatively complicated function that determines the “primary” contributors for an entry and formats them, or replaces them by `\sameauthors` if appropriate. It should be used when an entry type might have editors or translators instead of authors. It prefers authors over editors and editors over translators and generates a warning if there are no primary contributors.
- `\PrintAuthors` This is used to format the list of authors as the primary contributors for an entry type.
- `\PrintEditorsA` This is similar to `\PrintAuthors` but adds (ed.) or (eds.) following the editors.
- `\PrintEditorsB` This is similar to `\PrintEditorsA` but puts parentheses around the entire list of editors. It’s used by, for example, the `article` type to print the editors of a `proceedings` or `collection`.
- `\PrintEditorsC` Similar to `\PrintEditorsA` but precedes the editors by `Edited by`. It’s used when the editors should be treated as subsidiary contributors, rather than the primary contributor.
- `\PrintTranslatorsA` This is similar to `\PrintEditorsA` but adds (trans.) following the translators.
- `\PrintTranslatorsB` This is similar to `\PrintEditorsB`. It’s not currently used, but is provided for symmetry.
- `\PrintTranslatorsC` Similar to `\PrintEditorsC` but precedes the translators by `Translated by`.

---

<sup>1</sup>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.

- `\sameauthors` This is a function of one argument. If you use the default set of `\BibSpecs` from the `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.
- `\Plural`, `\SingularPlural` These are helper functions 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}.) ...
```
- `\PrintReviews` This is similar to `\AuthorList` but is used for printing (possibly multiple) MR numbers given in the `review` field.
- `\BibField` This is for more complicated programming tasks such as may be necessary for some `\BibSpecs`. 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 `\BibSpec` 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`).
- `\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 parentheses (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).
- `\mdash`, `\ndash` These are short forms for `\textendash` and `\textendash`, recommended instead of the more usual --- and -- notation. From the `textcmds` package.
- et cetera ...** [mjd,2002-01-03] See the `.dtx` files for further possibilities that I have not managed to get properly documented yet!



## 6 Implementation

### 6.1 Overview

It will be a while yet before we get to any actual code. First we need to understand what the code needs to accomplish in order to provide the user interface described above in a way that is as compatible as possible with existing  $\LaTeX$  mechanisms.

#### 6.1.1 Normal $\LaTeX$ processing of cites

**First  $\LaTeX$  pass** Various commands are written to the `.aux` file that are mostly used by  $\BibTeX$ .

1. A `\cite{moo}` command writes one line to the `.aux` file: `\citation{moo}`. This indicates to  $\BibTeX$  that it should include ‘moo’ in the list of cited items to be searched for. The `\cite` command also checks to see if `\b@mo` contains the corresponding citation label, but since this is the first pass, the label won’t be known yet, so  $\LaTeX$  emits an ‘Undefined citation’ warning and prints a placeholder (i.e., `???`) instead of the citation label.
2. A `\bibliographystyle{har}` command writes one line to the `.aux` file: `\bibstyle{har}`. This indicates to  $\BibTeX$  that it should use `har.bst` to determine the style for sorting and formatting the bibliography items.
3. A `\bibliography{hij,klm,...}` command writes one line to the `.aux` file: `\bibdata{hij,klm,...}`. This indicates to  $\BibTeX$  that it should look in `hij.bib`, `klm.bib`, ... for bibliographic data. The `\bibliography` also tries to input the `.bbl` file, but on the first pass it won’t exist yet.

On the first pass all `\cite`’s normally are reported as undefined because the `.bbl` file has not yet been created.

**$\BibTeX$  pass** For a document named `xyz.tex`, the command `bibtex xyz` is used to invoke  $\BibTeX$ . It looks in `xyz.aux` to find the citation information written there by  $\LaTeX$ . For each `\citation` line,  $\BibTeX$  searches for a corresponding entry in the specified `.bib` files and formats it. The entire list is then sorted in whatever way dictated by the bibliography style, and written out to the file `xyz.bbl`. This normally produces entries that look something like:

```
\bibitem{BGL} P. Busch, M. Grabowski and P. J. Lahti:
{\it Operational Quantum Physics.}
Springer Verlag, New York (1995).
```

**Second  $\LaTeX$  pass** Now the `.bbl` file exists and contains some `\bibitem` commands. At `\begin{document}`,  $\LaTeX$  reads the `.aux` file, hoping to find some `\bibcite` commands, but it will not find them until the next time around. `\citation`, `\bibstyle`, and `\bibdata` commands in the `.aux` file are simply ignored by  $\LaTeX$ . Then  $\LaTeX$  proceeds to typeset the body of the document.

1. Instances of `\cite` still print question marks.
2. The `\bibliography` command causes  $\LaTeX$  to input `xyz.bbl` and typeset its contents.

3. A `\bibitem{moo}` command writes one line to the `.aux` file: `\bibtite{moo}{9}`, where 9 is the current item number.
4. A `\bibitem[Moody]{moo}` command writes one line to the `.aux` file: `\bibtite{moo}{Moody}`, using the supplied label instead of a number.

**Third L<sup>A</sup>T<sub>E</sub>X pass** Now the `.aux` file contains some `\bibtite` commands. Once again, L<sup>A</sup>T<sub>E</sub>X reads the `.aux` file when it reaches `\begin{document}`.

1. A `\bibtite{moo}{Moody}` causes L<sup>A</sup>T<sub>E</sub>X to define `\b@moo` with ‘Moody’ as the replacement text.
2. If two `\bibtite` commands have the same citation key, L<sup>A</sup>T<sub>E</sub>X gives a warning message. This happens at `\begin{document}`, during the reading of the `.aux` file.
3. Instances of `\cite` in the body of the document will print the appropriate labels obtained from the `.aux` file.
4. If there are any `\cite` commands for which the `.aux` file did not have a `\bibtite` command, L<sup>A</sup>T<sub>E</sub>X will give an ‘Undefined citation’ warning. This often happens if the `.aux` file is incomplete due to a T<sub>E</sub>X error on the preceding pass.

## 6.2 How cites are processed by `amsrefs`

In order to support its additional features (e.g., author-year citations and the `backrefs` option), the `amsrefs` package stores additional information for each cite in the macro `\b@whatever`. Instead of simply using the defined or undefined status of this macro to trigger the standard warnings, we add some boolean flags to allow us to discriminate more finely what the current situation is.

- Each time an item is cited in the body of the document, a `backref` entry is added to the info of that item. The `backref` info is the current page and section location. Section location is a bit hard to get right without better support from the document class. So we provide a hook to allow it to work better when the support is there.
- When a cite occurs, if the info is undefined then a warning is issued and the info structure is created. A `\citation` command and a `\citedest` command (providing backref info) are written to the `.aux` file. Because the backref info includes page number, it has to be a non-immediate write. An undefined info structure would normally happen only on a first pass when no `.aux` file exists, or when a new cite is added. I.e., when the corresponding `\citation` command is not yet present in the `.aux` file.
- When a citation command occurs in the `.aux` file, it initializes the info structure if necessary, setting the “bib-info-present” flag to 0.
- When a `\citedest` command occurs in the `.aux` file, it initializes the info structure if necessary—but this shouldn’t happen: if the corresponding `\citation` command did not already get processed, then something is wrong. So normally, the `\citedest` command merely needs to add its backref info to the existing info structure.

- When a `\babcite` command occurs in the `.aux` file, it will normally find that `\b@whatever` is already defined, if the bibliography occurs after all the `\cite` commands. What it must do is fill in the appropriate blank slots in the info structure set up by a previous `\citation` command.
- The `.aux` file is actually processed two times, once at the beginning of the document and once at the end. In the latter case, `\babcite` should give a warning if the backref-list is empty, since that means there were no `\cite` commands for the given key.
- When processing the bibliography: The `\bib` command needs to check if it is using a key that is already used by another `\bib` command.

We therefore have

```
\b@xyz -> \citesel 00{label}{year}{backref-list}
```

where the first 0 is replaced by 1 if there has already been another citation for the same key earlier in the document (some citation styles use abbreviated forms for all instances after the first), and the second 0 is replaced by 1 if the same key was already used by an earlier `\bib` command.

Because the backref-list often includes page number information, it cannot be built on the fly as we go along; instead we have to write the information to the `.aux` file and read it in at the beginning of the next run.

If there was no `\babcite` in the `.aux` file for a given key, then the info is

```
\b@xyz -> \citesel 00{}{}{backref-list}
```

If there was neither `\citation` nor `\babcite` in the `.aux` file for a given key, then the `\cite` command should find that `\b@xyz` is undefined.

If the author-year option is in effect, the “label” contains the author last names instead of a label:

```
\b@xyz -> \citesel 00{\name{Smith}\name{Jones}}{...}{...}
```

Full name information is included in the data because some citation styles give full names at the first citation and abbreviated forms for subsequent instances.

### 6.3 Data structures

The result of scanning the key/value pairs of a `\bib` command is an assignment statement for `\rsk@toks`. (Cf. the `rkeyval` package.) For example, consider the entry

```
\bib{miller83}{article}{
  author={Miller, G.},
  title={Eine Bemerkung zur Darstellung von Polynomen \{"u}ber
    Verb\{a}nden}*{language={german}}},
  journal={J. Math. Sent.},
  volume={10},
  year={1983},
  pages={26\ndash 30},
}
```

The scanned result is to assign

```
\global\rsk@toks{%
```

```

\set:bib'author{Miller, G.}{}%
\set:bib'title{Eine Bemerkung zur Darstellung von Polynomen
  \"{u}ber Verb\{a}nden}{language={german}}%
\set:bib'journal{J. Math. Sent.}{}%
\set:bib'volume{10}{}%
\set:bib'year{1983}{}%
\set:bib'pages{26\ndash 30}{}%
}

```

The code in the last arg of `\RestrictedSetKeys` then invokes `\bib@exec` to do something with the value of `\rsk@toks`.

```
\bib@exec{miller83}{\the\rsk@toks}{\setbib@article}{}
```

## 6.4 Preliminaries

```
1 <*pkg>
```

Standard declaration of package name and date.

```
2 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3 \ProvidesPackage{amsrefs}[2013/01/16 v2.12]
```

```
\amsrefs@warning@nl
```

```
4 \def\amsrefs@warning@nl{\PackageWarningNoLine{amsrefs}}
```

Backward handling for beta and jpa options.

```
5 \@ifpackagewith{amsrefs}{beta}{%
6   \amsrefs@warning@nl{The beta option is obsolete}%
7 }{}
8 \@ifpackagewith{amsrefs}{jpa}{%
9   \amsrefs@warning@nl{The jpa option is obsolete}%
10 }{}

11 \IfFileExists{url.sty}{%
12   \RequirePackage{url}\relax
13   \@gobble
14 }{%
15   \@firstofone
16 }
17 {
18   \DeclareRobustCommand{\url}[1]{%
19     \def\@tempa{#1}%
20     \texttt{\@urlsetup $\expandafter\strip@prefix\meaning\@tempa$}%
21   }%
22   \def\@urlsetup{%
23     \check@mathfonts \textfont\@ne\the\font \textfont\z@\the\font
24     \@apply\@urlfix{\do+\do=\do\:\do-\do\.\do\;\do\;}%
25     \@apply\@urlbreak{\do\&\do\/\do\?}%
26   }%
27   \def\@urlbreak#1{%
28     \mathcode' #1="8000
29     \begingroup \lccode'\~=' #1 \lowercase{\endgroup \edef~}%
30     {\mathchar\number' #1\penalty\hyphenpenalty}%
31   }%

```

```

32 \def\urlfix#1{%
33   \mathcode'#1='#1\relax
34 }%
35 }
36 \ifundefined{NormalCatcodes}{\RequirePackage{pccatcode}\relax}{-}
37 \PushCatcodes\NormalCatcodes
38 %% WARNING WARNING WARNING: Catcode of apostrophe ' is letter
39 %% throughout this file.
40 \catcode'\'=11 % letter

```

## 6.5 Utilities

Some of these useful functions are also found in AMS document classes.

`\after@deleting@token` Similar in concept to `\afterassignment`, except it deletes the next token in the stream before putting its argument back into the input. Useful for skipping past tokens during parsing.

```

41 \def\after@deleting@token#1{%
42   \afterassignment#1%
43   \let\@let@token= % Don't delete this space!
44 }

```

`\@ifempty` Some frequently used tests for empty arguments. Note that an argument consisting entirely of spaces (e.g., `\@ifempty{ }`) counts as empty.

```

\@ifnotempty
45 \long\def\@ifempty#1{\@xifempty#1@@..\@nil}
46
47 \long\def\@xifempty#1#2@#3#4#5\@nil{%
48   \ifx#3#4\@xp\@firstoftwo\else\@xp\@secondoftwo\fi
49 }
50
51 \long\def\@ifnotempty#1{\@ifempty{#1}}

```

`\macrotext`

```

52 \def\macrotext{\expandafter\strip@prefix\meaning}

```

`\vdef` “Verbatim” def.

```

53 \def\vdef#1#2{%
54   \def#1{#2}%
55   \edef#1{\macrotext#1}%
56 }

```

`\auto@protect` Sometimes it's convenient to render a given control sequence unexpandable for a time. `\auto@protect` provides a way to do that.<sup>2</sup>

An earlier version of this code read simply `\let#1\relax` but that had the disadvantage of making all `\auto@protected` macros compare equal via `\ifx`. This version allows macros to keep their identities under comparisons.

```

57 \def\auto@protect#1{\def#1{\@nx#1}}

```

<sup>2</sup>There really should be a special name for macros that, like `\auto@protect`, take a control sequence as an argument and redefine that control sequence in order to achieve some special effect. Pending happier inspiration, I'm going to call them “wrapper” macros.

`\g@undef` Globally undefine a control sequence.  
 58 `\def\g@undef#1{\global\let#1\relax}`

`\@concat` Concatenate onto the end of a token list. Expands everything.  
 59 `\def\@concat#1#2{\edef#1{#1#2}}`

`\add@toks@` This saves a few tokens of main memory and a lot of typing.  
 60 `\def\add@toks@{\addto@hook\toks@}`

`\@lappend` Append an element to a `\do`-delimited list. As long as the element to be appended (`#2`) is a single token, nothing is expanded. If it contains multiple tokens, all tokens after the first will be expanded.  
 61 `\def\@lappend#1#2{%`  
 62 `\begingroup`  
 63 `\def\do{\@nx\do\@nx}%`  
 64 `\edef\@tempa{\def\@nx#1{#1\do#2}}%`  
 65 `\@xp\endgroup`  
 66 `\@tempa`  
 67 `}`

`\@apply` Apply a macro to each element of a `\do`-delimited list.  
 68 `\def\@apply#1#2{%`  
 69 `\let\do#1%`  
 70 `#2%`  
 71 `}`

`\get@numberof` This is a generic macro for counting the number of elements in a L<sup>A</sup>T<sub>E</sub>X-style list. The first argument is a `\count` register that will receive the final count; the second argument is the control sequence that separates elements of the list, and the third argument is the list itself. So, for example,

```
\get@numberof\@tempcnta\do\dospecials
```

would count the number of special characters in `\dospecials` and store the number in `\@tempcnta`.

```
72 \def\get@numberof#1#2#3{%
73   \begingroup
74     \def#2{\advance\@tempcnta\@ne \@gobble}%
75     \@tempcnta\z@
76     #3\relax
77     \edef\@tempb{#1=\the\@tempcnta\relax}%
78   \@xp\endgroup
79   \@tempb
80 }
```

`\safe@set` This is a quick and dirty way of extracting an integer prefix from a string and assigning it to a counter. If the string does not begin with an integer, the counter receives the value 0. The suffix after the integer prefix is discarded. (But bad things will happen if the string contains the token `\@nil`.)

```

81 \def\safe@set#1#2{%
82   \afterassignment\@nilgobble
83   #1=0#2\relax\@nil
84 }

```

`\@chomp` Vaguely reminiscent of Perl's `chomp` function, which removes a substring from the end of a variable, but ours works with tokens (more-or-less) and takes the substring to be removed as its second argument. Note the use of `\@empty` to anchor the chomped substring to the end of the string. Note also that the second argument will be fully expanded during the chomping.

```

85 \def\@chomp#1#2{%
86   \begingroup
87   \toks@\@emptytoks
88   \def\@chomper##1##2#2\@empty##3\@nil{%
89     \ifx\@let@token\bgroup
90       \toks@{##1}##2}%
91     \else
92       \toks@{##1##2}%
93     \fi
94   }%
95   \@xp\chomp@ #1\@empty#2\@empty\@nil
96   \edef\@tempa{\def\@nx#1\@xp{the\toks@}}%
97   \@xp\endgroup
98   \@tempa
99 }

```

`\chomp@` Before passing control to `\@chomper`, we peek ahead at the next token in the stream. That way, if the next token is an open brace, we know we need to surround `\@chomper`'s first argument with braces. Unfortunately, this might still remove braces from the second argument, but I think that's ok for our purposes.

```

100 \def\chomp@{%
101   \futurelet\@let@token
102   \@chomper
103 }

```

`\amsrefs@warning`

```

104 \def\amsrefs@warning{\PackageWarning{amsrefs}}

```

`\amsrefs@error`

```

105 \def\amsrefs@error{\PackageError{amsrefs}}

```

`\MessageBreakNS` This suppresses the leading space in `\on@line` in error and warning messages.

```

106 \def\MessageBreakNS{\MessageBreak\romannumeral'\^^@}

```

`\@addpunct` The `\@addpunct` function is defined by AMS document classes and the `amsgen` package. But if we find it undefined we had better define it.

```

107 \@ifundefined{\@addpunct}{%

```

```

108 \def\@addpunct#1{%
109     \relax\ifhmode
110         \ifnum\spacefactor>\@m \else#1\fi
111     \fi
112 }
113 \def\frenchspacing{%
114     \sfcode'\.1006
115     \sfcode'\?1005
116     \sfcode'\!1004
117     \sfcode'\:1003
118     \sfcode'\;1002
119     \sfcode'\,1001\relax
120 }
121 }-{}

```

`\nopunct` Omit any following punctuation that would normally be inserted by `\@addpunct`.

```
122 \providecommand{\nopunct}{\spacefactor \@nopunctsfcode}
```

`\@nopunctsfcode`

```
123 \def\@nopunctsfcode{1007 }
```

## 6.6 Declaring package options

We call the `ifoption` package to facilitate some option tests.

```
124 \RequirePackage{ifoption}[2000/02/15]
```

The `sorted` option is a no-op and is no longer documented. I'm only leaving it here for backwards compatibility.

```
125 \DeclareExclusiveOptions{sorted,citation-order}
```

The `alphabetic` option corresponds to the standard `alpha` biblio style with labels like `Knu66` (three letters from name plus two digits of year). Maybe should provide an alias `LIYY` for this option. Numeric is the default since it is commoner in AMS publications.

```
126 \DeclareExclusiveOptions{alphabetic,shortalphabetic,author-year,numeric}
```

`y2k`

```
127 \DeclareBooleanOption{y2k}
```

`nobysame`

```
128 \DeclareBooleanOption{nobysame}
```

The standard `abbrv` bibliography style uses abbreviations for month names and journal names, and first names of people are abbreviated to their initials. Since the second test bibliography that I tested with had unabbreviated month names but abbreviated journal names, perhaps it is a good idea to let these choices be specified separately.

```
129 \DeclareBooleanOption{short-journals}
```

```
130 \DeclareBooleanOption{short-publishers}
```



The `short-journals` and `short-publishers` options only affect journal and publisher names that are defined with `\DefineJournal` and `\DefinePublisher` commands.

```
131 \DeclareBooleanOption{short-months}
132 \DeclareBooleanOption{initials}
```

Nevertheless, it's to be expected that the preceding four options would typically be used together, so we provide a short-hand for requesting them all.

```
133 \DeclareOption{abbrev}{%
134     \@pass@options
135     \@currentx
136     {initials,short-months,short-journals,short-publishers}%
137     \@currname
138 }
```

In the bibliography, if a title or something is enclosed in quotes, should the closing quotes go inside the punctuation (logical position) rather than outside (traditional)? These options give you a choice.

```
139 \DeclareExclusiveOptions{traditional-quotes,logical-quotes}
```

A sequence of cites will be sorted and ranges of length three or greater will be compressed if these options so indicate. Note that the `non-sorted-cites` option automatically disables compression. This is probably a feature.

```
140 \DeclareExclusiveOptions{sorted-cites,non-sorted-cites}
141 \DeclareExclusiveOptions{non-compressed-cites,compressed-cites}
```

In the bibliography, print page numbers showing where each entry was cited.

```
142 \DeclareBooleanOption{backrefs}
```

Option for giving information about the available options:

```
143 \DeclareBooleanOption{?}
```

This option means to forgo loading of the `textcmds` and `mathscinet` packages.

```
144 \DeclareBooleanOption{lite}
```

This option can be used by later releases as a sign that fall-back adaptations need to be done.

```
145 \DeclareBooleanOption{beta}
```

```
146 \DeclareBooleanOption{bibtex-style}
```

```
147 \DeclareBooleanOption{msc-links}
```

```
148 \ExecuteOptions{numeric,traditional-quotes,sorted-cites,compressed-cites}
```

```
149
```

```
150 \ProcessOptions\relax
```

```
151
```

```
152 \ProcessExclusiveOptions
```

```
153 \IfOption{backrefs}{%
```

```
154     \IfFileExists{backref.sty}{%
```

```
155         \RequirePackage{backref}[1999/05/30]
```

```

156   }{%
157     \amsrefs@warning@nl{The backrefs option cannot be used^^J%
158       unless the backref package is also installed.^^J%
159       (backref is part of the hyperref package)}%
160   }%
161 }{}
162
163 \IfOption{msc-links}{%
164   \IfFileExists{hyperref.sty}{%
165     \RequirePackage{hyperref}[1999/07/08]
166   }{
167     \amsrefs@warning@nl{The msc-links option cannot be used^^J%
168       unless the hyperref package is installed}%
169   }%
170 }{}

```

### 6.6.1 The ? option

```

171 \IfOption{?}{%
172   \typeout{^^J%
173     Documentation for the amsrefs package is found in amsrdoc.dvi^^J%
174     (or .pdf or .tex).
175     ^^J%
176   }%
177 }{}%

```

## 6.7 Loading auxiliary packages

Now, if these other packages make use of the `pcatcode` package like they should, then we don't need to make any fuss here about the special catcode of `'`. Just load the packages.

```
178 \RequirePackage{rkeyval}[2001/12/22]
```

### 6.7.1 The lite option

In my opinion, this is misguided, since `amsrefs` shouldn't be loading these packages to begin with. But it's too late to change it now.

```

179 \IfOption{lite}{% True? Then don't load the next two packages.
180 }{% False? OK, let's load them:
181   \RequirePackage{textcmds}[2001/12/14]
182   \RequirePackage{mathscinet}[2002/01/01]
183 }

```

## 6.8 Key-value setup

`\BibField` This provides easy access to individual fields for user-defined formatting functions.

```
184 \newcommand{\BibField}[1]{\csname bib'#1\endcsname}
```

`\IfEmptyBibField` A convenient partial application of `\rkvIfEmpty`.

```
185 \newcommand{\IfEmptyBibField}{\rkvIfEmpty{bib}}
```

### 6.8.1 Standard field names (the bib group)

And here are the predefined key names. You could always add some more if you needed them. Only worry is about compatibility if you want to share your data with other people.

`\fld@elt` We want the list macros used above to be unexpandable except when special processing is done. (It's not clear to me there's any real benefit to using these instead of just using `\do.—dmj`)

```
186 \let\fld@elt=?
187 \let\name=?
```

First the fields that could be repeated more than once in a single entry. Maybe publisher should be allowed to repeat also, for co-published works. But then need to worry about the address handling.

```
188 \DefineAdditiveKey{bib}{author}{\name}
189 \DefineAdditiveKey{bib}{editor}{\name}
190 \DefineAdditiveKey{bib}{translator}{\name}
191 \DefineAdditiveKey{bib}{contribution}{\fld@elt}
192 \DefineAdditiveKey{bib}{isbn}{\fld@elt}
193 \DefineAdditiveKey{bib}{issn}{\fld@elt}
194 \DefineAdditiveKey{bib}{review}{\fld@elt}
195 \DefineAdditiveKey{bib}{partial}{\fld@elt}

196 \DefineSimpleKey{bib}{accessdate}
197 \DefineSimpleKey{bib}{address}
198 \DefineSimpleKey{bib}{book}
199 \DefineSimpleKey{bib}{booktitle}
200 \DefineSimpleKey{bib}{conference}
201 %\DefineSimpleKey{bib}{contributor}
202 \DefineSimpleKey{bib}{copula}
203 \DefineSimpleKey{bib}{date}
204 \DefineSimpleKey{bib}{doi}
205 \DefineSimpleKey{bib}{edition}
206 \DefineSimpleKey{bib}{eprint}
207 \DefineSimpleKey{bib}{fulljournal}
208 \DefineSimpleKey{bib}{hyphenation}
209 \DefineSimpleKey{bib}{institution}
210 \DefineSimpleKey{bib}{journal}
211 \DefineSimpleKey{bib}{label}
212 \DefineSimpleKey{bib}{language}
213 \DefineSimpleKey{bib}{name}
214 \DefineSimpleKey{bib}{note}
215 \DefineSimpleKey{bib}{number}
216 \DefineSimpleKey{bib}{organization}
217 \DefineSimpleKey{bib}{pages}
218 \DefineSimpleKey{bib}{part}
219 \DefineSimpleKey{bib}{place}
220 \DefineSimpleKey{bib}{publisher}
221 \DefineSimpleKey{bib}{reprint}
```

```

222 \DefineSimpleKey{bib}{school}
223 \DefineSimpleKey{bib}{series}
224 \DefineSimpleKey{bib}{setup}
225 \DefineSimpleKey{bib}{status}
226 \DefineSimpleKey{bib}{subtitle}
227 \DefineSimpleKey{bib}{title}
228 \DefineSimpleKey{bib}{translation}
229 \DefineSimpleKey{bib}{type}
230 \DefineSimpleKey{bib}{url}
231 \DefineSimpleKey{bib}{volume}
232 \DefineSimpleKey{bib}{xref}
233 \DefineSimpleKey{bib}{year}

```

The `transition` key is used when we want to insert punctuation or other material at a given point in the sequence unconditionally. The key appears to have a non-empty value to `\IfEmptyBibField`, but its value (expansion) is empty.

```
234 \DefineDummyKey{bib}{transition}
```

### 6.8.2 Auxiliary properties (the `prop` group)

```

235 \DefineSimpleKey{prop}{inverted}
236 \DefineSimpleKey{prop}{language}

```

## 6.9 Bibliography type specifications

`\BibSpec` Accumulate specification material in `\toks@`, then define `\setbib@TYPE` from it.

```

237 \newcommand{\BibSpec}[2]{%
238   \toks@\@emptytoks
239   \@ifnotempty{#2}{%

```

The `\@ifnextchar` removes an optional `+` at the beginning of a specification. From then on, each time `\bibspeg@scan` is invoked, it expects to find four arguments. The four `\@emptys` appended to the specification (`#2`) below ensure that this is so.

```

240     \ifnextchar+{\@xp\bibspeg@scan@gobble}{\bibspeg@scan}%
241     #2\@empty\@empty\@empty\@empty
242   }%
243   \@xp\edef\csname setbib@#1\endcsname{\the\toks@}%
244 }

```

`\bibspeg@scan` The `\bibspeg@scan` function scans one field specification from the second arg of `\BibSpec`. Each field specification has the form

```
+{punctuation}{prelim material}{field name}
```

Note however that because the initial `+` is stripped off by `\BibSpec` (see above), the actual order that `\bibspeg@scan` reads the field specification is

```
#1={punctuation} #2={prelim material} #3={field name} #4=+
```

where the fourth argument is actually expected to be either the `+` from the following specification, or one of the special `\@empty` tokens inserted by `\BibSpec`.

If it is neither of these special values, it means we have a malformed specification; so, we issue an error and then try to pick up where we left off.

```

245 \def\bibspec@scan#1#2#3#4{%
246   \add@toks@\bib@append{#1}{#2}}%
247   \edef\@tempa{%
248     \toks@\the\toks@ \exp\@nx\csname bib'#3\endcsname}%
249   }%
250   \@tempa
251   \ifx\@empty#4%
252     \exp@gobble % end the recursion
253   \else
254     \ifx +#4\else\bibspec@scan@error\fi
255   \fi
256   \bibspec@scan
257 }

```

`\bibspec@scan@error`

```

258 \def\bibspec@scan@error{\amsrefs@error{Bad BibSpec: Expected '+'}}

```

`\bib@append` The function `\bib@append` prints the value of a field, together with associated punctuation and font changes, unless the value is empty. Arg 1 is punctuation (that may need to be swapped with a preceding line break), arg 2 gives the space to be added after the punctuation, and possibly a function to be applied to the contents of arg 3, which is a macro containing the field value. So if we have `\moo` and `\bib'pages`, from `pages={21\ndash 44}`, then we want to arrange to call

```
\moo{21\ndash 44}
```

We don't want to simply call `\moo\bib'bar` because that makes it rather difficult for `\moo` to look at the contents of `\bib@bar`.

```

259 \def\bib@append#1#2#3{%
260   \ifx\@empty#3%
261   \else
262     \ifx\relax#3%
263       \errmessage{#3=\relax}%
264     \else
265       \begingroup
266         \series@index\m@ne
267         \def\current@bibfield{#3}%
268         \@ifempty{#1}{%
269           \@temptokena{\ifnum\lastkern=\@ne\ignorespaces\fi #2}%
270         }{%
271           \@temptokena{\SwapBreak{#1}{#2}}%
272         }%
273         \toks@\exp{#3}%
274         \edef\@tempa{\the\@temptokena{\the\toks@}}%

```

*Known bug:* Need better error message here.

```

275         \rkvIfAdditive#3}{-%
276         \get@current@properties
277         \select@auxlanguage
278         }%
279         \@tempa
280     \endgroup
281 \fi
282 \fi
283 }

\select@auxlanguage

284 \def\select@auxlanguage{%
285     \ifx\prop'language\@empty
286     \else
287         \xp\selectlanguage\@xp{\prop'language}%
288     \fi
289 }

```

`\erase@field` There are some fields that can appear in more than one place in a reference, depending on context. For example, if a book has an editor but no author, the editor appears at the beginning of the entry, but if the book has both an editor and an author, the editor appears at the end of the entry. A simple way to handle this is to “erase” the `editor` field after printing it, which is what `\erase@field` is for.

The obvious definition of `\erase@field` is

```
\def\erase@field#1{\global\let#1\@empty}
```

but that doesn’t work because the top-level value of `rkeyval` fields isn’t `\@empty`; instead, it contains a setter function used by `\RestrictedSetKeys` when processing a key-value list (see `\rkv@DSAK`, `\rsk@set@a` and `\rsk@set@b`).

On the other hand, rewriting the field locally won’t work either, since `\erase@field` will typically be executed inside the group established by `\bib@append`. Instead, we want to rewrite the value right after `\bib@append`’s group ends. One way to do this would be to keep a list of fields to be erased and have `\bib@append` iterate over the list after its `\endgroup`.

However, as long as the call to `\erase@field` is never nested within any deeper groups, it’s simpler just to use `\aftergroup`, which is what we’ll do (“Sufficient unto the day is the evil thereof” and all that).

```

290 \def\erase@field#1{%
291     \aftergroup\let\aftergroup#1\aftergroup\@empty
292 }

```

`\get@current@properties` This retrieves the auxiliary properties for the current field value, as defined by `\current@bibfield` and `\series@index`.

```

293 \def\get@current@properties{%
294     \begingroup
295         \xp\get@nth@property\@xp\@tempa\current@bibfield\series@index

```

```

296     \edef\@tempa{%
297         \@nx\RestrictedSetKeys}{prop}{%
298         \def\@nx\@tempa{\@nx\prop@reset \@nx\the\@nx\rsk@toks}%
299         }{\@tempa}%
300     }%
301     \@tempa
302     \@xp\endgroup
303     \@tempa
304 }

```

`\BibSpecAlias` This is a `\def` rather than a `\let` because using `\let` would make `\BibSpecAlias` statements order-sensitive in a way that seems frequently to be a stumbling block to unwary package writers. But then we should probably do at least the simplest kind of infinite loop check.

```

305 \newcommand{\BibSpecAlias}[2]{%
306     \@xp\def\@xp\@tempa\@xp{\csname setbib@#1\@xp\endcsname}%
307     \@xp\ifx\csname setbib@#2\endcsname\@tempa
308         \amsrefs@error{%
309             Mirror alias #1->#2 not allowed (infinite loop)}\@ehc
310     \else
311         \@xp\def\csname setbib@#1\@xp\endcsname
312         \@xp{\csname setbib@#2\endcsname}%
313     \fi
314 }

```

## 6.10 The standard bibliography types

```

315 \BibSpec{article}{%
316     +{ } { \PrintAuthors}           {author}
317     +{,} { \textit}                 {title}
318     +{.} { }                         {part}
319     +{:} { \textit}                 {subtitle}
320     +{,} { \PrintContributions}     {contribution}
321     +{.} { \PrintPartials}          {partial}
322     +{,} { }                         {journal}
323     +{ } { \textbf}                 {volume}

```

The date form is tricky depending on presence or absence of DOI.

```

324     +{ } { \PrintDatePV}             {date}
325     +{,} { \issuetext}               {number}
326     +{,} { \eprintpages}             {pages}
327     +{,} { }                         {status}
328     +{,} { \PrintDOI}                {doi}
329     +{,} { available at \eprint}     {eprint}
330     +{ } { \parenthesize}            {language}
331     +{ } { \PrintTranslation}        {translation}
332     +{;} { \PrintReprint}            {reprint}
333     +{.} { }                         {note}
334     +{.} { }                         {transition}
335     +{ } { \SentenceSpace \PrintReviews} {review}
336 }

```

```

337
338 \BibSpec{partial}{%
339     +{} {}                                {part}
340     +{:} { \textit}                        {subtitle}
341     +{,} { \PrintContributions}           {contribution}
342     +{,} { }                               {journal}
343     +{} { \textbf}                          {volume}
344     +{} { \PrintDatePV}                    {date}
345     +{,} { \issuetext}                     {number}
346     +{,} { \reprintpages}                 {pages}
347 }
348
349 \BibSpec{contribution}{%
350     +{} {}                                {type}
351     +{} { by \PrintNameList}              {author}
352 }
353
354 \BibSpec{book}{%
355     +{} { \PrintPrimary}                   {transition}
356     +{,} { \textit}                        {title}
357     +{.} { }                               {part}
358     +{:} { \textit}                        {subtitle}
359     +{,} { \PrintEdition}                  {edition}
360     +{} { \PrintEditorsB}                  {editor}
361     +{,} { \PrintTranslatorsC}            {translator}
362     +{,} { \PrintContributions}           {contribution}
363     +{,} { }                               {series}
364     +{,} { \voltext}                       {volume}
365     +{,} { }                               {publisher}
366     +{,} { }                               {organization}
367     +{,} { }                               {address}
368     +{,} { \PrintDateB}                   {date}
369     +{,} { }                               {status}
370     +{} { \parenthesize}                  {language}
371     +{} { \PrintTranslation}               {translation}
372     +{;} { \PrintReprint}                  {reprint}
373     +{.} { }                               {note}
374     +{.} {}                                {transition}
375     +{} { \SentenceSpace \PrintReviews}   {review}
376 }
377
378 \BibSpec{collection.article}{%
379     +{} { \PrintAuthors}                   {author}
380     +{,} { \textit}                        {title}
381     +{.} { }                               {part}
382     +{:} { \textit}                        {subtitle}
383     +{,} { \PrintContributions}           {contribution}
384     +{,} { \PrintConference}              {conference}
385     +{} { \PrintBook}                      {book}
386     +{,} { }                               {booktitle}

```



```

387   +{,} { \PrintDateB}           {date}
388   +{,} { pp.~}                 {pages}
389   +{,} { }                     {status}
390   +{,} { \PrintDOI}             {doi}
391   +{,} { available at \eprint}   {eprint}
392   +{} { \parenthesize}          {language}
393   +{} { \PrintTranslation}       {translation}
394   +{;} { \PrintReprint}         {reprint}
395   +{.} { }                     {note}
396   +{.} {}                      {transition}
397   +{} { \SentenceSpace \PrintReviews} {review}
398 }
399
400 \BibSpec{conference}{%
401   +{} {}                        {title}
402   +{} { \PrintConferenceDetails} {transition}
403 }
404
405 \BibSpec{innerbook}{%
406   +{,} { }                     {title}
407   +{.} { }                     {part}
408   +{:} { }                     {subtitle}
409   +{,} { \PrintEdition}         {edition}
410   +{} { \PrintEditorsB}        {editor}
411   +{,} { \PrintTranslatorsC}   {translator}
412   +{,} { \PrintContributions}  {contribution}
413   +{,} { }                     {series}
414   +{,} { \voltext}             {volume}
415   +{,} { }                     {publisher}
416   +{,} { }                     {organization}
417   +{,} { }                     {address}
418   +{,} { \PrintDateB}         {date}
419   +{.} { }                     {note}
420 }
421
422 \BibSpec{report}{%
423   +{} { \PrintPrimary}         {transition}
424   +{,} { \textit}             {title}
425   +{.} { }                     {part}
426   +{:} { \textit}             {subtitle}
427   +{,} { \PrintEdition}       {edition}
428   +{,} { \PrintContributions} {contribution}
429   +{,} { Technical Report }   {number}
430   +{,} { }                     {series}
431   +{,} { }                     {organization}
432   +{,} { }                     {address}
433   +{,} { \PrintDateB}         {date}
434   +{,} { \eprint}             {eprint}
435   +{,} { }                     {status}
436   +{} { \parenthesize}        {language}

```

```

437   +{} { \PrintTranslation}           {translation}
438   +{;} { \PrintReprint}             {reprint}
439   +{.} { }                          {note}
440   +{.} {}                            {transition}
441   +{} {\SentenceSpace \PrintReviews} {review}
442 }
443
444 \BibSpec{thesis}{%
445   +{} {\PrintAuthors}               {author}
446   +{,} { \textit}                  {title}
447   +{:} { \textit}                  {subtitle}
448   +{,} { \PrintThesisType}          {type}
449   +{,} { }                          {organization}
450   +{,} { }                          {address}
451   +{,} { \PrintDateB}               {date}
452   +{,} { \eprint}                  {eprint}
453   +{,} { }                          {status}
454   +{} { \parenthesize}              {language}
455   +{} { \PrintTranslation}          {translation}
456   +{;} { \PrintReprint}             {reprint}
457   +{.} { }                          {note}
458   +{.} {}                            {transition}
459   +{} {\SentenceSpace \PrintReviews} {review}
460 }
461
462 \BibSpec{webpage}{%
463   +{} {\PrintAuthors}               {author}
464   +{,} { \emph}                    {title}
465   +{:} { \emph}                    {subtitle}
466   +{} { \PrintDate}                 {date}
467   +{,} { \url}                      {url}
468   +{.} { Accessed \PrintDateField}  {accessdate}
469   +{.} { }                          {note}
470   +{.} {}                            {transition}
471 }
472 % \begin{macrocode}
473 \BibSpecAlias{periodical}{book}
474 \BibSpecAlias{collection}{book}
475 \BibSpecAlias{proceedings}{book}
476 \BibSpecAlias{manual}{book}
477 \BibSpecAlias{miscellaneous}{book}
478 \BibSpecAlias{misc}{miscellaneous}
479 \BibSpecAlias{unpublished}{book}
480 \BibSpecAlias{proceedings.article}{collection.article}
481 \BibSpecAlias{techreport}{report}

\setbib@incollection
482 \edef\setbib@incollection{%
483   \exp@nx\csname setbib@collection.article\endcsname
484 }

```

`\setbib@inproceedings`

```
485 \edef\setbib@inproceedings{%
486   \exp\@nx\csname setbib@collection.article\endcsname
487 }
```

Some more entry types for implementing abbreviations.

```
488 \BibSpec{name}{%
489   +{ } {\PrintAuthors}   {name}
490 }
491
492 \BibSpec{publisher}{%
493   +{,} { } {publisher}
494   +{,} { } {address}
495 }
```

### 6.11 The biblist environment

The `biblist` environment can be used with a section or chapter heading.  
Use a standard L<sup>A</sup>T<sub>E</sub>X counter for numbering bibliography items.

```
496 \newcounter{bib}
497 \DefineSimpleKey{biblist}{prefix}
498 \DefineSimpleKey{biblist}{labels}
```

`biblist`

```
499 \newenvironment{biblist}{%
500   \setcounter{bib}\z@
501   \@biblist
502 }{%
503   \@endbiblist
504 }
```

`biblist*`

```
505 \newenvironment{biblist*}{%
506   \@biblist
507 }{%
508   \@endbiblist
509 }
```

`\biblistfont`

```
510 \newcommand{\biblistfont}{%
511   \normalfont
512   \footnotesize
513 }
```

`\amsrefs@lbibitem` Reference processing at the AMS sometimes results in raw `\bibitem` entries being interspersed with `\bib` entries in a bibliography. For that to work, we need to modify `\@lbibitem` and `\@bibitem` to interoperate more smoothly with `amsrefs`.

```
514 \def\amsrefs@lbibitem[#1]#2{%
```

```

515 \begingroup
516   \def\CurrentBib{#2}%
517   \def\thebib{#1}%
518   \@nmbolistfalse
519   \item\leavevmode
520   \if@filesw
521     {\let\protect\noexpand
522     \immediate\write\@auxout{\string\bibcite{#2}{#1}}}%
523   \fi
524 \endgroup
525 \ignorespaces
526 }
527
528 \def\amsrefs@bibitem#1{%
529   \def\CurrentBib{#1}%
530   \item
531   \if@filesw
532     \immediate\write\@auxout{\string\bibcite{#1}{\the\value{\@listctr}}}%
533   \fi
534   \ignorespaces
535 }

```

\@biblist

```

536 \newcommand\@biblist[1][ ]{%
537   \stepcounter{bib@env}
538   \biblistfont
539   \labelsep .5em\relax
540   \let\@bibitem\amsrefs@bibitem
541   \let\@lbibitem\amsrefs@lbibitem
542   \list{\BibLabel}{%
543     \restore@labelwidth
544     \@maxlabelwidth\z@
545     \@nmbolisttrue
546     \def\@listctr{bib}%
547     \let\makelabel\bib@mklab
548     #1\relax
549   }%
550   \sloppy

```

Discourage page breaks within bibliography entries and disable them completely for entries that are less than four lines long.

```

551   \interlinepenalty\@m
552   \clubpenalty\@M
553   \widowpenalty\clubpenalty
554   \frenchspacing
555   \ResetCapSFCodes
556   \star@\@biblistsetup}{%
557 }

```

\@biblistsetup

```

558 \newcommand{\@biblistsetup}[1]{%
559   \RestrictedSetKeys{}{biblist}{\the\rsk@toks}{#1}%
560   \rkvIfEmpty{biblist}{prefix}{}{%
561     \let\amsrefs@label@prefix\biblist'prefix
562   }%
563   \rkvIfEmpty{biblist}{labels}{}{%
564     \@ifundefined{amsrefs@option@\biblist'labels}{%
565       \amsrefs@warning{Invalid label style '\biblist'labels'}%
566     }{%
567       \csname amsrefs@option@\biblist'labels\endcsname
568     }%
569   }%
570 }

```

\@endbiblist Change error for empty list (no items) to warning, to allow authors to leave their bibliography temporarily empty during writing:

```

571 \def\@endbiblist{%
572   \save@labelwidth
573   \def\@noitemerr{\@latex@warning{Empty bibliography list}}%
574   \global\let\previous@primary\@empty
575   \endlist
576 }

```

\@maxlabelwidth

```
577 \newdimen\@maxlabelwidth
```

\bib@mklab

```

578 \def\bib@mklab#1{%
579   \settowidth\@tempdima{#1}%
580   \ifdim \@tempdima > \@maxlabelwidth
581     \global\@maxlabelwidth\@tempdima
582   \fi
583   #1\hfil
584 }

```

```
585 \newcounter{bib@env}
```

\save@labelwidth

```

586 \def\save@labelwidth{%
587   \if@files
588     \immediate\write\@auxout{%
589       \string\newlabel{[bibenv:\the\c@bib@env]}{\the\@maxlabelwidth}%
590     }%
591   \fi
592 }

```

\restore@labelwidth

```

593 \def\restore@labelwidth{%
594   \@xp\ifx \csname r@[bibenv:\the\c@bib@env]\endcsname \relax

```

```

595     \resetbiblist{00}%
596   \else
597     \xp\labelwidth\csname r@[bibenv:\the\c@bib@env]\endcsname
598     \leftmargin\labelwidth
599     \advance\leftmargin\labelsep
600   \fi
601 }

```

`\ResetCapSFCodes` Presumably this is here because there has been a problem in the past with packages that change the `\catcodes` of capital letters.

```

602 \providecommand{\ResetCapSFCodes}{%
603   \count@='A
604   \def\@tempa{%
605     \sfcode\count@=\@m
606     \advance\count@\@ne
607     \ifnum\count@>'Z\relax \expandafter\@gobble \fi
608     \@tempa
609   }%
610   \@tempa
611 }

```

`\CurrentBib` In case this is undefined sometimes.

```
612 \def\CurrentBib{??}
```

`\BibLabel`

```

613 \newcommand{\BibLabel}{%
614   \hfill
615   \Hy@raisedlink{\hyper@anchorstart{cite.\CurrentBib}\hyper@anchorend}%
616   [\thebib]%
617 }

```

`\resetbiblist`

```

618 \newcommand{\resetbiblist}[1]{%
619   \settowidth\labelwidth{\def\thebib{#1}\BibLabel}%
620   \leftmargin\labelwidth
621   \ifdim\labelwidth=\z@
622     \leftmargin=1em
623     \itemindent=-\leftmargin
624   \else
625     \advance\leftmargin\labelsep
626   \fi
627 }

```

## 6.12 Processing bibliography entries

There are several things one might want to do when a `\bib` entry is encountered:

1. Format and print it. This corresponds to the direct entry of bibliography items as described in section 2.1 of the users's guide.

2. Copy it into a .bbl file. This corresponds to the use of `\bibselect` and an external .ltb database as described in section 2.2 of the user's guide.
3. Store the full information in memory. This is done by `\bib*`.

```

\bib Here is where the rubber hits the road.
628 \newcommand{\bib}{%
629     \begingroup
630     \@ifstar{%
631         \@tempswatrue
632         \let\@bibdef\star@bibdef
633         \BibItem
634     }{%
635         \@tempswafalse
636         \BibItem
637     }%
638 }

```

`\BibItem` *Arguments:*

- #1 <- *citekey*.
- #2 <- *bibtype*.

```

639 \newcommand{\BibItem}[2]{%
640     \vdef\@tempa{#1}%
641     \edef\@tempa{%
642         \edef\@nx\@tempa{\@nx\@xp\@nx\zap@space\@tempa\space\@nx\@empty}%
643     }%
644     \@tempa
645     \edef\@tempb{%
646         \@nx\@bibdef\@xp\@nx\cname setbib@#2\endcsname{#2}\@tempa}%
647     }%
648     \@tempb
649 }

```

`\@bibdef` `\@bibdef` is a pointer to the procedure that should be handed the entry's key-value pairs. It has one of four values:

1. `\star@bibdef`
2. `\normal@bibdef`
3. `\copy@bibdef`
4. `\selective@bibdef`

*Arguments:*

- #1 <- `\setbib@bibtype`.
- #2 <- *bibtype*.
- #3 <- *citekey*.

```

650 \AtBeginDocument{\let\@bibdef\normal@bibdef}

```

`\bib@exec` And `\bib@exec` is a pointer to the procedure that `\normal@bibdef` will invoke to process the key-value pairs after they've been parsed. It has one of these values:

1. `\bib@store`
2. `\bib@print`

*Arguments:*

- `#1` <- `citekey`.
- `#2` <- `\the\rsk@toks`.
- `#3` <- `\setbib@bibtype`.

```
651 \AtBeginDocument{\let\bib@exec\bib@print}
```

### 6.12.1 `\@bibdef` Implementations

`\normal@bibdef` *Arguments:*

- `#1` <- `\setbib@bibtype`.
- `#2` <- `bibtype`.
- `#3` <- `citekey`.

```
652 \def\normal@bibdef#1#2#3{%
```

`\CurrentBibType` is used by `export-bibtex`, but there might be a better way to handle it. (dmj)

```
653   \def\CurrentBibType{#2}%
654   \ifx\relax#1%
655     \amsrefs@error{Undefined entry type: #2}\@ehc
656     \let#1\setbib@misc
657   \fi
658   \RestrictedSetKeys{}{bib}%
659   {\bib@exec{#3}{\the\rsk@toks}{#1}\endgroup}%
660 }
661
662 \let\@bibdef\normal@bibdef
```

`\star@bibdef` *Arguments:*

- `#1` <- `\setbib@bibtype`.
- `#2` <- `bibtype`.
- `#3` <- `citekey`.

```
663 \def\star@bibdef{%
664   \let\bib@exec\bib@store
665   \normal@bibdef
666 }
```

`\copy@bibdef` This is a variation that copies everything into the `.bb1` file. Used by `\bibselect*` and `\bib*` inside `.ltb` files.

```
667 \def\copy@bibdef{%
668   \if@tempwa
669     \@xp\defer@bibdef
```



```

670 \else
671     \exp\copy@bibdef@a
672 \fi
673 }

```

\copy@bibdef@a

```

674 \def\copy@bibdef@a#1#2#3#4{%
675     \open@bbl@file
676     \process@xrefs{#4}%
677     \bbl@write{%
678         \string\bib\if@tempswa*\fi{#3}{#2}\string{\iffalse}\fi
679     }%

```

Since we're supplying our own definition of `\rsk@set`, we don't actually need the group argument, so we leave it out to save a few tokens.

```

680     \RestrictedSetKeys{\global\let\rsk@set\bbl@copy}\@empty
681     {\bbl@write{\iffalse{\fi\string}^^J}%
682     \endgroup}{#4}%
683 }

```

```

684 \catcode'\:=11

```

```

685
686 \def\modify@xref@fields{%
687     \let\set:bib'author\output@xref@a
688     \let\set:bib'editor\output@xref@a
689     \let\set:bib'translator\output@xref@a
690     \let\set:bib'journal\output@xref@a
691     \let\set:bib'publisher\output@xref@a
692     \def\set:bib'xref##1##2{\output@xref@{##1}\@empty}%
693     \def\set:bib'book##1##2{\output@inner@xref@{##1}\@empty}%
694     \let\set:bib'conference\set:bib'book
695     \let\set:bib'partial\set:bib'book
696     \let\set:bib'reprint\set:bib'book
697     \let\set:bib'translation\set:bib'book
698 }

```

```

699

```

```

700 \catcode'\:=12

```

```

701
702 \def\process@xrefs#1{%
703     \begingroup
704         \RestrictedSetKeys{\modify@xref@fields}{bib}{\the\rsk@toks}{#1}%
705     \endgroup
706 }

```

```

707

```

```

708 \def\output@xref@a#1#2{%
709     \def\@tempa{#1}%
710     \lowercase{\def\@tempb{#1}}%
711     \ifx\@tempa\@tempb
712         \output@xref@{#1}%
713     \fi

```

```

714 }
715
716 \def\output@xref@#1{%
717   \@ifnotempty{#1}{%
718     \@ifundefined{bi@#1}{-%
719       \begingroup
720         \let\star@bibdef\copy@bibdef@a
721         \csname bi@#1\endcsname
722       \endgroup
723     }%
724     \@xp\g@undef\csname bi@#1\endcsname
725   }%
726 }
727
728 \def\output@inner@xref@#1{%
729   \in@=#1}%
730   \ifin@\else
731     \output@xref@{#1}%
732   \fi
733 }

```

`\bbl@copy`

```

734 \def\bbl@copy#1\endcsname#2{%
735   \begingroup
736     \def\@tempa{#1}%
737     \toks@{#2}%
738     \star@\bbl@copy@a}%
739 }

```

`\bbl@copy@a`

```

740 \def\bbl@copy@a#1{%
741   \@ifnotempty{#1}{%
742     \add@toks@{*#1}%
743   }%
744   \bbl@write{ \space \@tempa=\the\toks@,}%
745   \endgroup
746   \rsk@resume
747 }

```

`\selective@bibdef` This is a variation that ignores anything not having a known citation key. Used by `\bibselect`.

*Arguments:*

```

#1 <- \setbib@bibtype.
#2 <- bibtype.
#3 <- citekey.

```

```

748 \def\selective@bibdef#1#2#3{%
749   \@xp\selbibdef@a\csname b@#3\endcsname{#1}{#2}{#3}%
750 }

```

```

\selbibdef@a
751 \def\selbibdef@a#1{%
752   \def\@tempa{\endgroup\@gobblefour}%
753   \ifx\relax#1\else \@xp\selbibdef@b#1\@nil \fi
754   \@tempa
755 }

```

```

\selbibdef@b
756 \def\selbibdef@b#1#2#3\@nil{%
757   \ifx 1#2\let\@tempa\copy@bibdef\fi
758 }

```

`\defer@bibdef` This is a variation that ignores anything not having a known citation key. Used by `\bibselect`.

*Arguments:*

```

#1 <- \setbib@bibtype.
#2 <- bibtype.
#3 <- citekey.
#4 <- key-val pairs.

```

```

759 \def\defer@bibdef#1#2#3#4{%
760   \@xp\gdef\csname bi@#3\endcsname{%
761     \bib*{#3}{#2}{#4}%
762   }%
763   \@xp\addto@defer@list \csname bi@#3\endcsname
764   \endgroup
765 }

```

```

\bibdefer@list
766 \let\bibdefer@list\@empty

```

```

\addto@defer@list
767 \def\addto@defer@list#1{%
768   \begingroup
769   \def\do{\@nx\do\@nx}%
770   \xdef\bibdefer@list{\bibdefer@list\do#1}%
771   \endgroup
772 }

```

### 6.12.2 `\bib@exec` Implementations

`\bib@store` This is the easy one. It just stores the entire set of key-value pairs in `\bi@citekey`.

```

773 \def\bib@store#1{%
774   \afterassignment\@gobble
775   \@xp\xdef\csname bi@#1\endcsname
776 }

```

`\bib@print` *Arguments:*

```

#1 <- citekey.

```

```

#2 <- \the\rsk@toks.
#3 <- \setbib@bibtype.

777 \def\bib@print#1#2#3{%
778   \bib@start{#1}%
779   \let\setbib@@#3%
780   #2\relax      % execute definitions locally
781   \bib@resolve@xrefs
782   \bib@field@patches
783   \bib@selectlanguage
784   \generate@label
785   \bib'setup
786   \bib@cite{#1}%
787   \kern\@ne sp
788   \ifx\setbib@@\setbib@article
789     \ifx\bib'booktitle\@empty
790       \ifx\bib'book\@empty
791         \ifx\bib'conference\@empty
792           \else
793             \let\setbib@@\setbib@incollection
794           \fi
795         \else
796           \let\setbib@@\setbib@incollection
797         \fi
798       \else
799         \let\setbib@@\setbib@incollection
800       \fi
801     \fi
802   \setbib@@
803   \bib@end
804 }

```

`\bib@print@inner` Note that the order of the arguments is reversed with respect to `\bib@print`. Maybe that isn't such a great idea.

*Arguments:*

```

#1 <- \setbib@bibtype.
#2 <- \the\rsk@toks.

805 \def\bib@print@inner#1#2{%
806   \begingroup
807   #2\relax      % execute definitions locally
808   \bib@resolve@xrefs
809   \bib@field@patches
810   \bib'setup
811   #1%
812   \endgroup
813 }

```

`\current@citekey`

```
814 \let\current@citekey\@empty
```

```

\prev@citekey
815 \let\prev@citekey\@empty

\bib@start There used to be more to it.
816 \def\bib@start#1{%
817     \begingroup
818     \def\current@citekey{#1}%
819 }

\bib@end Instead of being handled by \bib@end, ending punctuation is normally handled
via the transition field (q.v.)
820 \def\bib@end{%
821     \relax
822     \@xp\PrintBackRefs\@xp{\CurrentBib}%
823     \par
824     \save@primary
825     \global\let\prev@citekey\current@citekey
826     \endgroup
827 }

```

### 6.12.3 Resolving cross-references

```

\bib@resolve@xrefs
828 \def\bib@resolve@xrefs{%
829     \xref@check@c\bib'xref
830     \xref@check@a\bib'author
831     \xref@check@a\bib'editor
832     \xref@check@a\bib'translator
833     \xref@check@b\bib'journal
834     \xref@check@b\bib'publisher
835 }

\xref@check@a Resolve a contributor (typically a \DefineName) alias. Requires rebuilding the
list.
836 \def\xref@check@a#1{%
837     \ifx\@empty#1\relax
838     \else
839         \begingroup
840             \toks@\@emptytoks
841             \@temptokenb\@emptytoks
842             \series@index\z@
843             \def\name{\xref@check@aa#1}%
844             #1\relax
845             \edef\@tempa{%
846                 \def\@nx#1{\the\toks@}%
847                 \the\@temptokenb
848             }%
849             \@xp\endgroup
850             \@tempa

```

```
851 \fi
852 }
```

`\xref@check@aa`

```
853 \def\xref@check@aa#1#2{%
854   \advance\series@index\@ne
855   \def\@tempa{#2}%
856   \lowercase{\def\@tempb{#2}}%
857   \ifx\@tempa\@tempb
858     \ifx\@tempa\@empty
859       \add@toks@\{name}\}%
860     \else
861       \ifundefined{bi@#2}{%
862         \BibAbbrevWarning{#2}%
863         \add@toks@\{name{#2}}%
864       }{%
865         \xref@check@ab#1{#2}%
866       }%
867     \fi
868   \else
869     \add@toks@\{name{#2}}%
870   \fi
871 }
```

`\xref@check@ab`

```
872 \def\xref@check@ab#1#2{%
873   \csname bi@#2\endcsname
874   \ifx\@empty\bib'name
875     \@temptokena{#2}%
876   \else
877     \@temptokena\exp{\bib'name}%
878     \get@property\@tempa\bib'name
879     \edef\@tempa{%
880       \@nx\addto@hook\@temptokenb{%
881         \@nx\reset@nth@property\@nx#1\the\series@index{\@tempa}%
882       }%
883     }%
884     \@tempa
885   \fi
886   \edef\@tempa{\@nx\add@toks@\{\@nx\name{\the\@temptokena}}}%
887   \@tempa
888 }
```

`\xref@check@b` Resolve a journal or publisher alias (typically a `\DefinePublisher` or `\DefineJournal` alias).

```
889 \def\xref@check@b#1{%
890   \ifx\@empty#1%
891   \else
892     \toks@\@xp{#1}%
893   \fi
894 }
```

```

893     \edef\@tempb{\lowercase{\def\nx\@tempa{\the\toks@}}}%
894     \@tempb
895     \ifx\@tempa#1\relax % all lowercase
896         \@ifundefined{bi@#1}{%
897             \BibAbbrevWarning{#1}%
898         }{%

```

We pass control to `\xref@check@c` here to handle inheritance of multiple fields properly. This means some of the checking we've just done gets done again, but I can live with that.

```

899         \let#1\empty
900         \xref@check@c\@tempa
901     }%
902 \fi
903 \fi
904 }

```

`\xref@check@c` Resolve an xref field.

```

905 \def\xref@check@c#1{%
906     \ifx#1\empty
907     \else
908         \begingroup
909             \@apply\auto@protect\amsrefs@textsymbols
910             \@apply\auto@protect\amsrefs@textaccents
911             \let\DSK@def\xref@add@toks
912             \let\DSK@append\xref@append
913             \toks@\@emptytoks
914             \let\bib@reset\empty

```

The `\@for` here is just a fancy way of expanding `#1`. (Or is it?)

```

915     \@for\xref@ID:=#1\do{%
916         \@ifundefined{bi@\xref@ID}{%
917             \XRefWarning{\xref@ID}%
918         }{%
919             \csname bi@\xref@ID\endcsname
920         }%
921     }%
922     \edef\@tempa{\endgroup\the\toks@}%
923     \@tempa
924 \fi
925 }

```

`\xref@add@toks` If any title occurs in an xrefed item, assume that it is a book title. This might not always be the best assumption? Let's see how it goes though. [mjd,2001-12-11]

*Arguments:*

`#1` <- `\bib'` field.

`#2` <- value.

```

926 \def\xref@add@toks#1#2#3{%

```

```

927 \ifx#1\@empty
928   \edef\@tempa{%
929     \@nx\add@toks@{\@xp\@nx\csname\rkv@setter#1\endcsname{#2}{#3}}%
930   }%
931   \@tempa
932 \else
933   \in@\bib'title{#1}%
934   \ifin@
935     \ifx\bib'booktitle\@empty
936       \edef\@tempa{%
937         \@nx\add@toks@{%
938           \@xp\@nx\csname set:bib'booktitle\endcsname
939         }%
940       }%
941       \@tempa
942       \add@toks@{{#2}{#3}}%
943     \fi
944   \fi
945 \fi
946 }

947 \def\xref@append#1#2#3#4{%
948   \edef\@tempa{%
949     \@nx\add@toks@{\@xp\@nx\csname\rkv@setter#2\endcsname{#3}{#4}}%
950   }%
951   \@tempa
952 }

```

`\BibAbbrevWarning`

```
953 \def\BibAbbrevWarning#1{\amsrefs@warning{Abbreviation '#1' undefined}}
```

`\XrefWarning`

```
954 \def\XRefWarning#1{\amsrefs@warning{Xref '#1' undefined}}
```

#### 6.12.4 Bib field preprocessing

`\current@primary`

```
955 \let\current@primary\@empty
```

`\previous@primary`

```
956 \let\previous@primary\@empty
```

`\save@primary`

```

957 \IfOption{nobysame}{%
958   \let\save@primary\@empty
959 }{%
960   \def\save@primary{%
961     \global\let\previous@primary\current@primary
962   }%
963 }

```



`\bib@field@patches` Depending on your point of view, this macro either puts the bibitem into a canonical form or, alternatively, it fudges the data to fit our model. Either way, it simplifies formatting the bibliography.

```

964 \def\bib@field@patches{%
965     \ifx\bib'author\@empty
966         \ifx\bib'editor\@empty
967             \let\current@primary\bib'translator
968             \let\print@primary\PrintTranslatorsA
969         \else
970             \let\current@primary\bib'editor
971             \let\print@primary\PrintEditorsA
972         \fi
973     \else
974         \let\current@primary\bib'author
975         \let\print@primary\PrintAuthors
976     \fi
977     \ifx\bib'address\@empty
978         \let\bib'address\bib'place
979     \fi
980     \ifx\bib'organization\@empty
981         \ifx\bib'institution\@empty
982             \let\bib'organization\bib'school
983         \else
984             \let\bib'organization\bib'institution
985         \fi
986     \fi
987     \ifx\bib'date\@empty
988         \ifx\bib'year\@empty
989             \let\bib@year\bib'status
990         \else
991             \bib@parsedate\bib'year
992         \fi
993     \else
994         \bib@parsedate\bib'date
995     \fi

```

Example 21 on page 74 of *Mathematics into Type* [2] seems to indicate that when the year serves as the volume number, the date should be suppressed. If so, this is where that is done.

```

996     \def\@tempa{year}%
997     \ifx\bib'volume\@tempa
998         \let\bib'volume\bib@year
999         \let\bib'date\@empty
1000     \fi

```

Some journals have “numbers” but no “volumes”. AMS house style is to treat the number as volume.

```

1001     \ifx\setbib@\setbib@article
1002         \ifx\bib'volume\@empty
1003             \ifx\bib'number\@empty\else

```

```

1004             \let\bib'volume\bib'number
1005             \let\bib'number\@empty
1006             \fi
1007         \fi
1008     \fi

```

\bib'language is used for producing the printed rendition of the language.  
\bib@language needs to be in the form required by \selectlanguage.

```

1009     \bib@language@fixup
1010 }

```

### 6.12.5 Date setup

\bib@year

```

1011 \let\bib@year\@empty

```

\bib@month

```

1012 \let\bib@month\@empty

```

\bib@day

```

1013 \let\bib@day\@empty

```

\bib@parsedate Parse an ISO 8601 date into its year, month and day components, but without actually verifying that any of the components are numeric. Hmmm.

```

1014 \def\bib@parsedate#1{%
1015     \@xp\bib@parsedate@a#1---\@nil
1016 }

```

\bib@parsedate@a

```

1017 \def\bib@parsedate@a#1-#2-#3-#4\@nil{%
1018     \def\bib@year{#1}%
1019     \def\bib@month{#2}%
1020     \def\bib@day{#3}%

```

The rest of this macro tries to rewrite \bib'date into a normalized form. I'm not sure if this is a good idea.

```

1021     \ifx\@empty\bib@day
1022         \ifx\@empty\bib@month
1023             \let\bib'date\bib@year
1024         \else
1025             \def\bib'date{#1-#2}%
1026         \fi
1027     \else
1028         \def\bib'date{#1-#2-#3}%
1029     \fi
1030 }

```

## 6.12.6 Language setup

`\bib@language@fixup`

```

1031 \def\bib@language@fixup{%
1032   \ifx\bib'hyphenation\@empty
1033     \ifx\bib'language\@empty
1034       \let\bib@language\biblanguagedefault
1035     \else
1036       \let\bib@language\bib'language
1037     \fi
1038   \else
1039     \let\bib@language\bib'hyphenation
1040   \fi
1041   \def\@tempa##1 ##2\@nil{\lowercase{\def\bib@language{##1}}}%
    The mysterious \@firstofone here is to preserve the space before the \@nil.
1042   \@firstofone{\@xp\@tempa\bib@language} \@nil
1043 }
```

`\bib@selectlanguage` For `\bib` purposes we are interested mainly in testing whether the hyphenation patterns are the same. So we use an `if-same-patterns` test (by which `babel`'s 'english' and 'american' compare as equal) rather than an `if-same-language` test. Also, the way that the `\selectlanguage` command checks to see whether a language has been properly defined for `babel` use is to see if `\dateLANGUAGE` is defined. And if we tried to select an undefined language, the result would be a  $\LaTeX$  error.

```

1044 \def\bib@selectlanguage{%
1045   \@ifsame@patterns{\languagename}{\bib@language}{}{}%
1046   \@ifundefined{date\bib@language}{}{%
1047     \@xp\selectlanguage\@xp{\bib@language}%
1048   }%
1049 }%
1050 }
```

`\@ifsame@patterns`

```

1051 \def\@ifsame@patterns#1#2{%
1052   \@xp\@ifsamepat\csname l@#1\@xp\endcsname\csname l@#2\endcsname
1053 }
```

`\@ifsamepat`

```

1054 \def\@ifsamepat#1#2{%
1055   \ifnum \ifx\relax#1\m@ne\else#1\fi = \ifx\relax#2\m@ne\else#2\fi
1056     \@xp\@firstoftwo
1057   \else
1058     \@xp\@secondoftwo
1059   \fi
1060 }
```

`\languagename`

`\biblanguageEnglish`

`\biblanguagedefault`

`\bib@language`

```

1061 \providecommand{\languagename}{english}
1062 \def\biblanguagEnglish{english}
1063 \let\biblanguagedefault\biblanguagEnglish
1064 \let\bib@language\@empty

```

### 6.12.7 Citation label setup

```

\generate@label
1065 \let\generate@label\relax

\cite@label
1066 \def\cite@label{\@currentlabel}

\alpha@label
1067 \let\alpha@label\relax

\alpha@label@
1068 \def\alpha@label@{%
1069   \ifx\@empty\bib'label
1070     \def\thebib{\CurrentBib}%
1071   \else
1072     \let\thebib\bib'label
1073   \fi
1074 }%

\amsrefs@option@numeric
1075 \def\amsrefs@option@numeric{%
1076   \let\alpha@label\relax
1077   \let\generate@label\relax
1078   % \@nmbrlisttrue
1079 }

\amsrefs@option@alphabetic
1080 \def\amsrefs@option@alphabetic{%
1081   \let\alpha@label\alpha@label@
1082   \let\generate@label\generate@alphalabel
1083   \let\calc@author@part\calc@author@part@
1084   \let\@suffix@format\@alph
1085   \let\append@label@year\append@label@year@
1086   % \@nmbrlistfalse
1087 }

\amsrefs@option@shortalphabetic
1088 \def\amsrefs@option@shortalphabetic{%
1089   \let\alpha@label\alpha@label@
1090   \let\generate@label\generate@alphalabel
1091   \let\calc@author@part\calc@author@part@short
1092   \let\@suffix@format\@arabic
1093   \let\append@label@year\@empty
1094   % \@nmbrlistfalse
1095 }

```

`\bibcite` When `\bibcite` is called, author name and year are available in `\bibauthor` and `\bibyear`.

*Arguments:*

`#1` <- *citekey*.

```

1096 \def\bibcite#1{%
1097   \def\CurrentBib{#1}%
1098   \alpha@label          % modify \thebib if necessary
1099   \item\leavevmode
1100   \SK@SK@@label{#1}%
1101   \@xp\bibcite@{a\csname b@#1\endcsname
1102   \bibcite@write{#1}%
1103 }

1104 \def\bibcite@a#1{%
1105   \ifx\relax#1%
1106     \begingroup
1107     \auto@protect\etaltex
1108     \protected@edef\@tempa{%
1109       \gdef\@nx#1{%
1110         \@nx\citesel 01{\cite@label}{\bib@label@year}{}}%
1111       }%
1112     }%
1113   \@xp\endgroup
1114   \@tempa
1115 \else
1116   \@xp\bibcite@check\@xp#1#1\@empty\@empty\@empty\@empty\@empty
1117 \fi
1118 }

```

`\bibcite@check` For the citation key we want to check if it is already defined. But there is a slight problem. There is already one control sequence in use for each bibliography entry, to store the label or the author/year information needed by `\cite`. If we introduce another control sequence to check whether a particular cite is multiply defined, then we double the number of control sequences used. For a large bibliography in a book this is fairly serious. This is addressed by using a `\citesel` function.

*Arguments:*

`#1` <- *\b@citekey*.

`#2` <- *\citesel*.

`#3` <- *cited?*.

`#4` <- *used?*.

`#5` <- *label*.

`#6` <- *year*.

`#7` <- *backrefs*.

```

1119 \def\bibcite@check#1#2#3#4#5#6#7{%
1120   \ifx 1#4\relax
1121     \DuplicateBibKeyWarning

```

```

1122     \else
      This has gotten way out of hand.
1123     \begingroup
1124     \auto@protect\etaltext
1125     \@apply\auto@protect\amsrefs@textsymbols
1126     \@apply\auto@protect\amsrefs@textaccents
1127     \@tempswafalse
1128     \in\CitePrintUndefined{#5}%
1129     \ifin@
1130     \let\@tempa\@empty
1131     \else
1132     \def\@tempa{#5}%
1133     \fi
1134     \ifx\@tempa\@empty
1135     \else
1136     \exp\ifx\@xp\@currentlabel\cite@label
1137     \edef\@tempb{\cite@label}%
1138     \else
1139     \let\@tempb\cite@label
1140     \fi
1141     \ifx\@tempa\@tempb
1142     \def\@tempa{#6}%
1143     \ifx\@tempa\bib@label@year
1144     \else
1145     \@tempswatrue
1146     \fi
1147     \else
1148     \@tempswatrue
1149     \fi
1150     \fi
1151     \if@tempswa
1152     \@ifempty{#6}{%
1153     \def\@tempa{#5}%
1154     \let\@tempb\cite@label
1155     }{%
1156     \def\@tempa{#5, #6}%
1157     \def\@tempb{\cite@label, \bib@label@year}%
1158     }%
1159     \amsrefs@warning{Citation label for \extr@cite#1 is
1160     changing from '\@tempa ' to '\@tempb '}%
1161     \fi
1162     \protected@edef\@tempa{%
1163     \gdef\@nx#1{%
1164     \@nx\citesel #31{\cite@label}{\bib@label@year}{#7}%
1165     }%
1166     }%
1167     \@xp\endgroup
1168     \@tempa
1169     \fi

```

```
1170 }
```

```
\bib@label@year
```

```
1171 \let\bib@label@year\@empty
```

```
\DuplicateBibKeyWarning
```

```
1172 \def\DuplicateBibKeyWarning{%
1173   \amsrefs@warning{%
1174     Duplicate \protect\bib\space key
1175     ‘\CurrentBib ’ detected\MessageBreakNS}%
1176 }
```

```
\DuplicateBibKeyWarning
```

```
1177 \def\DuplicateBibLabelWarning{%
1178   \amsrefs@warning{%
1179     Duplicate biblabel stem ‘\current@stem ’ detected.\MessageBreakNS
1180     This usually means the order of the bibitems\MessageBreakNS
1181     is incompatible with the style of labels\MessageBreakNS
1182     you are using}%
1183 }
```

```
\bibcite@write
```

```
1184 \def\bibcite@write#1{%
1185   \if@filesw
1186     \begingroup
1187     \let\citesel\citesel@write
1188     \csname b@#1\endcsname
1189     \endgroup
1190   \fi
1191 }
```

```
\citesel@write
```

```
1192 \def\citesel@write#1#2#3#4#5{%
1193   \toks@{#3}{#4}}%
1194   \immediate\write\@auxout{\string\bibcite{\CurrentBib}{\the\toks@}}%
1195 }
```

Because duplicate bibs are caught immediately, we don't need `\bibcite` to run `\@testdef`.

```
1196 \AtEndDocument{\let\bibcite\@gobbletwo}
```

### 6.12.8 Printing the bibliography

```
\bibname
```

```
1197 \providecommand{\bibname}{Bibliography}
```

```
\refname
```

```
1198 \providecommand{\refname}{References}
```

`\bib@div@mark` The AMS document classes automatically take care of the page marks for `\section*` and `\chapter*`, but for the standard classes, we need to make sure that `\mkboth` gets invoked.

```
1199 \let\bib@div@mark@gobble
```

This is verbose, but probably safer than any alternative.

```
1200 \@ifclassloaded{amsbook}{}{}%
1201     \@ifclassloaded{amsart}{}{}%
1202         \@ifclassloaded{amsproc}{}{}%
1203             \def\bib@div@mark#1{%
1204                 \mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}%
1205             }%
1206         }%
1207     }%
1208 }
```

`bibchapter` We need to take a little extra trouble here to pre-expand the `\bibname`.

```
1209 \newenvironment{bibchapter}[1][\bibname]{%
1210     \begingroup
1211         \protected@edef\@{%
1212             \endgroup
1213             \protect\chapter*{#1}%
1214             \protect\bib@div@mark{#1}%
1215         }%
1216     \@
1217 }\par}
```

`bibsection` And here to pre-expand the `\refname`.

```
1218 \newenvironment{bibsection}[1][\refname]{%
1219     \begingroup
1220         \protected@edef\@{%
1221             \endgroup
1222             \ifx\@bibtittlestyle\undefined
1223                 \protect\section*{#1}%
1224             \else
1225                 \protect\@bibtittlestyle
1226             \fi
1227             \protect\bib@div@mark{#1}%
1228         }%
1229     \@
1230 }\par}
```

`bibdiv` Here we try to guess whether this is a book-like document or an article-like document.

```
1231 \@ifundefined{chapter}{%
1232     \newenvironment{bibdiv}{\bibsection}{\endbibsection}
1233 }{%
1234     \newenvironment{bibdiv}{\bibchapter}{\endbibchapter}
1235 }
```



This is what the standard book class has for the bibliography title:

```
\newenvironment{thebibliography}[1]
  {\chapter*{\bibname
    \@mkboth{\MakeUppercase\bibname}{\MakeUppercase\bibname}}%
    \list{\@biblabel{\@arabic\c@enumiv}}%
```

thebibliography

```
1236 \renewenvironment{thebibliography}[1]{%
1237   \bibdiv
1238   \biblist[\resetbiblist{#1}]%
1239 }{%
1240   \endbiblist
1241   \endbibdiv
1242 }
```

### 6.13 Name, journal and publisher abbreviations

The commands `\DefineName`, `\DefinePublisher`, and `\DefineJournal` are provided to make abbreviations a little easier.

`\DefineName`

```
1243 \newcommand{\DefineName}[2]{%
1244   \bib*{#1}{name}{name={#2}}%
1245 }
```

`\DefineJournal`

```
1246 \newcommand{\DefineJournal}[4]{%
1247   \bib*{#1}{periodical}{
1248     issn={#2},
1249     journal={#4}
1250   }%
1251 }
```

`\DefinePublisher` Note that an explicit address field in a `\bib` entry will override the address supplied as part of a `\DefinePublisher`.

```
1252 \newcommand{\DefinePublisher}[4]{%
1253   \bib*{#1}{publisher}{%
1254     publisher={#3},
1255     address={#4}
1256   }%
1257 }
```

### 6.14 Processing .ltx files

If you have a file that contains `amsrefs`-style `\bib` entries, you can use it as a database and extract items from it for use in another document. In typical relatively simple scenarios, the extraction can be done by  $\LaTeX$  itself on the first pass, so that citations in the text will be successfully resolved on the second pass (possibly even the first, depending on what kind of bibliography sorting is used).

`\bibselect`

```

1258 \newcommand{\bibselect}{%
1259     \@ifstar{%
1260         \let\@bibdef\copy@bibdef
1261         \BibSelect
1262     }{%
1263         \let\@bibdef\selective@bibdef
1264         \BibSelect
1265     }%
1266 }

```

`\BibSelect`

```

1267 \newcommand{\BibSelect}[2][\bblname]{%
1268     \if@filesw
1269         \typeout{Trying to create bbl file '#1.bbl' ...}%
1270         \def\bibselect@msg{%
1271             \typeout{ ... rats. Unable to create bbl file.}%
1272         }%
1273         \let\@open@bbl@file\OpenBBLFile
1274         \@for\@tempa:=#2\do{\ReadBibData{\@tempa}}%
1275     \fi
1276     \@close@bbl@file
1277     \@apply@g@undef\bibdefer@list
1278     \global\let\bibdefer@list\@empty

```

Now read the .bbl file we just created.

```

1279     \let\@bibdef\normal@bibdef
1280     \@input@{#1.bbl}%
1281     \let\BibSelect\MultipleBibSelectWarning
1282 }

```

`\MultipleBibSelectWarning`

```

1283 \newcommand\MultipleBibSelectWarning[2][ ]{%
1284     \amsrefs@warning{%
1285         Multiple \string\bibselect 's found (only one
1286         \string\bibselect\space per biblist environment is allowed)%
1287     }%
1288 }

```

`\bblname`

```

1289 \def\bblname{\jobname}

```

`\bib@dbfile`

```

1290 \newread\bib@dbfile

```

`\ReadBibData`

```

1291 \newcommand{\ReadBibData}[1]{%
1292     \IfFileExists{#1.ltb}{%
1293         \openin\bib@dbfile=\@filef@und \relax

```

```

1294 }{%
1295   \IfFileExists{#1.ltx}{%
1296     \openin\bib@dbfile=@filef@und \relax
1297   }{%
1298     \IfFileExists{#1.tex}{%
1299       \openin\bib@dbfile=@filef@und \relax
1300     }{%
1301       \begingroup
1302         \NoBibDBFile{#1}%
1303         \let\ReadBibData@a\endgroup
1304       }%
1305     }%
1306   }%
1307   \ReadBibData@a
1308 }

```

\NoBibDBFile

```

1309 \def\NoBibDBFile#1{%
1310   \amsrefs@warning{No data file #1.ltb (.ltx, .tex) found}%
1311 }

```

\ReadBibData@a

```

1312 \def\ReadBibData@a{%
1313   \ProvidesFile{@filef@und}\relax
1314   \begingroup
1315     \let\star@bibdef\defer@bibdef
1316     \ReadBibLoop
1317   \endgroup
1318   \closein\bib@dbfile
1319 }

```

\ReadBibLoop

```

1320 \def\ReadBibLoop{%
1321   \ifeof\bib@dbfile
1322     \@xp@gobble
1323   \else
1324     \read\bib@dbfile to\CurLine
1325     The \@empty is in case \CurLine is empty.
1326     \@xp\ReadBibLoop@a\CurLine\@empty\@nil
1327   \fi
1328 }

```

\ReadBibLoop@e This traps top-level \bib commands. Note that:

- If \CurLine doesn't contain a complete \bib entry, the code chokes.
- If \bib is not the very first non-space token in a line, it will not be recognized.

```

1329 \long\def\ReadBibLoop@a#1#2\@nil{%
1330   \ifx\bib#1%
1331     \CurLine % just exec it
1332   \else

```

We're not done yet. The line may contain something like `\DefineName`, so we need to expand the first macro in the line and see if it starts with `\bib`. But first we check to make sure that the token we're about to expand isn't `\endinput`.

```

1333     \ifx\endinput#1%
1334       \let\ReadBibLoop\@empty
1335     \else

```

And this `\@empty` is for the admittedly unlikely case that `\CurLine` isn't empty, but its expansion is.

```

1336       \@xp\ReadBibLoop@b#1#2\@empty\@nil
1337     \fi
1338   \fi
1339 }

```

`\ReadBibLoop@b`

```

1340 \long\def\ReadBibLoop@b#1#2\@nil{%
1341   \ifx\bib#1%
1342     \CurLine % just exec it
1343   \fi
1344 }

1345 \let\bbl@out=\relax
1346 \let\bbl@write\@gobble
1347 \let\@open@bbl@file\relax
1348 \let\@close@bbl@file\relax

```

`\OpenBBLFile`

```

1349 \def\OpenBBLFile{%
1350   \if@filesw
1351     % Just use the next unused output stream
1352     \count@\count17
1353     \advance\count@\@ne
1354     \ifnum\count@<\sist@\@n
1355       \global\chardef\bbl@out=\count@
1356       \immediate\openout\bbl@out=\bblname.bbl\relax
1357       \global\let\@close@bbl@file\CloseBBLFile
1358       \gdef\bbl@write{\immediate\write\bbl@out}%
1359     \else
1360       \ch@ck\count@\sist@\@n\write
1361     \fi
1362   \fi
1363   \global\let\@open@bbl@file\relax
1364 }

```

`\CloseBBLFile`

```

1365 \def\CloseBBLFile{%
1366   \immediate\closeout\bbl@out\relax
1367   \global\let\@close@bbl@file\relax
1368   \global\let\bbl@write@gobble
1369   \global\let\bbl@out\relax
1370 }

```

## 6.15 Citation processing

### 6.15.1 The `\citesel` structure

The information used by `\cite` for key moo is stored in `\b@moo` in the form

```
\citesel{status1}{status2}{label}{year}{backref-info}
```

The first status flag is 1 if this key has already been cited earlier in the same document; 0 otherwise. This is used in some bibliography schemes to print a full list of author names for the first citation and an abbreviated author list for subsequent citations.

The second status flag is 1 if this key has already been used by a define-cite command (such as `\bib`); 0 otherwise. This makes it possible to issue a warning message as soon as the conflict is seen, on the first L<sup>A</sup>T<sub>E</sub>X run, instead of on a subsequent run during the processing of the `.aux` file.

When an author/year citation scheme is in use, args 3 and 4 hold respectively author names and year. Otherwise arg 3 simply holds a cite label and arg 4 is empty.

And finally, arg 5 holds a list of backref pointers indicating the locations in the document where this entry has been cited.

```

\citesel@update
1371 \def\citesel@update#1#2#3#4#5#6{%
1372   \gdef#6{\citesel 1#2{#3}{#4}{#5}}%
1373 }

\citesel@number
1374 \def\citesel@number#1#2#3#4#5{#3}

\citesel@year
1375 \def\citesel@year#1#2#3#4#5{#4}

\citesel
1376 \let\citesel\citesel@number

```

### 6.15.2 The basic `\cite` command

Here is the difference between the various optional forms of `\cite`:

```

\cite{xyz}      -> \cite@a\citesel{xyz}{ }
                -> \cite@bc\b@xyz\citesel{ }

\cite{xyz}*{blub} -> \cite@a\citesel{xyz}{blub}
                -> \cite@bc\b@xyz\citesel{blub}

```

```

\cite[blub]{xyz} -> \cite@a\citesel{xyz}{blub}
                -> \cite@bc\b@xyz\citesel{blub}

```

Canceling the old L<sup>A</sup>T<sub>E</sub>X definition of `\citel` prevents certain problems that could arise with the `showkeys` package.

```
1377 \expandafter\let\csname cite \endcsname\relax
```

`\cite` Need to handle the standard [...] option for compatibility's sake.

```

1378 \renewcommand{\cite}[2] [] {%
1379   \if\cite@single#2,\@gobble \else\MultipleCiteKeyWarning{#2}{#1}\fi
1380   \@ifempty{#1}{%
1381     \cites@o{#2}%
1382   }{%
1383     \ObsoleteCiteOptionWarning
1384     \cites@a{*{#1}}{#2}%
1385   }%
1386 }

```

`\MultipleCiteKeyWarning`

```

1387 \def\MultipleCiteKeyWarning#1#2{%
1388   \amsrefs@warning{%
1389     Use of \string\cites\space is recommended instead of %
1390     \string\cite\space\MessageBreak
1391     for multiple cites '#1'%
1392   \@ifnotempty{#2}{%
1393     \amsrefs@warning{Star option requires \string\citelist\space here}%
1394   }%
1395   \global\let\MultipleCiteKeyWarning\@gobbletwo
1396 }

```

`\ObsoleteCiteOptionWarning`

```

1397 \def\ObsoleteCiteOptionWarning{%
1398   \amsrefs@warning{%
1399     The form \string\cite{...}*{...} is recommended\MessageBreak
1400     instead of \string\cite[...]{...}%
1401   \global\let\ObsoleteCiteOptionWarning\@empty
1402 }

```

`\cite@single`

```
1403 \edef\cite@single#1,#2{\iffalse{\fi\iffalse{\fi\string}#2.\string}}
```

`\cites@o`

```
1404 \def\cites@o#1{\star@{\cites@oo{#1}}{}}
```

`\cites@oo`

```
1405 \def\cites@oo#1#2{\@ifempty{#2}{\cites@a-}{#1}}{\cites@a*{#2}}{#1}}
```

```

\cites@a
1406 \def\cites@a#1#2{%
1407     \begingroup
1408         \toks@{\endgroup \cites@b{#1}}%
1409         \vdef\@tempa{#2}%
1410         \edef\@tempa{%
1411             \the\toks@ \@firstofone{\@xp\zap@space\@tempa} \@empty
1412         }%
1413         \@tempa,\@empty
1414         \edef\@tempa{\endgroup\@nx\citelist{\the\toks@}}%
1415         \@tempa
1416 }

```

```

\cites@b
1417 \def\cites@b#1#2,#3{%
1418     \begingroup
1419         \toks@{\InnerCite{#2}#1}%
1420         \ifx\@empty#3\@xp\@gobble\fi
1421         \cites@c#3%
1422 }

```

```

\cites@c
1423 \def\cites@c#1,#2{%
1424     \add@toks@{\InnerCite{#1}}%
1425     \ifx\@empty#2\@xp\@gobble\fi
1426     \cites@c#2%
1427 }

```

`\citeleft` These variables are named to follow the precedent set by Arseneau’s `cite` package. `\citeright` age. `\citemid` is used to separate a citation label from additional information such as “Theorem 4.9”. `\citepunct` is used to separate multiple cites, unless one of the cites has additional associated information, in which case `\CiteAltPunct` is used.

```

1428 \def\citeleft{[]}
1429 \def\citeright{[]}
1430 \def\citemid{,\penalty9999 \space}
1431 \def\citepunct{,\penalty9999 \hskip.13em plus.1em minus.05em\relax}

```

`\citeAltPunct` When a citation list contains one or more citations with optional arguments, we replace `\citemid` by `\CiteAltPunct`.

```

1432 \def\citeAltPunct{; \ }

```

`\citeform` This is used for formatting the citation label. It can be used, for example, to bolden the labels (as in `amsbook` and `amsproc`) or to do more elaborate things such as convert the numbers to roman numerals. By default, it’s just a no-op.

Note that currently there is no corresponding macro for changing the formatting of `\cite`’s optional argument. This is probably a bug.

```

1433 \providecommand{\citeform}{\@firstofone}

```

`\citetlist` The `\citetlist` indirection turns out to be helpful in implementing the `\ocites` command for the author-year option.

```

1434 \DeclareRobustCommand{\citetlist}{\@citetlist}

\@citetlist
1435 \def\@citetlist#1{%
1436     \leavevmode
1437     \begingroup
1438         \@citetstyle
1439         \citeleft\nopunct    % suppress first \citepunct
1440         \cite@begingroup
1441         \in@*{#1}%
1442         \ifin@
1443             \let\citepunct\citeAltPunct
1444             \fi
1445             \let\cite@endgroup\@empty
1446             \cites@init
1447             \def\citeleft{\@addpunct{\citepunct}}%
1448             \let\citeright\ignorespaces
1449             \def\cite{\InnerCite}%
1450             \process@citetlist{#1}%
1451         \endgroup
1452     \citeright
1453 \endgroup
1454 }
```

`\@citetstyle` Reset the font to an upright, medium font (e.g. `cmr`), per AMS style. Also set `\mathsurround = 0pt` just in case there are subscripts in the cite numbers (from `\etalchar`, for example).

```
1455 \providecommand{\@citetstyle}{\m@th\upshape\mdseries}
```

`\cite@begingroup` Grouping that encloses an entire cite block (a single cite or a list of cites).

```
1456 \def\cite@begingroup{\begingroup\let\cite@begingroup\relax}
```

`\cite@endgroup`

```
1457 \let\cite@endgroup\endgroup
```

`\cites@init` This needs to be called at the beginning of a list of cites to reset a few things.

```

1458 \def\cites@init{%
1459     \gdef\prev@names{???}%
1460     \let\cites@init\@empty
1461 }
```

`\InnerCite`

```
1462 \newcommand{\InnerCite}[1]{\star@\cite@a\citesel{#1}}{}}
```

`\cite@a` The job of `\cite@a` is to convert the cite key to all catcode-12 characters and remove any spaces it might contain before passing it on to `\cite@b`.



*Arguments:*

```
#1 <- \CITESEL.
#2 <- citekey.
```

```
1463 \def\cite@a#1#2{%
1464   \BackCite{#2}%
1465   \cite@begingroup
1466     \cites@init
1467     \let\citesel#1\relax
1468     \ifx\citesel\citesel@author
1469       \let\citeleft\@empty
1470       \let\citeright\@empty
1471     \fi
1472     \begingroup
1473     \toks@{\endgroup \cite@b}%
1474     \vdef\@tempa{#2}%
1475     \edef\@tempa{%
1476       \the\toks@{\@firstofone{\@xp\zap@space\@tempa} \@empty}%
1477     }%
1478     \@tempa
1479 }
```

`\cite@b` *Arguments:*

```
#1 <- citekey.
#2 <- star-optional-arg.
```

```
1480 \def\cite@b#1#2{%
1481   \@xp\cite@bc\csname b@#1\@xp\endcsname {#1}{#2}%
1482 }
```

`\cite@bc` If it's uninitialized, plug in an empty cite structure. `\cite@bc` should be executed only once for a given instance of a cite key. All further processing should go through `\cite@cj`.

```
1483 \def\cite@bc#1#2{%
1484   \ifx#1\@undefined \global\let#1\relax \fi
1485   \ifx#1\relax \global\let#1\empty@cite \fi
1486   \@xp\cite@nobib@test#1{}{}{} \@nil#1%
1487   \cite@cj#1%
1488 }
```

`\empty@cite`

```
1489 \def\empty@cite{\citesel 00{}{}{}}
```

`\cite@nobib@test` If arg 4 is empty, it means there wasn't any `\bib` command that defined a valid label.

*Arguments:*

```
#1 <- \citesel.
#2 <- cited?.
#3 <- used?.
```

```

#4 <- label.
#5 <- backrefs.
#6 <- \b@citekey.

```

```

1490 \def\cite@nobib@test#1#2#3#4#5\@nil#6{%
1491   \@ifempty{#4}{%
1492     \G@refundefinedtrue
1493     \UndefinedCiteWarning#6%
1494     \xdef#6{\@nx\citesel #2#3{%
1495       \@nx\CitePrintUndefined{\extr@cite#6}}{}}}%
1496   }{}%
1497 }

```

`\UndefinedCiteWarning` This is a copy of the standard warning from `\@citex`.

```

1498 \def\UndefinedCiteWarning#1{%
1499   \@latex@warning{%
1500     Citation ‘\extr@cite#1’ on page \thepage\space undefined}%
1501 }

```

`\CitePrintUndefined`

```

1502 \DeclareRobustCommand{\CitePrintUndefined}[1]{%
1503   \begingroup\fontshape{n}\fontseries\mddefault \ttfamily ?#1\endgroup
1504 }

```

`\CPU@normal` This has to be a `\let`, not a `\def`.

```

1505 \let\CPU@normal\CitePrintUndefined

```

`\cite@cj` *Arguments:*

```

#1 <- \b@citekey.
#2 <- star-optional-arg.

```

```

1506 \def\cite@cj#1#2{%
1507   \leavevmode
1508   \begingroup
1509     \cite@cb#1% write info to aux file
1510     \ar@SK@cite#1%
1511     \@citeleft
1512     \ar@hyperlink{#1}%
1513     \@ifnotempty{#2}{\citimid{#2}}%
1514     \citeright
1515   \endgroup
1516   \ignorespaces % ignore spaces inside \citelist
1517   \cite@endgroup
1518 }

```

`\@citeleft` The following definition provides some indirection that helps to deal with author-year object cites.

```

1519 \def\@citeleft{\citeleft}

```

`\cite@cb`

```

1520 \def\cite@cb#1{%
1521   \if@filesw
1522     \immediate\write\@auxout{\string\citation{\extr@cite#1}}%
1523   \fi
  Define \citesel to make \b@whatever update itself.
1524   \begingroup
1525     \let\citesel\citesel@update
1526     #1#1%
1527   \endgroup
1528 }
```

`\extr@cite` Extract *citekey* from `\b@citekey`.

```
1529 \def\extr@cite{\@xp\@gobblethree\string}
```

### 6.15.3 Fancier `\cite` commands

`\cites` A list of simple cites. Make it robust in case used inside a figure caption. (But then also, by the way, `listoffigures` should provide special handling.)

```
1530 \DeclareRobustCommand{\cites}{\cites@a{}}
```

`\citen` This is just to keep the `showkeys` package from clobbering the wrong part of our definition of `\cite`:

```
1531 \providecommand{\citen}{\ocite}
```

`\ycite` `\cite` gets redefined inside of `\citelist`, so we need to `\def \ycite` here instead of just `\letting` everything to `\cite`.

```
1532 \def\ycite{\cite}
```

`\ycites`

```
1533 \let\ycites\cites
```

`\ocite`

```
1534 \let\ocite\ycite
```

`\ocites`

```
1535 \let\ocites\cites
```

`\fullcite`

```
1536 \let\fullcite\cite
```

`\fullocite`

```
1537 \let\fullocite\ocite
```

`\citeauthor`

```
1538 \let\citeauthor\ycite
```

`\citeauthority`

```
1539 \let\citeauthority\ycite
```

6.15.4 The `\nocite` command`\nocite`

```
1540 \renewcommand{\nocite}[1]{\othercites{#1}}
```

`\othercites`

```
1541 \newcommand{\othercites}[1]{%
1542   \cite@begingroup
1543     \let\BackCite@gobble
1544     \let\cite@endgroup@empty
1545     \def\citelist{\othercitelist}%
1546     \cites{#1}%
1547 }
```

`\othercitelist`

```
1548 \newcommand{\othercitelist}[1]{%
1549   \cite@begingroup
1550     \let\cite@endgroup@empty
1551     \cites@init
1552     \let\citeleft\relax
1553     \let\citeright\ignorespaces
1554     \def\InnerCite{\OtherCite}%
1555     \def\cite@cj ##1##2{%
1556       \begingroup
1557         \@xp\citesel##1%
1558         \cite@cb ##1%
1559       \endgroup
```

If we detect `\nocite{*}`, we globally alias `\selective@bibdef` to `\copy@bibdef` so that all succeeding `\bibselect` commands act like `\bibselect*`.

```
1560     \@xp\ifx\csname b@*\endcsname ##1%
1561     \global\let\selective@bibdef\copy@bibdef
1562     \fi
1563     \ignorespaces
1564     \cite@endgroup
1565   }%
1566   #1\relax
1567 \endgroup
1568 }
```

`\OtherCite`

```
1569 \def\OtherCite#1{\cite@a\citesel@other{#1}-{}}
```

`\citesel@other`

```
1570 \def\citesel@other#1#2#3#4#5#6{}
```

`\b@*` This provides a dummy definition to keep things like `\nocite{*}` from generating an error message.

```
1571 \@namedef{b@*}{\citesel 11{*}{*}{*}}
```

### 6.15.5 Citation sorting

`\process@citelist@sorted`

```
1572 \def\process@citelist@sorted#1{%
1573   \ifx\citesel\citesel@number
1574     \cite@sorted@s #1\cite@sorted@e
1575   \else
1576     \NonNumericCiteWarning
1577     \process@citelist@unsorted{#1}%
1578   \fi
1579 }
```

`\NonNumericCiteWarning`

```
1580 \def\NonNumericCiteWarning{%
1581   \amsrefs@warning{%
1582     Unable to confirm that cite keys are numeric: not sorting%
1583   }%
1584 }
```

`\process@citelist@unsorted`

```
1585 \def\process@citelist@unsorted#1{%
1586   \ignorespaces#1\relax
1587 }
```

`\process@citelist` By default, citation lists will be sorted.

```
1588 \let\process@citelist\process@citelist@sorted
```

`\CPU@sort` By defining this as T<sub>E</sub>X's maxint, undefined cites migrate to the end of a sorted list.

```
1589 \def\CPU@sort#1{2147483647}
```

`\cite@sorted@s` Here's where we prepare to sort the citations and (optionally) compress ranges.

```
1590 \def\cite@sorted@s{%
1591   \begingroup
1592     \let\CitePrintUndefined\CPU@sort
1593     \let\cite@cjs\cite@cj
1594     \let\cite@cj\cite@compress
1595   \begingroup
1596     \toks@\@emptytoks
1597     \let\cite@cj\cite@sort
1598     \ignorespaces
1599 }
```

`\cite@sorted@e`

```
1600 \def\cite@sorted@e{%
1601   \@xp\endgroup
1602   \the\toks@
1603   \cite@dash
1604   \prev@cite
1605 \endgroup
1606 }
```

`\cite@sort` This is essentially an insertion sort. I think.

*Arguments:*

`#1` <- `\b@citekey`.  
`#2` <- *optional arg*.

```
1607 \def\cite@sort#1#2{%
1608   \safe@set\@tempcnta#1% highest number so far
1609   \toks@\cite@cj#1{#2}}%
1610   \temptokena\toks@
1611   \let\cite@cj\cite@sort@a
1612   \ignorespaces
1613 }
```

`\cite@sort@a`

```
1614 \def\cite@sort@a#1#2{%
1615   \safe@set\@tempcntb#1%
1616   \ifnum\@tempcntb > \@tempcnta
1617     \add@toks@\cite@cj#1{#2}}%
1618     \@tempcnta\@tempcntb
1619   \else
1620     \let\cite@cj\cite@sort@b
1621     \toks@\@emptytoks
1622     \def\@tempb{\add@toks@\cite@cj#1{#2}}%
1623     \the\@temptokena
1624     \@tempb
1625     \let\cite@cj\cite@sort@a
1626   \fi
1627   \temptokena\toks@
1628   \ignorespaces
1629 }
```

`\cite@sort@b`

```
1630 \def\cite@sort@b#1#2{%
1631   \safe@set\count@#1%
1632   \ifnum\@tempcntb < \count@
1633     \@tempb
1634     \let\@tempb\@empty
1635   \fi
1636   \add@toks@\cite@cj#1{#2}}%
1637   \ignorespaces
1638 }
```

### 6.15.6 Range compression

When the time comes to apply compression, we have at our disposal a list of internal cite calls that looks like this:

```
\cite@cj\b@aaa{opta}\cite@cj\b@bbb{optb}... \cite@cj\b@zzz{optz}
```

where

$$\b@aaa < \b@bbb < \dots < \b@zzz$$

and the `opt` arguments are possibly null. To print the citations while collapsing sequences of 3 or more contiguous numbers into ranges of the form  $n$ – $m$ , we bind `\cite@cj` to a suitably clever function and then execute the list. In the absence of optional arguments, here's the algorithm:

- Begin. Enter state 0. This is done by `\cite@sorted@s`.
- State 0. The current citation is the beginning of a range (possibly a singleton range). Print it. Then, set  $prevnum := number$  and enter state 1.
- State 1. The current citation might be the second element of a range.
- Case a)  $number = prevnum + 1$ . Then the current item is definitely the second element of a range. It might be the last element of the range, but we won't know until we examine the following citation. So, save the current citation in `\prev@cite`, set  $prevnum := number$ , and go to state 2.
  - Case b)  $number \neq prevnum + 1$ . The current citation is the beginning of a new range. Print it, set  $prevnum := number$  and remain in state 1. (This is essentially identical to stage 0.)
- State 2. The current citation might be the third (or later) element of a range.
- Case a)  $number = prevnum + 1$ . The current element is definitely part of a range. It might be the last element of the range, but again we won't know until we examine the following citation. Save the current citation in `\prev@cite` and set  $prevnum := number$ . Remain in state 2.
  - Case b)  $number \neq prevnum + 1$ . The previous citation was the end of a range and the current citation is the beginning of a new range. Print a dash followed by `\prev@cite`, then set  $prevnum := number$  and enter state 1.
- End. If `\prev@cite` is not empty, print it, preceded by a dash if we were in the middle of a range. (This is done by `\cite@sorted@e`.)

The presence of optional arguments complicates things somewhat, since a citation with an optional argument should never participate in range compression. In other words, when we come across an optional argument, we should finish off the preceding range, print the current citation, and then return to the initial state. More precisely, here are the actions taken in each state when there is an optional argument:

- State 0. Print the current citation and remain in state 0.
- State 1. Print the current citation and return to state 0.
- State 2. Print a dash followed by `\prev@cite`. Then print the current citation and return to state 0.

`\prev@cite`

1639 `\let\prev@cite\@empty`

`\prev@cite@cb` There's one further complication: Even though we're suppressing some of the citation numbers, we need to make sure that each citation is recorded in the `.aux` file. So, in case 2a, before we overwrite `\prev@cite`, we first invoke `\prev@cite@cb` to record the previous citation (if any).

```

1640 \def\prev@cite@cb{%
1641     \ifx\@prev@cite\@empty
1642     \else
1643         \begingroup
1644             \def\cite@print##1##2{%
1645                 \cite@cb##1%
1646             }%
1647             \prev@cite
1648         \endgroup
1649     \fi
1650 }

```

`\cite@print`

```

1651 \def\cite@print#1#2{%
1652     \begingroup
1653         \let\CitePrintUndefined\CPU@normal
1654         \cite@cjs#1{#2}%
1655     \endgroup
1656 }

```

`\cite@dash` Ok, I lied. There was more than one further complication. Suppose that when we hit the end of the list, we're in state 2. We need to know whether to output a dash or a comma. (For example, both the sequences [2, 3] and [1, 2, 3] will end in state 2 with `prevcite = 3`, but in the former case we want a comma before the 3 and in the latter case we want a dash.) So, rather than printing the dash explicitly, we use `\cite@dash` to keep track of whether a dash is needed.

```

1657 \let\cite@dash\@empty

```

`\print@one@dash`

```

1658 \def\print@one@dash{%
1659     \textendash \nopunct
1660     \let\cite@dash\@empty
1661 }

```

State 0, 1 and 2 each correspond to a different binding for `\cite@cj`. Here they are. The role of `prevnum` is played by `\@tempcnta`, with `\@tempcntb` assisting as *number* at times.

`\cite@compress` State 0:

```

1662 \def\cite@compress#1#2{%
1663     \cite@print#1{#2}%
1664     \@ifempty{#2}{%
1665         \safe@set\@tempcnta#1%
1666         \let\cite@cj\cite@compress@a

```



```
1667   }{)%
1668 }
```

`\cite@compress@a` State 1:

```
1669 \def\cite@compress@a#1#2{%
1670   \@ifempty{#2}{%
1671     \advance\@tempcnta\@ne
1672     \safe@set\@tempcntb#1%
1673     \ifnum\@tempcnta=\@tempcntb
1674       \def\prev@cite{\cite@print#1{}}%
1675       \let\cite@cj\cite@compress@b
1676     \else
1677       \cite@print#1{)%
1678       \@tempcnta\@tempcntb
1679     \fi
1680   }{)%
1681   \cite@print#1{#2}%
1682   \let\cite@cj\cite@compress
1683   }%
1684 }
```

`\cite@compress@b` State 2:

```
1685 \def\cite@compress@b#1#2{%
1686   \@ifempty{#2}{%
1687     \advance\@tempcnta\@ne
1688     \safe@set\@tempcntb#1%
1689     \ifnum\@tempcnta=\@tempcntb
1690       \let\cite@dash\print@one@dash
1691       \prev@cite@cb
1692       \def\prev@cite{\cite@print#1{}}%
1693     \else
1694       \cite@dash
1695       \prev@cite
1696       \let\prev@cite\@empty
1697       \cite@print#1{)%
1698       \@tempcnta\@tempcntb
1699       \let\cite@cj\cite@compress@a
1700     \fi
1701   }{)%
1702   \cite@dash
1703   \prev@cite
1704   \let\prev@cite\@empty
1705   \cite@print#1{#2}%
1706   \let\cite@cj\cite@compress
1707   }%
1708 }
```

### 6.15.7 Munging the .aux file

`\amsrefs@bibcite` When processing the .aux file at begin-document, this is what `\bibcite` will do:

```
1709 \def\amsrefs@bibcite#1{\@xp\bibcite@a\csname b@#1\endcsname}
```

However, `hyperref` also redefines `\bibcite`, so to avoid conflicts and also ensure that it doesn't matter whether `amsrefs` or `hyperref` is loaded first, rather than redefining `\bibcite` directly, we redefine it inside the `.aux` file.

```
1710 \AtBeginDocument{%
1711   \if@filesw
1712     \immediate\write\@auxout{%
1713       \string\@ifundefined{amsrefs@bibcite}{}{%
1714         \string\let\string\bibcite\string\amsrefs@bibcite
1715       }%
1716     }%
1717   \fi
```

For good measure, we'll redefine it here as well, even though it really shouldn't matter any longer.

```
1718   \let\bibcite\amsrefs@bibcite
1719 }
```

`\bibcite@a` *Arguments:*

```
#1 <- \b@citekey.
#2 <- {label}{ } or {author}{year}.
```

```
1720 \def\bibcite@a#1#2{%
```

Most of the time arg 1 will already be defined, by an earlier `\citedest` command in the `.aux` file. Then we just need to change the number.

```
1721   \ifx\relax#1%
1722     \gdef#1{\citesel 00#2{}}%
1723   \else
1724     \begingroup
1725       \@xp\bibcite@b\@xp#1#1{#2}%
1726     \endgroup
1727   \fi
1728 }
```

`\bibcite@b` *Arguments:*

```
#1 <- \b@citekey.
#2 <- \citesel.
#3 <- cited?.
#4 <- used?.
#5 <- label.
#6 <- year.
#7 <- backrefs.
#8 <- {newlabel}{newyear}.
```

```
1729 \def\bibcite@b#1#2#3#4#5#6#7#8{\gdef#1{\citesel#3#4#8{#7}}}
```

`\citedest` The `\citedest` command goes into the `.aux` file to provide back-reference support.

```
1730 \newcommand{\citedest}[1]{\@xp\cite@dest\csname b@#1\endcsname}
```

`\cite@dest`

```
1731 \def\cite@dest#1{%
1732   \ifx\relax#1%
1733     \gdef#1{\citesel 00{}{}{}}%
1734   \fi
1735   \@xp\cite@dest@b\@xp#1#1%
1736 }
```

`\cite@dest@b` *Arguments:*

```
#1 <- \b@citekey.
#2 <- \citesel.
#3 <- cited?.
#4 <- used?.
#5 <- label.
#6 <- year.
#7 <- backrefs.
#8 <- {more backrefs}.
```

```
1737 \def\cite@dest@b#1#2#3#4#5#6#7#8{%
1738   \@ifempty{#7}{%
1739     \def#1{\citesel #3#4{#5}{#6}{#8}}%
1740   }{%
1741     \gdef#1{\citesel #3#4{#5}{#6}{#7,#8}}%
1742   }%
1743 }
```

### 6.15.8 Back references

`\ifBR@verbose`

```
1744 \@ifundefined{ifBR@verbose}{\let\ifBR@verbose\iffalse \let\fi\fi}{%}
```

`\BackCite`

```
1745 \let\BackCite\@gobble
```

`\back@cite`

```
1746 \def\back@cite#1{%
1747   \ifBR@verbose
1748     \PackageInfo{backref}{back cite \string '\extr@cite#1'}%
1749   \fi
1750   \Hy@backout{#1}%
1751 }
```

`\print@backrefs` In an AMS-style bibliography, the backref info might follow the final period of the reference, or it might follow some *Mathematical Reviews* info, without a period.

```

1752 \def\print@backrefs#1{%
1753   \space\SentenceSpace$\uparrow$\csname br@#1\endcsname
1754 }

```

\PrintBackRefs

```
1755 \let\PrintBackRefs@gobble
```

### 6.15.9 hyperref, showkeys and shaderef support

\shade@cite

```
1756 \newcommand{\shade@cite}{\printref}
```

\format@cite

```
1757 \def\format@cite#1{\shade@cite{\citeform{#1}}}
```

\ar@hyperlink

```

1758 \def\ar@hyperlink#1{%
1759   \hyper@link[cite]{}{cite.\extr@cite#1}{\format@cite{#1}}%
1760 }

```

\ar@SK@cite

```
1761 \def\ar@SK@cite#1{\@bsphack\@xp\SK@\@xp\SK@@ref\@xp{\extr@cite#1}\@espack}
```

Turn off hyperref and showkeys support if those packages don't appear to be loaded.

```

1762 \AtBeginDocument{%
1763   \ifpackageloaded{shaderef}{}{%
1764     \let\shade@cite\@firstofone
1765   }%
1766   \ifpackageloaded{hyperref}{}{%
1767     \def\ar@hyperlink{\format@cite}%
1768     \let\hyper@anchorstart@gobble
1769     \let\hyper@anchorend\relax
1770     \let\Hy@raisedlink\@firstofone
1771   }%
1772   \ifpackageloaded{showkeys}{}%
1773     \ifpackagewith{showkeys}{notcite}{}%
1774     \let\ar@SK@cite@gobble
1775   }{}
1776 }{}%
1777 \let\ar@SK@cite@gobble
1778 \let\SK@@label@gobble
1779 \let\SK@\@gobbletwo
1780 }%
1781 }

```

## 6.16 Lexical structure of names

Before we can begin parsing names, we need to give some thought to the lexical structure of names. For the remainder of this document, when we refer to a “name” and especially when we speak of a name as a macro argument, we assume that the only tokens contained in the name are

- letters and punctuation (i.e., characters with catcode 11 or 12),
- ties (the token  $\sim_{13}$ ),
- accent commands, such as  $\backslash$ " or  $\backslash$ k,
- text symbol macros, such as  $\backslash$ i,  $\backslash$ ae or  $\backslash$ cprime,
- grouping characters (braces).

In addition to their normal function of delimiting macro arguments, braces inside names have the following special functions:

1. They are used to indicate that multiple characters should be considered a single “compound” character when extracting initials. For example, *Yuri* becomes *Y.*, but  $\{Yu\}ri$  becomes *Yu*.

An important aspect of this use of braces is that it only applies to the first characters of a given name. As we’ll see below, this has important implications for our parsing code, which must preserve braces at the beginning of given names, but can be more cavalier with braces in other positions.

2. Spaces and commas are ordinarily interpreted as name separators, rather than name components. Similarly, periods and hyphens usually have a special interpretation. All these characters can be stripped of their special meanings by putting them within braces.

In practice, it might be possible to insert other tokens (such as macros) into names as long as they either (a) are non-expandable or (b) expand into a series of tokens of the above enumerated types. However, in such cases it will probably be safer to declare the macro in question as either a text accent or a text symbol.

### 6.16.1 Text accents

Syntactically, a text accent is a macro that takes a single, undelimited argument, i.e. it has a “prototype” of `macro:#1->`. Semantically, the implication is that it takes a letter (the *base*) as an argument and produces a glyph that for certain purposes can be considered equivalent to the base (see the discussion of stem comparison on page 86).<sup>3</sup>

`\amsrefs@textaccents` This will contain a list of accent commands in standard L<sup>A</sup>T<sub>E</sub>X format (i.e., separated by the token `\do`). For example, after registering the  $\backslash$ " and  $\backslash$ ' accents, it will contain

```
\do \"\do \'
```

```
1782 \let\amsrefs@textaccents\@empty
```

<sup>3</sup>Note that this is meant to be a pragmatic definition for the purposes of this package. No claim is made to greater generality.

```

\DeclareNameAccent Arguments:
    #1 <- accent.
1783 \def\DeclareNameAccent{%
1784     \@lappend\amsrefs@textaccents
1785 }

```

Here are all the standard L<sup>A</sup>T<sub>E</sub>X accents, as well as a few nonstandard accents from the `mathscinet` package.

```

1786 \DeclareNameAccent\"
1787 \DeclareNameAccent\'
1788 \DeclareNameAccent\
1789 \DeclareNameAccent=
1790 \DeclareNameAccent^
1791 \DeclareNameAccent`
1792 \DeclareNameAccent~%
1793 \DeclareNameAccent\b
1794 \DeclareNameAccent\c
1795 \DeclareNameAccent\d
1796 \DeclareNameAccent\H
1797 \DeclareNameAccent\k
1798 \DeclareNameAccent\r
1799 \DeclareNameAccent\t
1800 \DeclareNameAccent\u
1801 \DeclareNameAccent\v

```

From `mathscinet`:

```

1802 \DeclareNameAccent\utilde
1803 \DeclareNameAccent\uarc
1804 \DeclareNameAccent\dudot
1805 \DeclareNameAccent\lfhook
1806 \DeclareNameAccent\udot
1807 \DeclareNameAccent\polhk
1808 \DeclareNameAccent\soft

```

`\etalchar` and `\etaltext` are sort of accent-like if you look at them in the right light.

```

1809 \DeclareNameAccent\etalchar
1810 %\DeclareNameAccent\etaltext

```

### 6.16.2 Text symbols

Syntactically, a text symbol is a macro with a empty parameter `text`, i.e., a prototype of `macro:->`. Semantically, it's a letter-like glyph that should not be considered equivalent to any other glyph or group of glyphs. In addition, it may exist in both upper- and lowercase variants, unlike text accents, where we consider the case to be an attribute of the base letter, not of the accent.<sup>4</sup>

`\amsrefs@textsymbols` This is analogous to `\amsrefs@textaccents` but a little more complicated due to the need to store lowercase equivalents. It consists of a list of double entries of the form

---

<sup>4</sup>As with text accents, this is not intended as a fully general definition.

```
\do \symbol \do \lcsymbol
```

which means that `\symbol` is a text symbol whose corresponding lowercase version is `\lcsymbol`. (Note that nothing is implied about whether `\symbol` is to be considered as uppercase or lowercase.) For example, in

```
\do \ae \do \ae \do \OE \do \oe
```

the first four tokens indicate that `\ae` is a text symbol with lowercase equivalent `\ae`, while the last four tokens indicate that `\OE` is a text symbol with lowercase equivalent `\oe`. This scheme is somewhat redundant, but pleasingly simple.

This also duplicates some of the information in `\@uclclist`, but it seems safer to do this than to modify `\@uclclist`.

```
1811 \let\amsrefs@textsymbols\@empty
```

```
\DeclareNameSymbol Arguments:
```

```
    #1 <- symbol.
    #2 <- lowercase.
```

```
1812 \def\DeclareNameSymbol#1#2{%
1813     \@lappend\amsrefs@textsymbols#1%
1814     \@lappend\amsrefs@textsymbols#2%
1815     \ifx#1#2\else
1816         \@lappend\amsrefs@textsymbols#2%
1817         \@lappend\amsrefs@textsymbols#2%
1818     \fi
1819 }
```

Here are the standard L<sup>A</sup>T<sub>E</sub>X and `mathscinet` text symbols.

Note that `\i` and `\j` are anomalous in being syntactically like text symbols, but semantically more like text accents.

```
1820 \DeclareNameSymbol\i\i
1821 \DeclareNameSymbol\j\j
1822 \DeclareNameSymbol\AE\ae
1823 \DeclareNameSymbol\OE\oe
1824 \DeclareNameSymbol\O\o
1825 \DeclareNameSymbol\DH\dh
1826 \DeclareNameSymbol\DJ\dj
1827 \DeclareNameSymbol\L\l
1828 \DeclareNameSymbol\NG\ng
1829 \DeclareNameSymbol\SS\ss
1830 \DeclareNameSymbol\TH\th
```

From `mathscinet`:

```
1831 \DeclareNameSymbol\Dbar\dbar
1832 \DeclareNameSymbol\lasp\lasp
1833 \DeclareNameSymbol\rasp\rasp
1834 \DeclareNameSymbol\cprime\cprime
1835 \DeclareNameSymbol\cdprime\cdprime
1836 \DeclareNameSymbol\bud\bud
1837 \DeclareNameSymbol\cydot\cydot
```

`~` can be considered a text symbol in much the same way that `\etalchar` can be considered an accent.

```
1838 \DeclareNameSymbol~~%
```

### 6.16.3 `\edef`-like macros for names

The following macros all behave sort of like `\edef`, in the sense that

```
\X@edef\foo{name}
```

defines `\foo` to be the result of expanding `name` and applying a certain transformation to it.

`\normalize@edef` This converts accents in the name to a normalized form where the accent and its argument are surrounded by braces. E.g., after

```
\normalize@edef\cs{P\'olya}
```

`\cs` will contain `P{\'o}lya`. (This might result in a redundant layer of braces if the original text contained, say, “`P{\'o}lya`”, but that’s ok.) This lets us extract the first  $n$  characters from a name by using TeX’s macro argument-gobbling mechanism without worrying that an accent will be separated from its base letter. As a bonus, it also replaces ties (`~`) by spaces.

```
1839 \def\normalize@edef#1#2{%
1840     \begingroup
1841         \@apply\auto@protect\amsrefs@textsymbols
1842         \@apply\wrap@accent\amsrefs@textaccents
    Redefine \@tabacckludge in case someone wants to use this with the inputenc
    package.
1843     \let\@tabacckludge\use@accent
1844     \let~\space
1845     \edef\@tempa{\def\@nx#1{#2}}%
1846     \@xp\endgroup
1847     \@tempa
1848 }
```

`\use@accent` This is identical to `\@nameuse` except for the addition of the `\string`, which, as per `ltoutenc.dtx`, guards against the eventuality that something like `'` might be active at the point of use. We don’t expect to find a `\bib` in the middle of a `\tabbing` environment (do we?) so we

```
1849 \def\use@accent#1{\csname\string#1\endcsname}
```

`\wrap@accent` Here’s a wrapper macro that causes an accent to become auto-wrapping. E.g., after `\wrap@accent\'`, `\'o` will expand to `{\'o}`.

```
1850 \def\wrap@accent#1{%
1851     \def##1#1{\@nx##1}%
1852 }
```

`\lc@edef` This converts all the characters in a name to all lowercase, using the mapping defined by `\amsrefs@textsymbols`. So, after

```
\lc@edef\cs{P\'olya}
```



`\cs` will contain `p\’olya`. Note that accents are not wrapped and ties are passed through unmolested.

```

1853 \def\lc@edef#1#2{%
1854     \begingroup
1855         \let\@tabacckludge\use@accent %%??
1856         \@apply\auto@protect\amsrefs@textaccents
1857         \@apply\lc@do\amsrefs@textsymbols
1858         \edef\@tempa{\lowercase{\def\@nx#1{#2}}}%
1859     \@xp\endgroup
1860     \@tempa
1861 }

```

`\lc@do` This is a slightly more complicated wrapper macro than previous ones. The first argument is a text symbol; the second argument is the lowercase variant of the symbol. If they’re the same (i.e., the first argument is a lowercase text symbol), we `\auto@protect` it. Otherwise we define the first symbol to expand to the second.

```

1862 \def\lc@do#1\do#2{%
1863     \ifx#1#2%
1864         \auto@protect#1%
1865     \else
1866         \def#1{#2}%
1867     \fi
1868 }

```

`\purge@edef` Removes accents and braces from a name and converts ties to spaces, leaving only letters, punctuation and text symbols. For example,

```
\lc@edef\cs{P{\’o}lya}
```

will put `Polya` in `\cs`.

```

1869 \def\purge@edef#1#2{%
1870     \begingroup
1871         \@apply\auto@protect\amsrefs@textsymbols
1872         \let~\space
1873         \@apply\purge@accent\amsrefs@textaccents
1874         \let\@tabacckludge\@gobble

```

As mentioned above (page 71), `\i` and `\j` are semantically like text accents; hence, they require special treatment here.

```

1875     \def\i{i}%
1876     \def\j{j}%
1877     \edef\@tempa{#2}%
1878     \toks@\@emptytoks
1879     \@xp\purge@edef@ \@tempa \@nil
1880     \edef\@tempa{\def\@nx#1{\the\toks@}}%
1881     \@xp\endgroup
1882     \@tempa
1883 }

```

`\purge@edef@` Peek ahead so `\purge@edef@a` will know whether its argument was originally surrounded by braces.

```
1884 \def\purge@edef@{%
1885   \futurelet\@let@token
1886   \purge@edef@a
1887 }
```

`\purge@edef@a` Process a single “chunk” (i.e., one macro-argument’s worth) of the name.

```
1888 \def\purge@edef@a#1{%
  If we’ve run into the \@nil terminator, we’re done.
1889   \ifx\@let@token\@nil
1890     \let\@tempa\@empty
1891   \else
```

Otherwise, if the argument was originally surrounded by braces, process it recursively before processing the remainder of the token stream.

```
1892     \ifx\@let@token\bgroup
1893       \def\@tempa{%
1894         \purge@edef@ #1\@nil
1895         \purge@edef@
1896       }%
1897     \else
```

If the argument is a single unbracketed token, just copy it into the output.

```
1898       \add@toks@{#1}%
1899       \let\@tempa\purge@edef@
1900     \fi
1901   \fi
1902   \@tempa
1903 }
```

`\purge@accent` This is similar to `\wrap@accent` but it removes the accent command (and possibly a layer of braces surrounding the accent’s argument).

```
1904 \def\purge@accent#1{%
1905   \def##1##1{##1}%
1906 }
```

### 6.17 Name parsing

Parsing names is somewhat complicated because parts of the name can (in principle) be empty (G=given, S=surname, J=jr):

```
author={Doe, John, Jr.}: G={John} S={Doe} J={Jr.}
author={Doe, John}: G={John} S={Doe} J={}
author={Doe, , Jr.}: G={} S={Doe} J={Jr.}
author={Doe}: G={} S={Doe} J={}
author={, John, Jr.}: G={John} S={} J={Jr.}
author={, John}: G={John} S={} J={}
author={, , Jr.}: G={} S={} J={Jr.}
author={}: G={} S={} J={}
```

Not all of these forms are legal, of course, but that’s no excuse for not parsing them correctly.

We also want to be somewhat lenient about the placement of spaces:

```
author={ Doe, John, Jr. }: G={John} S={Doe} J={Jr.}
```

However, because one must have some standards, we assume there are no spaces in the following positions in the input:

1. before periods,
2. before commas,
3. at the end of the name,
4. before or after hyphens.

Thus, we make no attempt to compensate for the misplaced spaces in examples like these:

```
author={Doe , J . , Jr. } : G={J .} S={Doe } J={Jr. }
author={Doe, J. - M.}: G={J. - M.} S={Doe} J={}
```

Also, unless we are generating initials, we don’t try to normalize spaces *after* periods:

```
author={Doe, J.M.}: G={J.M.} S={Doe} J={}
(not G={J. M.})
```

Finally, since we allow authors to group together characters that should be treated as a single unit, we need to be careful to preserve the author’s markup in cases like these:

```
author={Doe, {Yu}ri}: G={{Yu}ri} S={Doe} J={}
author={Doe, {Yu}}: G={{Yu}} S={Doe} J={}
```

This is harder than it seems. For example, consider a naive implementation that uses delimited arguments to pull the name apart:

```
\def\parsename#1,#2\nil{%
  \def\bib'surname{#1}%
  \def\bib'given{#2}%
}
```

```
\parsename Doe, {Yu}ri\nil
```

Unfortunately, this results in the space after the comma becoming part of `\bib'given`: “{Yu}ri”.

Our next thought would be to modify the definition slightly to trick TeX into gobbling the space:

```
\def\parsename#1,#2#3\nil{%
  \def\bib'surname{#1}%
  \def\bib'given{#2#3}%
}
```

Now the space is gone, but—surprise!—so are the braces: “Yuri”. In addition, this approach makes it difficult to handle empty name parts correctly.

To sidestep these problems, instead of blindly gobbling macro arguments, we use `\futurelet` to look ahead at certain strategic moments so we can take the appropriate action (see `\get@namepart@d-f`). We only really care about preserving braces at the start of names (page 69), which simplifies things somewhat.

`\name@split` `\name@split` parses a name into its three parts and stores them in `\bib'surname`, `\bib'given` and `\bib'jr`. If the `initials` option is in force, it also extracts the initials from the given name and stores them in `\bib'initials`.

It expects the name to be parsed to be terminated by `\@nil` and to contain at least three commas. Thus the usual way to invoke it is

```
\name@split <name>,,,\@nil
```

`\name@split` just uses `\get@namepart` to peel off the surname and then passes control to `\name@split@given`. (Note the spiffy continuation-passing programming style.)

```
1907 \def\name@split{%
1908     \get@namepart\bib'surname\name@split@given
1909 }
```

`\name@split@given` Pretty much the same, *mutatis mutandis*...

```
1910 \def\name@split@given{%
1911     \get@namepart\bib'given\name@split@jr
1912 }
```

`\name@split@jr` And again...

```
1913 \def\name@split@jr{%
1914     \get@namepart\bib'jr\name@split@finish
1915 }
```

`\name@split@finish` We have all three parts now. Do some consistency checking, extract the initials from the given name, and then call `\@nilgobble` to remove anything (such as extra commas) left on the stack.

```
1916 \def\name@split@finish{%
1917     \ifx\bib'surname\@empty \EmptyNameWarning \fi
```

Theoretically, we could try to check for uninverted names here, but only at the risk of producing spurious warnings when the name really does only have one part (`author={Arvind}`).

A possible solution: Now that we have the `inverted` attribute, we could issue a warning if the given name is empty and the family name contains a space. I'm sure someone could find valid input that would still generate a spurious warning, but this would take care of the most common cases. This bears more thinking about.

```
1918 %%     \ifx\@empty\bib'given
1919 %%         \NameCheck \bib'surname ??\@nil
1920 %%     \else
1921         \extract@initials\bib'given
```

```

1922 %%      \fi
1923      \@nilgobble
1924 }

```

`\get@namepart` Now for the fun part. `\get@namepart` takes two arguments. The first (the destination) should be a control sequence; the second (the continuation) will normally also be a control sequence, though technically we only require that it be a single token. `\get@namepart` scans everything up to the next level-0 comma, places it in the destination, and then calls the continuation.

```

1925 \def\get@namepart#1#2{%

```

Save the destination in `\toks@` and the continuation in `\@temptokena`. It's unfortunate that this trashes the previous contents of those token lists (as well as the contents of `\@tempa` later on), but preliminary attempts to rewrite the code to leave the calling environment unchanged were not encouraging.

```

1926      \toks@{#1}%
1927      \@temptokena{#2}%
1928      \get@namepart@a
1929 }

```

`\get@namepart@a` Now peek ahead at the next token in the stream and call `\get@namepart@b` to examine it.

```

1930 \def\get@namepart@a{%
1931     \futurelet\@let@token
1932     \get@namepart@b
1933 }

```

`\get@namepart@b` If the next token is a space token, we want to delete it. Otherwise we're ready to read the name.

```

1934 \def\get@namepart@b{%
1935     \ifx\@let@token\@sptoken
1936         \@xp\get@namepart@c
1937     \else
1938         \@xp\get@namepart@d
1939     \fi
1940 }

```

`\get@namepart@c` The next token is a space; we delete it and restart `\get@namepart@a`, in case there are multiple spaces.

```

1941 \def\get@namepart@c{%
1942     \afterdeleting@token\get@namepart@a
1943 }

```

`\get@namepart@d` We're at the beginning of the name part. However, there are still two special cases we have to watch out for. First, the next token might be a comma, meaning that this name part is empty. Second, the next token might be an open brace (`{`), which we have to be sure to copy into the destination. So, we peek ahead again before proceeding.

```

1944 \def\get@namepart@d{%
1945     \futurelet\@let@token
1946     \get@namepart@e
1947 }

```

`\get@namepart@e` If the next token is a comma, it means the name part is empty; so, we set the destination to an empty list and then arrange to execute the continuation after deleting the comma. Otherwise we call `\get@namepart@f` to read a non-empty name, leaving `\@let@token` undisturbed so that `\get@namepart@f` knows what's coming up.

```

1948 \def\get@namepart@e{%
1949     \ifx\@let@token,%
1950         \@xp\let\the\toks@\@empty
1951         \edef\@tempa{%
1952             \@nx\after@deleting@token\the\@temptokena
1953         }%
1954         \@xp\@tempa
1955     \else
1956         \@xp\get@namepart@f
1957     \fi
1958 }

```

`\get@namepart@f` We know whether or not the name begins with a brace, but we don't know if the corresponding group contains the entire name or only part of it. By reading the name as two arguments, we can handle all cases correctly.<sup>5</sup>

Note that the arguments are not expanded.

```

1959 \def\get@namepart@f#1#2,{%
1960     \ifx\@let@token\bgroup
1961         \@xp\def\the\toks@{#1}#2}%
1962     \else
1963         \@xp\def\the\toks@{#1#2}%
1964     \fi
1965     \the\@temptokena
1966 }

```

`\EmptyNameWarning` Or translator or contributor or...

```

1967 \def\EmptyNameWarning{\amsrefs@warning{Empty contributor name}}

```

## 6.18 Extracting initials

Extracting initials from the author's given name is tricky because of the numerous special cases that need to be handled. Consider the following examples, some of which are admittedly contrived:

```

author={Arvind}: I={ }
author={Bing, R H}: I={R H}
author={Harish, \ 'Etienne}: I={É.}

```

<sup>5</sup>More or less. If the second argument is brace-delimited, the braces will be lost. But as mentioned above (page 76), we don't really care.

```

author={Harish, \’E.}: I={É.}
author={Harish, \’{E}.}: I={É.}
author={Harish, {\’E}.}: I={É.}
author={Harish, \’E}: I={É}
author={Harish, \’Etienne-P\^{\i }erre}: I={É.-P.}
author={Jones, David}: I={D.}
author={Jones, David-Michael}: I={D.-M.}
author={Katzenbach, Nicholas {deB}elleville}: I={N. deB.}
author={Katzenbach, Nicholas deB.}: I={N. deB.}
author={Matiyasevich, {Yu}ri}: I={Yu.}
author={Matiyasevich, {Yu}}: I={Yu}
author={Matiyasevich, Yu.}: I={Yu.}

```

When processing initials, we loosen our strictures on spaces inside the given name by not requiring spaces after periods and tolerating them around hyphens and after the name:

```

author={Jones, D.M.}: I={D. M.}
author={Jones, David - Michael}: I={D.-M.}
author={Jones, David , Jr.}: I={D.}

```

(Strictly speaking, only the support for the first of these examples was a deliberate design decision; the other two are side-effects of the implementation. In any case, toleration of these quirks is in no way an endorsement of them, especially since they may make it more difficult for third-party software to correctly process bibliography entries.)

### 6.18.1 The algorithm

As a running example, consider the following contrived input:

```
\’E.-P\^{\i }erre J.K. M
```

which we want to turn into “É.-P. J. K. M”.

We precede by stages.

1. Normalize the name by surrounding accents and their arguments by braces:

```
{\’E}.-P{\^{\i }erre} J.K. M
```

We also replace ~s by spaces at this stage.

2. Replace each hyphen (-) by “\ini@hyphen”:

```
{\’E}. \ini@hyphen P{\^{\i }erre} J.K. M
```

3. Add a space after each period:

```
{\’E}. \ini@hyphen P{\^{\i }erre} J. K. M
```

4. Now we have the name as a list of space-separated components. (In our example, the components are “{\’E}.”, “\ini@hyphen”, “P{\^{\i }erre}”, “J.”, “K.”, and “M”.) We loop through the components and replace each one by its “initialized” form. There are four cases:

- (a) The component ends in a period. Copy it and add the token ~. (In our example, these are the components “{\’E}.”, “J.” and “K.”.)

- (b) The component consists of a single (possibly compound) character without a period. Again, copy it and add `~`. (In our example, this is the component “M”.)
  - (c) The component is the token `\ini@hyphen`. Copy it.
  - (d) The component consists of two or more (possibly compound) characters without a period (e.g., “P{`^{\i }erre`”). Copy the first character and add the tokens `.~`.
5. The token list generated above will end with an unwanted `~`. Delete it.

The end result is

```
{\’E}.~\ini@hyphen P.~J.~K.~M
```

which, when typeset, does indeed produce “É.-P. J. K. M”.<sup>6</sup>

### 6.18.2 The implementation

`\extract@initials` This is pretty straightforward.

```
1968 \def\extract@initials#1{%
1969     \begingroup
1970         \auto@protect\ini@hyphen
1971         \auto@protect\nobreakspace
1972         \let~\relax
1973         \@apply\auto@protect\amsrefs@textsymbols
1974         \@apply\auto@protect\amsrefs@textaccents
1975         \normalize@edef\@tempa{#1}%
1976         \ifx\@tempa\@empty
1977         \else
```

It would be nice if `\process@hyphens` and `\process@dots` commuted, and they almost do. However, suppose you have the (admittedly contrived) name `Yu.-{Yu}`, which should be turned into “Yu.-Yu”. If `\process@dots` is applied first, the braces around the second “Yu” get removed, so the output is “Yu.-Y.”. (Even worse would be `P.-\’E`, which would produce “P.-:”)

```
1978         \process@hyphens\@tempa
1979         \process@dots\@tempa
1980         \process@names\@tempa
1981         \@chomp\@tempa{~}%
1982     \fi
1983     \edef\@tempa{\def\@nx\bib’initials{\@tempa}}%
1984 \@xp\endgroup
1985 \@tempa
1986 }
```

`\ini@hyphen` The `\unskip` removes the space at the end of a potential (and probable) preceding `~`, but leaves the `\nobreak` penalty.

```
1987 \def\ini@hyphen{\unskip-\nobreak}
```

<sup>6</sup>Tying all the characters together is potentially undesirable when, as in the example, there are a large number of pieces in the given name.



`\process@hyphens` This follows the same general pattern as `\get@namepart`, but with an extra layer of grouping to avoid unwanted side-effects. Otherwise, it uses the same parsing techniques.

One difference is that there is no explicit continuation: instead, we iterate by repeatedly calling `\process@one@hyphen@d` until we run into the `\@nil` marker.

```
1988 \def\process@hyphens#1{%
1989     \begingroup
1990         \toks@\emptytoks
1991         \@xp\process@one@hyphen #1-\@nil
1992         \edef\@tempa{\the\toks@}%
```

Because of the - we have to stick in as a delimiter above, `\process@one@hyphen` will always generate unwanted code at the end of the name. We now delete it. (This also has the necessary side-effect of expanding the `\space` macros into space characters.)

```
1993         \@chomp\@tempa{ \ini@hyphen\space}%
1994         \edef\@tempa{\def\@nx#1{\@tempa}}%
1995     \@xp\endgroup
1996     \@tempa
1997 }
```

`\process@one@hyphen` Cf. `\get@namepart@a`.

```
1998 \def\process@one@hyphen{%
1999     \futurelet\@let@token
2000     \process@one@hyphen@a
2001 }
```

`\process@one@hyphen@a` Cf. `\get@namepart@b` and `\extract@initial@a`.

The tests for `\@nil` and - here are purely to supply better error recovery. Without them, a hyphen at the end of the given name (e.g. `author={Doe, John-}`) would produce a very mysterious error message. Since it's unlikely the hyphen really belongs there, we delete it, but we also issue a warning to the author. (It will still show up as part of the full given name, though.)

We borrow `\f@n` from `rkeyval` to keep track of the appropriate next action.

```
2002 \def\process@one@hyphen@a{%
2003     \ifx\@let@token\@nil
2004         \let\f@n\@gobble
2005     \else
2006         \ifx\@let@token -%
2007             \TrailingHyphenWarning
2008             \let\f@n\process@one@hyphen@b
2009         \else
2010             \ifx\@let@token\@sptoken
2011                 \let\f@n\process@one@hyphen@b
2012             \else
2013                 \let\f@n\process@one@hyphen@c
```

```

2014         \fi
2015     \fi
2016 \fi
2017 \f@sa@n
2018 }

```

`\process@one@hyphen@b` Cf. `\get@namepart@c`.

```

2019 \def\process@one@hyphen@b{%
2020     \after@deleting@token\process@one@hyphen
2021 }

```

`\process@one@hyphen@c` Cf. `\get@namepart@f`.

```

2022 \def\process@one@hyphen@c#1#2-{%
2023     \ifx\bgroup\@let@token
2024         \add@toks@{#1}#2 \ini@hyphen\space}%
2025     \else
2026         \add@toks@{#1#2 \ini@hyphen\space}%
2027     \fi
2028     \futurelet\@let@token
2029     \process@one@hyphen@d
2030 }

```

`\process@one@hyphen@d` Here we just check for `\@nil` and terminate if we detect it. Otherwise, we start over.

```

2031 \def\process@one@hyphen@d{%
2032     \ifx\@let@token\@nil
2033         \@xp\@gobble
2034     \else
2035         \@xp\process@one@hyphen
2036     \fi
2037 }

```

`\TrailingHyphenWarning` Or translator or contributor or...

```

2038 \def\TrailingHyphenWarning{%
2039     \amsrefs@warning{Trailing hyphen deleted from name}%
2040 }

```

`\process@dots` This is almost completely parallel to `\process@hyphens`.

```

2041 \def\process@dots#1{%
2042     \begingroup
2043         \toks@\@emptytoks
2044         \@xp\process@one@dot #1.\@nil
2045         \edef\@tempa{\the\toks@}%
2046         \@chomp\@tempa{. }%

```

Since it's legitimate for names to end in periods, we might still have an unwanted space at the end of the name, so we delete it too.

```

2047         \@chomp\@tempa{ }%
2048         \edef\@tempa{\def\@nx#1{\@tempa}}%

```

```

2049 \exp\endgroup
2050 \@tempa
2051 }

```

`\process@one@dot`

```

2052 \def\process@one@dot{%
2053 \futurelet\@let@token
2054 \process@one@dot@a
2055 }

```

`\process@one@dot@a` This is a bit different from `\process@one@hyphen@a` since we expect names sometimes to end in a period—or even two periods—not least because of the `.` we add as a delimiter when invoking `\process@one@dot`.

```

2056 \def\process@one@dot@a{%
2057 \ifx\@let@token .%
2058 \def\fsa@n{\after@deleting@token\process@bare@dot}%
2059 \else
2060 \ifx\@let@token\@sptoken
2061 \let\fsa@n\process@one@dot@b
2062 \else
2063 \let\fsa@n\process@one@dot@c
2064 \fi
2065 \fi
2066 \fsa@n
2067 }

```

`\process@bare@dot`

```

2068 \def\process@bare@dot{%
2069 \add@toks@{. }%
2070 \futurelet\@let@token
2071 \process@one@dot@d
2072 }

```

`\process@one@dot@b`

```

2073 \def\process@one@dot@b{%
2074 \after@deleting@token\process@one@dot
2075 }

```

`\process@one@dot@c`

```

2076 \def\process@one@dot@c#1#2.{%
2077 \ifx\bgroup\@let@token
2078 \add@toks@{#1}#2.}%
2079 \else
2080 \add@toks@{#1#2.}%
2081 \fi
2082 \futurelet\@let@token
2083 \process@one@dot@d
2084 }

```

`\process@one@dot@d`

```
2085 \def\process@one@dot@d{%
2086   \ifx\@let@token\@nil
2087     \xp@gobble
2088   \else
2089     \xp\process@one@dot
2090   \fi
2091 }
```

`\process@names` This is very similar to `\process@hyphens` and `\process@dots`, but with a couple of twists, as noted below.

```
2092 \def\process@names#1{%
2093   \begingroup
2094     \toks@\@emptytoks
2095     \xp\extract@initial #1 \@nil
2096     \edef\@tempa{\def\@nx#1{\the\toks@}}%
2097   \xp@endgroup
2098   \@tempa
2099 }
```

`\extract@initial` Scan through the token stream replacing words by their initials until we hit the terminating `'11`

```
2100 \def\extract@initial{%
2101   \futurelet\@let@token
2102   \extract@initial@a
2103 }
```

`\extract@initial@a` As with `\process@one@hyphen@a`, the test for `'11` here is purely to provide better recovery, this time in case the given name has a trailing space (e.g, `author={Doe, John }`). But since we're just deleting whitespace, we don't bother issuing a warning.

```
2104 \def\extract@initial@a{%
2105   \ifx\@let@token\@nil
2106     \let\fsa@n@gobble
2107   \else
2108     \ifx\@let@token\@sptoken
2109       \let\fsa@n\extract@initial@b
2110     \else
2111       \let\fsa@n\extract@initial@c
2112     \fi
2113   \fi
2114   \fsa@n
2115 }
```

`\extract@initial@b`

```
2116 \def\extract@initial@b{%
2117   \after@deleting@token\extract@initial
2118 }
```

`\extract@initial@c` Here, instead of just copying the name, we extract its initials and copy those.

```

2119 \def\extract@initial@c#1#2 {%
2120   \ifx\@let@token\bgroup
      Note that we double-brace the first argument to avoid having to test \@let@token
      again inside \@extract@initial.
2121     \@extract@initial {{#1}}#2\@nil
2122   \else
2123     \@extract@initial #1#2\@nil
2124   \fi
2125   \futurelet\@let@token
2126   \extract@initial@d
2127 }
```

`\extract@initial@d`

```

2128 \def\extract@initial@d{%
2129   \ifx\@let@token\@nil
2130     \xp@gobble
2131   \else
2132     \xp\extract@initial
2133   \fi
2134 }
```

`\@extract@initial` This handles the four cases mentioned on page 79.

```

2135 \def\@extract@initial#1#2\@nil{%
2136   \ifx\ini@hyphen#1%
2137     \add@toks@\ini@hyphen%
2138   \else
2139     \in@{.\@nil}{#1#2\@nil}% Look for a period at the end of the name
2140     \ifin@
2141       \add@toks@{#1#2~}%
2142     \else
2143       \count@chars\@tempcnta{#1#2}%
2144       \ifnum\@tempcnta > \@ne
2145         \add@toks@{#1.~}%
2146       \else
2147         \add@toks@{#1~}%
2148       \fi
2149     \fi
2150   \fi
2151 }
```

`\count@chars` This sets its first argument (which is assumed to be a count register) to the number of characters in the second argument. Compound characters are counted as a single character.

```

2152 \def\count@chars#1#2{%
2153   \begingroup
2154     \@tempcnta\z@
2155     \@count@chars#2\@nil
```

```

2156     \edef\@tempb{#1=\the\@tempcnta\relax}%
2157     \@xp\endgroup
2158     \@tempb
2159 }

```

```
\@count@chars
```

```

2160 \def\@count@chars#1{%
2161     \ifx #1\@nil
2162     \else
2163         \advance\@tempcnta\@ne
2164         \@xp\@count@chars
2165     \fi
2166 }

```

## 6.19 Generating alphabetic labels

### 6.19.1 The algorithm

Like Gaul, an alphabetic label is divided into three parts.

1. The author part. In the simplest case, this is formed by extracting the first character of each word of each last name of each author. Thus, if there were two authors with last names “Vaughan Williams” and “Tallis”, the author part would be “VWT”.

If there are more than four authors, only the first three names are used, and a superscript “+” is appended to represent the elided names. Similarly, if an author name is “others”, it is replaced by a superscript “+” and any following author names (of which there shouldn’t be any) are ignored.

Finally, if there is only one author and the author’s last name consists of a single word, the first three characters of that name are used.

2. The year part. If the `y2k` option is in force, or if the year is less than 1901, the entire year is used. Otherwise the last two digits of the year are used.<sup>7</sup> The combination of author part and year part will be referred to as the *stem*.
3. The suffix. If two or more items have the same stems, a suffix consisting of a lowercase latin letter will be appended to each label to make it unique.

This third part is more subtle than it might first appear. First, case is ignored when comparing stems, so that, for example, “Ahl1999” and “AHL1999” are considered identical. Second, existing practice (in English, at least), is to ignore diacritics so that, for example, “Ahl1999” and “Ähl1999” are considered identical.

Note that when checking for duplicate stems, we assume that bibliography items appear sorted by label, which means that all items with the same stem will be adjacent. This means we can use the naive algorithm (check to see if the current item has the same stem as the previous item and, if so, append a suffix) to detect clashes. This sorting will be done automatically by `amsxport`, but the document author is responsible for ensuring the appropriate order if

<sup>7</sup>Years with more than 4 digits are not currently handled correctly. *Caveat lector*.

amsxport is not used. This is why it's an error to mix the alphabetic and citation-order options.

### 6.19.2 The implementation

```
2167 \let\previous@stem\@empty
2168 \let\current@stem\@empty
2169 \let\previous@year\@empty
2170 \let\current@year\@empty
```

`\amsrefs@label@prefix`

```
2171 \let\amsrefs@label@prefix\@empty
```

`\thebib`

```
2172 \def\thebib{\amsrefs@label@prefix\@arabic@c@bib}
```

`\append@to@stem`

```
2173 \def\append@to@stem{\global\@concat\current@stem}
```

`\generate@alphalabel`

```
2174 \def\generate@alphalabel{%
```

If the user supplied an explicit label field, we use it. Otherwise, we generate our own.

```
2175 \ifx\bib'label\@empty
2176 \begingroup
```

We begin by saving the previous stem and initializing the current stem to the empty string.

```
2177 \global\let\previous@stem\current@stem
2178 \global\let\current@stem\amsrefs@label@prefix
```

The list of primary contributors is available to us in `\current@primary` in the form

$$\name{Last_1, First_1} \name{Last_2, First_2} \dots \name{Last_n, First_n}$$

We will be executing this list multiple times with various definitions of `\name`. So the first thing we want to do is establish a safe environment and normalize the names.

```
2179 \@apply\auto@protect\amsrefs@textsymbols
2180 \@apply\auto@protect\amsrefs@textaccents
2181 \auto@protect\name
2182 \auto@protect\etaltext
2183 \normalize@edef\@tempa\current@primary
```

Now we count the number of authors in the list and invoke the appropriate macro to calculate the author part of the reference label.

```
2184 \get@numberof\@tempcnta\name\@tempa
2185 \calc@author@part
```

Next append the year part.

```
2186 \append@label@year
```

At this point, the `\current@stem` is complete and we're ready to determine what (if any) suffix is needed to disambiguate it from the previous label.

```
2187         \calc@alpha@suffix
```

We have all the pieces now. Arrange to end the current group and then define `\bib@label` in the enclosing group. (This keeps `\bib@label` from being defined outside the group started by `\bib@start`. This isn't strictly necessary, but it provides a bit of compartmentalization.)

```
2188         \edef\@tempa{%
2189             \def\@nx\bib'label{%
2190                 \current@stem
2191                 \alpha@label@suffix
2192             }%
2193         }%
2194     \exp@endgroup
2195     \@tempa
2196 \fi
2197 }
```

```
\calc@author@part@
```

```
2198 \def\calc@author@part@{%
2199     \ifnum \@tempcnta = 1
2200         \exp\@oneauthorlabel\exp{\@tempa}%
2201     \else
2202         \exp\@multiauthorlabel\exp{\@tempa}%
2203     \fi
2204 }
```

```
\calc@author@part@short
```

```
2205 \def\calc@author@part@short{%
2206     \exp\@multiauthorlabel\exp{\@tempa}%
2207 }
```

`\@firststone` This extracts the first character from a properly prepared author name (i.e., one in which accents are properly wrapped).

```
2208 \def\@firststone#1{\@car#1\@empty\@nil}
```

`\@firstthree` And this extracts the first three characters.

```
2209 \def\@firstthree#1{\@carcube#1\@empty\@empty\@empty\@nil}
```

```
\@nametoken
```

```
2210 \let\@nametoken\@firststone
```

```
\hyph@to@space
```

```
2211 \def\hyph@to@space#1-#{#1 \hyph@to@space}
```

`\@marknames` Since we have a ' with funny catcode already, let's use it (being able to easily put a space after the ' makes things easier).



```

2212 \def\@marknames#1{%
2213   \@ifnotempty{#1}{\surround@names#1 ' }%
2214 }

```

## \surround@names

```

2215 \def\surround@names#1 {%
2216   \ifx '#1%
2217   \else
2218     \@nx\@nametoken{#1}%
2219     \@xp\surround@names
2220   \fi
2221 }

```

## \extract@surnames

```

2222 \def\extract@surnames#1#2{%
2223   \get@namepart\@tempb\@nilgobble #2,\@nil
2224   \edef\@tempb{\@nx\@marknames{\@xp\hyph@to@space\@tempb\@gobble-}}%
2225   \edef#1{\@tempb}%
2226 }

```

## \@oneauthorlabel This is the easy case.

```

2227 \newcommand{\@oneauthorlabel}[1]{%
2228   \def\name##1{%
2229     \extract@surnames\@tempa{##1}%
2230     \get@numberof\@tempcnta\@nametoken\@tempa
2231     \ifnum \@tempcnta = 1
2232       \let\@nametoken\@firstthree
2233     \fi
2234     \append@to@stem{\@tempa}%
2235   }%
2236   #1%
2237 }

```

## \@threeauthors

```

2238 \def\@threeauthors\name#1\name#2\name#3#4\@empty{%
2239   \name{#1}\name{#2}\name{#3}%
2240   \append@to@stem{\etalchar{+}}%
2241 }

```

## \@multiauthorlabel

```

2242 \newcommand{\@multiauthorlabel}[1]{%
2243   \def\name##1{%
2244     \ifx\etaltext ##1%
2245       \def\@tempa{\@nx\etalchar{+}}%
2246       \let\name\@gobble
2247     \else
2248       \extract@surnames\@tempa{##1}%
2249     \fi
2250     \append@to@stem{\@tempa}%

```

```

2251   }%
2252   \ifnum \@tempcnta > 4 \exp \@threeauthors \fi
2253   #1\@empty
2254 }

```

`\etalchar`

```
2255 \newcommand{\etalchar}[1]{${#1}$}
```

`\year@short` For alphanumeric labels, we want to extract the last 2 digits of the year. Here's a way to do that, assuming a 4-digit year.

```
2256 \def\year@short#1#2#3#4\@nil{#3#4}
```

`\append@label@year@`

```

2257 \def\append@label@year@{%
2258   \safe@set\@tempcnta\bib@year
2259   \edef\bib@citeyear{\the\@tempcnta}%
2260   \append@to@stem{%
2261     \ifx\bib@year\@empty
2262     \else
2263       \exp\year@short \bib@citeyear \@nil
2264     \fi
2265   }%
2266 }

```

```
2267 \let\alpha@label@suffix\@empty
```

```
2268
```

```
2269 \newcount\alpha@suffix
```

```
2270 \alpha@suffix\@one
```

```
2271 \let\@suffix@format\@alph
```

`\calc@alpha@suffix`

```

2272 \def\calc@alpha@suffix{%
2273   \@tempwafalse
2274   \compare@stems\previous@stem\current@stem
2275   \ifsame@stems

```

Under the alphabetic option, `\previous@year` and `\current@year` will always be the same (namely, both will be empty), but including the test allows this code to work with the author-year option as well.

```

2276     \ifx\previous@year\current@year
2277     \@tempwatrue
2278     \fi
2279   \else
2280     \begingroup
2281     \let\name\@firstofone
2282     \@ifundefined{amsrefs@stem\current@stem}{%
2283       \expandafter\gdef\csname amsrefs@stem\current@stem\endcsname{}}%
2284     }{%
2285       \DuplicateBibLabelWarning
2286     }%

```

```

2287     \endgroup
2288     \fi
2289     \if@tempswa
2290         \global\advance\alpha@suffix\@ne
2291         \edef\alpha@label@suffix{\@suffix@format\alpha@suffix}%
2292         \ifnum\alpha@suffix=\tw@
2293             \immediate\write\@auxout{%
2294                 \string\ModifyBibLabel{\prev@citekey}%
2295             }%
2296         \fi
2297     \else
2298         \let\alpha@label@suffix\@empty
2299         \global\alpha@suffix\@one
2300         \exp\ifx \csname b@\current@citekey @suffix\endcsname \relax
2301         \else
2302             \edef\alpha@label@suffix{\@suffix@format\alpha@suffix}%
2303         \fi
2304     \fi
2305 }

\ifsame@stems
2306 \newif\ifsame@stems

\compare@stems
2307 \def\compare@stems#1#2{%
2308     \begingroup
2309         \purge@edef\@tempa{#1}%
2310         \purge@edef\@tempb{#2}%
2311         \lc@edef\@tempa{\@tempa}%
2312         \lc@edef\@tempb{\@tempb}%
2313         \ifx\@tempa\@tempb
2314             \def\@tempa{\same@stemstrue}%
2315         \else
2316             \def\@tempa{\same@stemsfalse}%
2317         \fi
2318     \@xp\endgroup
2319     \@tempa
2320 }

\ModifyBibLabel
2321 \def\ModifyBibLabel#1{%
2322     \global\@xp\let\csname b@#1@suffix\endcsname\@empty
2323 }

```

## 6.20 Generating short alphabetic labels

This style for alphabetic labels is somewhat simpler than the regular alphabetic style. The stem consists only of an author part without a year part. The author part is formed in the same way, except that even when there is only a single author with a one-word last name, only the first letter of the name is used,

not the first three. Finally, the suffix used to disambiguate identical stems is numeric rather than alphabetic.

See section 6.26.2 on page 108 for the implementation.

## 6.21 Formatting series

The `\PrintSeries` command prints a list of objects in series form. The essential idea is to produce something like “A, B, and C” when we are given three elements “A”, “B”, and “C”, with suitable variations in the punctuation and other intervening material depending on the number of elements.

More precisely, we can envision `\PrintSeries` being called as

```
\PrintSeries{S}{i1}{i2}{i3}{E}{\do{T1} ... \do{Tn}}
```

where *S* and *E* are material to be interpolated before the start and after the end of the list, respectively, *i*<sub>1</sub>, . . . , *i*<sub>3</sub> are material to be interpolated between the elements, and the final argument is a list of indeterminate length where each element consists of a macro and its argument. If there are exactly two elements, *i*<sub>1</sub> is inserted between them; otherwise, *i*<sub>2</sub> is inserted between each pair of items except for the last pair, where *i*<sub>3</sub> is inserted. Thus,

|          |                                                                                                              |
|----------|--------------------------------------------------------------------------------------------------------------|
| <i>n</i> | output                                                                                                       |
| 1        | <i>S T<sub>1</sub> E</i>                                                                                     |
| 2        | <i>S T<sub>1</sub> i<sub>1</sub> T<sub>2</sub> E</i>                                                         |
| 3        | <i>S T<sub>1</sub> i<sub>2</sub> T<sub>2</sub> i<sub>3</sub> T<sub>3</sub> E</i>                             |
| 4        | <i>S T<sub>1</sub> i<sub>2</sub> T<sub>2</sub> i<sub>2</sub> T<sub>3</sub> i<sub>3</sub> T<sub>4</sub> E</i> |

and so forth. For example, a standard comma-separated list could be formatted by

```
\PrintSeries{ }{ and }{, }{, and }{...}
```

That is the simple case but in practice there are additional complications. What if user-supplied line breaks have to be supported at the boundaries between elements? What if in addition to adding material between elements we also want to apply some handy function to each element (e.g., `\textsc`)? Even worse, what if we want the function to be different depending on the position of the element in the list? Indeed if this did not happen to be the case with the current application I would not have gone to the extra trouble of supporting it. But if it must be so, then the output that we need from a list `\do{A}\do{B} . . .` is

```
f0{A}
f0{A} p1 i1 f1{B}
f0{A} p2 i2 f2{B} p3 i3 f3{B}
```

and so on, where

- *f<sub>n</sub>* is a macro taking one argument,
- *p<sub>n</sub>* is punctuation—material that must precede a line break if one occurs at this boundary,
- *i<sub>n</sub>* other interpolated material, as before.

To reduce the number of distinct required objects we decree that each element will get braces wrapped around it as a matter of course; then it is possible for  $f_1$ ,  $f_2$ ,  $f_3$  to be assimilated onto the tail end of  $i_1$ ,  $i_2$ ,  $i_3$ . Since we also have to specify the macro that delimits the elements of the list, we end up with the following rather formidable signature:

$$\backslash\text{PrintSeries}\{\backslash\text{m}\} \{f_0\} \{p_1\}\{i_1f_1\} \{p_2\}\{i_2f_2\} \{p_3\}\{i_3f_3\} \\ \{S\} \{\backslash\text{m}\{T_1\}\dots\backslash\text{m}\{T_n\}\} \{E\}$$

and our comma-separated list example becomes

$$\backslash\text{PrintSeries}\{\backslash\text{do}\}\{ \}\{ \text{ and } \}\{ , \}\{ \}\{ \text{ and } \}\{ \}\{ \dots \}\{ \}$$

`\series@index` First we define a dedicated count register to be used in tracking the ordinal number of the item currently being processed.

```
2324 \newcount\series@index
```

`\PrintSeries`

```
2325 \def\PrintSeries#1#2#3#4#5#6#7#8{%
2326     \begingroup
2327         \def\series@add@a{#2}%
2328         \def\series@add@b{\SwapBreak{#3}#4}%
2329         \def\series@add@c{\SwapBreak{#5}#6}%
2330         \def\series@add@d{\SwapBreak{#7}#8}%
2331         \def\series@add@e{\SwapBreak{#7}}%
2332         \PrintSeries@a{#1}%
2333 }
```

`\PrintSeries@a` For `\PrintSeries@a` the first arg is the iterator function present in the list which is arg 3. Args 2 and 4 are extra material to be added before and after the list that may require the use of `\Plural` or `\SingularPlural`.

```
2334 \def\PrintSeries@a#1#2#3#4{%
2335     \get@numberof\@tempcnta#1{#3}%
2336     \chardef\series@total=\@tempcnta
2337     \ifnum\series@total=\@one
2338         \let\SingularPlural\@firstoftwo
2339     \else
2340         \let\SingularPlural\@secondoftwo
2341     \fi
2342     \series@index=\z@
2343     \let#1\series@add
2344     #2#3#4\relax
2345     \endgroup
2346 }
```

`\series@add` This is the inner function called by `\PrintSeries` that carefully distributes all the material stored previously in `\series@add@...` macros.

Note that the handling of “et al.” cases is somewhat hardcoded. This seemed preferable to adding yet another argument (or two!) to `\PrintSeries`.

```
2347 \def\series@add#1{%
```

```

2348     \advance\series@index\@ne
2349     \ifx\etaltext#1\relax
2350         \ifnum\series@index=\tw@
2351             \def\@tempa{\space\SubEtal}%
2352         \else
2353             \def\@tempa{\series@add@e\space\SubEtal}%
2354         \fi

```

We assume there are fewer than 20,000 items in the list.

```

2355     \series@index\@MM
2356     \else
2357         \ifcase\series@index
2358         \or

```

Material before name 1:

```

2359         \let\@tempa\series@add@a
2360     \or

```

Material before name 2:

```

2361         \ifnum\series@total<\thr@@
2362             \let\@tempa\series@add@b
2363         \else
2364             \let\@tempa\series@add@c
2365         \fi
2366     \else

```

Material before names 3, 4, 5,...

```

2367         \ifnum\series@index=\series@total
2368             \let\@tempa\series@add@d
2369         \else
2370             \ifnum\series@index<\series@total
2371                 \let\@tempa\series@add@c
2372             \else
2373                 \let\@tempa\@gobble
2374             \fi
2375         \fi
2376     \fi
2377 \fi
2378 \@tempa{#1}%
2379 }

```

**\SwapBreak** This takes a single argument, which should begin with a punctuation character, and conditionally appends it to the current horizontal list after removing any preceding whitespace. If there was also a penalty at the end of the hlist (presumed to be the result of a `\linebreak` at the end of a field value), it moves the penalty to *after* the argument.

*Known bug:* `\SwapBreak` interferes with  $\TeX$ 's kerning mechanism. For example, consider a field value that ends with a “y” and that should have a comma automatically appended. `amsrefs` generates the equivalent of `y\SwapBreak{,}`, which results in “y,” (no kern before the comma) rather than “y,”. Unfortunately, fixing this would

likely require a disproportionate effort. In cases where the lack of kerning is unacceptable, a workaround is to add the punctuation mark to the field value manually. For example, `title={...y,}` would generate the equivalent of `y,\SwapBreak{,}`, which in turn would produce “y,” since `\SwapBreak` is careful not to add duplicate punctuation.

```
2380 \def\SwapBreak#1{%
2381   \relax\ifvmode\leavevmode\fi
2382   \@tempcnta\@MM
2383   \toks@{#1}%
```

First, remove any preceding glue. (There usually shouldn't be any of this.)

```
2384   \unskip
```

There might be also be kern, typically an italic correction left there by a previous `TextFontCommand` like `\textit`. But don't remove the special 1 sp kern used to mark the beginning of a bibliography entry.

*Known bug:* Sometimes we want to keep the italic correction.

```
2385   \ifnum \lastkern>\@ne \unkern \fi
```

And now look for a penalty and stash it in a safe place.

```
2386   \ifnum\lastpenalty=\z@
2387   \else
2388     \@tempcnta\lastpenalty
2389     \unpenalty
2390   \fi
```

Now we add the punctuation, *unless* one of the following conditions is true:

1. The last item on the horizontal list was a kern of 1 sp, indicating that we're at the very beginning of a bibliography item.
2. The current space factor is equal to the `\sfcode` of the punctuation mark we are adding, meaning that the mark is already on the list.
3. The current space factor is equal to the special value `\@nopunctsfcode`, meaning that `\nopunct` was specified.

This relies on distinct punctuation marks having distinct space factors, as established by our definition of `\frenchspacing`.

```
2391   \edef\@tempaf%
2392     \@nx\deferredquoteslogical
2393     \ifnum\lastkern=\@ne
2394     \else
2395       \ifnum\spacefactor=\sfcode\@xp\@xp\@xp'\@xp\@car\string#1)\@nil
2396     \else
2397       \ifnum\spacefactor=\@nopunctsfcode
2398       \else
2399         \the\toks@
2400       \fi
2401     \fi
2402   \fi
```

```

2403     \@nx\deferredquotes
2404     \ifnum\@tempcnta=\@MM \else \penalty\number\@tempcnta\space \fi
2405     \ifnum\lastkern=\@ne \ignorespaces \fi
2406   }%
2407   \@tempa
2408 }

```

`\Plural` `\Plural` takes one argument and prints it if there were two or more elements in the current list. So, to get “editors” instead of “editor” after printing a list of editor names, write `editor\Plural{s}`.

`\SingularPlural` takes two arguments and prints the first if there was only one element, otherwise prints the second arg.

```

2409 \newcommand{\SingularPlural}[2]{#1}
2410 \newcommand{\Plural}{\SingularPlural{}}

```

## 6.22 Formatting names and series of names

Now that we have a general mechanism for formatting series, we can easily specialize to the common case of a comma-separated list of names. First we provide specifications for the three most common name formats.

`\setbib@nameLE` This sets a name in standard western uninverted order, e.g., “John Doe Jr.” (The “LE” stands for little-endian.)

```

2411 \BibSpec{nameLE}{
2412   +{ }{ }{given}
2413   +{ }{\IfEmptyBibField{given}{ }{ }}{surname}
2414   +{ }{ }{jr}
2415 }

```

`\setbib@nameBE` Big-endian order, as used for example in traditional Chinese, Japanese, Vietnamese, and Hungarian names: “Doe John”. Big-endian formatting can be requested for name by setting the “inverted” property to “yes.”

```

2416 \BibSpec{nameBE}{
2417   +{ }{ }{surname}
2418   +{ }{ }{given}

```

I don’t know what should happen if there’s a suffix, so I’m going to just leave it out for now (although I should probably issue a warning). I suspect that either (a) it never comes up or (b) if it does come up, there’s no set standard for how it should be handled.

```

2419 %   +{ }{ }{jr}
2420 }

```

`\setbib@nameinverted` Inverted western-style names: “Doe, John, Jr.”

```

2421 \BibSpec{nameinverted}{
2422   +{ } { } {surname}
2423   +{ },{ } {given}
2424   +{ },{ } {jr}
2425 }

```



Incidentally, it would probably be cleaner if names had their own namespace like properties do, i.e., something like

```
\DefineSimpleKey{name}{given}
\DefineSimpleKey{name}{initials}
\DefineSimpleKey{name}{surname}
\DefineSimpleKey{name}{jr}
```

followed by

```
\NameSpec{nameLE}{...}
```

or

```
\BibSpec[name]{nameLE}{...}
```

But this seems a little extravagant at this stage, so I've decided to leave things as-is for now.

`\PrintNames` `\PrintNames` is a simplified interface to `\PrintSeries` that takes only the last three arguments:

```
\PrintNames {S} {E} {\name{T1}... \name{Tn}}
```

The order of the last two arguments is reversed to make it moderately easier to use; cf. `\PrintEditorsA`, etc.

The first name in a series is treated differently than the other names in the `author-year` style, so we use a separate formatting macro for it.

```
2426 \newcommand{\PrintNames}{%
2427   \@ifstar{\PrintNames@a\set@othername}{\PrintNames@a\set@firstname}%
2428 }
```

`\PrintNames@a`

```
2429 \newcommand{\PrintNames@a}[4]{%
2430   \PrintSeries{\name}
2431     {#1}
2432     {}{ and \set@othername}
2433     {,}{ \set@othername}
2434     {,}{ and \set@othername}
2435     {#2}{#4}{#3}%
2436 }
```

`\set@firstname` By default, the first name is formatted in little-endian format. The `author-year` option changes this to inverted order.

```
2437 \def\set@firstname#1{%
2438   \set@name{#1}\setbib@nameLE
2439 }
```

`\set@othername` The rest of the names are set in little-endian format by default.

```
2440 \def\set@othername#1{%
2441   \set@name{#1}\setbib@nameLE
2442 }
```

`\set@name` Parse the name into its components and then pass control to `\set@name@a`, which will decide what format to use for the name.

```
2443 \def\set@name#1{%
2444     \name@split#1,,,\@nil
2445     \set@name@a
2446 }
```

`\set@namea` Use the requested format unless the `order` property has been set to “inverted.”

```
2447 \def\set@name@a#1{%
2448     \begingroup
2449         \get@current@properties
2450         \select@auxlanguage
2451         \def\@tempa{yes}%
2452         \ifx\@tempa\prop'inverted
2453             \setbib@nameBE
2454         \else
2455             #1%
2456         \fi
2457     \endgroup
2458 }
```

`\PrintPrimary`

```
2459 \def\PrintPrimary{%
2460     \ifx\current@primary\@empty
2461         \EmptyPrimaryWarning
2462     \else
2463         \print@primary\current@primary
2464     \fi
2465 }
```

`\EmptyPrimaryWarning`

```
2466 \def\EmptyPrimaryWarning{%
2467     \amsrefs@warning{No authors, editors or translators}%
2468 }
```

`\PrintAuthors` The comparison of `\previous@primary` and `\current@primary` doesn't look at auxiliary properties (see also `\PrintEditorsA` and `\PrintTranslatorsA`). This is probably ok.

```
2469 \newcommand{\PrintAuthors}[1]{%
2470     \ifx\previous@primary\current@primary
2471         \sameauthors\@empty
2472     \else
2473         \def\current@bibfield{\bib'author}%
2474         \PrintNames{}-{}{#1}%
2475     \fi
2476 }
```

`\sameauthors`

```
2477 \newcommand{\sameauthors}[1]{\bysame#1}
```

`\bysame`

```
2478 \def\bysame{%
2479     \leavevmode\hbox to3em{\hrulefill}\thinspace
2480     \kern\z@
2481 }
```

`\PrintNameList` This just prints the names without any additional information.

```
2482 \newcommand{\PrintNameList}{\PrintNames{}}}
```

`\PrintEditorsC`

```
2483 \newcommand{\PrintEditorsC}[1]{%
2484     \PrintNames{Edited by }{#1}%
2485 }
```

`\PrintEditorsA` When we consider editor names we have to think about some further complications. First, for the case of a book where editor names are listed in place of author names, just copy the same style with a bit of added text at the end.

```
2486 \newcommand{\PrintEditorsA}[1]{%
2487     \ifx\previous@primary\current@primary
2488         \sameauthors{(ed\Plural{s}.)}%
2489     \else
2490         \def\current@bibfield{\bib' editor}%
2491         \PrintNames{{ (ed\Plural{s}.)}{#1}%
2492     \fi
2493     \erase@field\bib' editor
2494 }
```

`\PrintEditorsB`

```
2495 \newcommand{\PrintEditorsB}{%
2496     \PrintNames*{(\}{\SwapBreak{,}~ed\Plural{s}.)}%
2497 }
```

`\PrintContributions`

```
2498 \newcommand{\PrintContributions}[1]{%
2499     \PrintSeries
2500         {\fld@elt}
2501         {\print@contribution}
2502         }{ and \print@contribution}
2503         {,}{ \print@contribution}
2504         {,}{ and \print@contribution}{#1}{%
2505 }
```

`\print@contribution`

```
2506 \newcommand{\print@contribution}[1]{%
2507     \in@=#1}%
2508     \ifin@
2509         \ifnum\series@index=\@one with \fi
2510         \RestrictedSetKeys{bib}{%
```

```

2511         \bib@print@inner\setbib@contribution{\the\rsk@toks}%
2512     }{#1}%
2513 \else
2514     #1%
2515 \fi
2516 }

```

## \resolve@inner

```

2517 \def\resolve@inner#1#2{%
2518     \in@=#2}%
2519 \ifin@
2520     \RestrictedSetKeys-{}{bib}{#1{\the\rsk@toks}}{#2}%
2521 \else
2522     \@ifundefined{bi@#2}{%
2523         \XRefWarning{#2}%
2524     }{%
2525         #1{\csname bi@#2\endcsname}%
2526     }%
2527 \fi
2528 }

```

## \PrintConference

```

2529 \def\PrintConference{%
2530     \resolve@inner{\bib@print@inner\setbib@conference}
2531 }

```

## \PrintConferenceDetails

```

2532 \def\PrintConferenceDetails#1{%
2533     \ifx\@empty\bib'address
2534         \ifx\@empty\bib'date
2535             \else
2536                 \PrintConferenceDetails@
2537             \fi
2538         \else
2539             \PrintConferenceDetails@
2540         \fi
2541 }

```

## \PrintConferenceDetails@

```

2542 \def\PrintConferenceDetails@{%
2543     \ifnum\lastkern=\@ne\else\space\fi(\kern 1sp
2544     \ifx\@empty\bib'address
2545         \else
2546             \bib'address
2547         \fi
2548     \ifx\@empty\bib'date
2549         \else
2550             \SwapBreak{,}\space
2551             \print@date

```

```

2552   \fi
2553   )%\spacefactor\sfcode'\,%
2554 }

```

**\PrintBook**

```

2555 \def\PrintBook{%
2556   \resolve@inner{\bib@print@inner\setbib@innerbook}
2557 }

```

**\PrintReprint**

```

2558 \def\PrintReprint{%
2559   \resolve@inner{\bib@reprint}
2560 }

```

**\bib@reprint**

```

2561 \def\bib@reprint#1{%
2562   \begingroup
2563   #1\relax           % execute definitions locally
2564   \bib@resolve@xrefs
2565   \bib@field@patches
2566   \bib'setup
2567   \IfEmptyBibField{copula}{reprinted in}{\bib'copula} \nopunct
2568   \let\bib'language\@empty
2569   \setbib@book
2570   \endgroup
2571 }

```

**\PrintTranslation**

```

2572 \def\PrintTranslation{%
2573   \resolve@inner{\bib@translation}
2574 }

```

**\bib@translation**

```

2575 \def\bib@translation#1{%
2576   \begingroup
2577   #1\relax           % execute definitions locally
2578   \bib@resolve@xrefs
2579   \bib@field@patches
2580   \bib'setup
2581   \let\PrintPrimary\@empty
2582   \bib@append{;}{ % keep this space!
2583     \IfEmptyBibField{language}{English}{\bib'language} transl.%
2584     \IfEmptyBibField{pages}{ in \kern\@ne sp}{, }%
2585   }\bib'transition
2586   \let\bib'language\@empty
2587   \setbib@@
2588   \endgroup
2589 }

```

**\PrintTranslatorsC**

```
2590 \newcommand{\PrintTranslatorsC}[1]{%
2591   \PrintNames{translated by }{ }{#1}%
2592 }
```

**\PrintTranslatorsA**

```
2593 \newcommand{\PrintTranslatorsA}[1]{%
2594   \ifx\previous@primary\current@primary
2595     \sameauthors{(trans.)}%
2596   \else
2597     \def\current@bibfield{\bib'translator}%
2598     \PrintNames{{ (trans.)}{#1}%
2599   \fi
2600   \erase@field\bib'translator
2601 }
```

**\PrintTranslatorsB**

```
2602 \newcommand{\PrintTranslatorsB}[1]{
2603   \PrintNames*{(}\{\SwapBreak{,}~tran\Plural{s}.)}%
2604 }
```

Some special handling for “et alii” or “and others”.

```
2605 \DefineName{alii}{\etaltext}
2606 \DefineName{others}{\etaltext}
```

**\etaltext** The Chicago Manual of Style suggests that it is slightly better not to italicize  
**\SubEtal** ‘et al’ and some other extremely common abbreviations inherited from Latin.  
 (Compare ‘etc’.)

```
2607 \newcommand{\etaltext}{et al.}
2608 \newcommand{\SubEtal}[1]{\etaltext}
```

### 6.23 The partial field

**\print@partial**

```
2609 \newcommand{\print@partial}{%
2610   \resolve@inner{\bib@print@inner\setbib@partial}
2611 }
```

### 6.24 Special formatting for other fields

**\parenthesize** The **\parenthesize** function adds parentheses around its argument, calling  
**\upn** to optionally prevent italic parentheses from being used.

```
2612 \newcommand{\parenthesize}[1]{%
2613   \leavevmode\push@bracket\upn{ }{#1\pop@bracket
2614 }
```

**\upn** By default, **\upn** is a no-op, meaning that this refinement lies dormant unless  
 the **upref** package or other activation is done. (Probably better done via special  
 fonts, anyway.)

```
2615 \providecommand{\upn}[1]{#1}
```

```

\push@bracket
\pop@bracket 2616 \let\bracket@stack\@empty
                2617
                2618 \def\push@bracket#1{%
                2619     \xdef\bracket@stack{#1\bracket@stack}%
                2620 }
                2621
                2622 \def\pop@bracket{%
                2623     \iffalse\fi
                2624     \@xp\pop@bracket@a\bracket@stack \@empty}%
                2625 }
                2626
                2627 \def\pop@bracket@a#1{%
                2628     \leavevmode\/\upn{#1}%
                2629     \xdef\bracket@stack{\iffalse}\fi
                2630 }

\bibquotes
                2631 \newcommand{\bibquotes}[1]{%
                2632     \textquotedblleft#1%
                2633     \gdef\deferredquotes{%
                2634         \global\let\deferredquotes\@empty
                2635         \textquotedblright
                2636     }%
                2637 }

\mdash Cf. textcmds, where there's also a penalty added.
\ndash 2638 \providecommand{\mdash}{\textemdash}
                2639 \providecommand{\ndash}{\textendash}

\strip@MRprefix
                2640 \def\strip@MRprefix#1#2#3#4\@nil{%
                2641     \def\@tempa{#1#2#3#4}%
                2642     \if#1M%
                2643         \if#2R%
                2644             \def\@tempa{#3#4}%
                2645         \fi
                2646     \fi
                2647 }

\MR
                2648 \def\MR#1{%
                2649     \relax\ifhmode\unskip\spacefactor3000 \space\fi
                2650     \begingroup
                2651         \strip@MRprefix#1\@nil
                2652         \edef\@tempa{MR\@nx\MRhref{\@tempa}{\@tempa}}%
                2653     \@xp\endgroup
                2654     \@tempa
                2655 }

```

```

\MRhref   For older versions of some AMS document classes, this patch is needed.
2656 \providecommand{\MRhref}[2]{#1}

\PrintReviews   Reviews are handled as a list to support the theoretical possibility of multiple
                reviews.
2657 \newcommand{\PrintReviews}[1]{%
2658     \PrintSeries{\fld@elt}{,}
2659 }

\PrintPartials
2660 \newcommand{\PrintPartials}[1]{%
2661     \PrintSeries
2662         {\fld@elt}
2663         {\print@partial}
2664         {;}{\print@partial}
2665         {;}{\print@partial}
2666         {;}{\print@partial}{#1}{%
2667 }

\PrintISBNs   And similarly for ISBNs. There seem to be a few different situations where
                one book might have two different ISBN numbers. Here are the ones I know of
                so far [mjd,2002-02-18]: separate ISBN numbers for hardback and paperback;
                separate ISBN numbers for U.S. edition and European edition.
2668 \newcommand{\PrintISBNs}[1]{%
2669     \PrintSeries{\fld@elt}{,}
2670 }

\voltext
2671 \newcommand{\voltext}{\IfEmptyBibField{series}{Vol.~}{vol.~}}

\issuetext
2672 \newcommand{\issuetext}{no.~}

\DashPages   Scan the contents of a page value to see if it is a single page. Presence of
                \ndash or hyphen is taken to mean no. Probably should test also for spaces
                and commas. [mjd,2000/01/24]
2673 \newcommand{\DashPages}[1]{%
2674     p\pp@scan@a#1@\ndash p@\ndash{\pp@scan#1@-p@-{\@nil}\@nil.~#1%
2675 }
2676
2677 \def\pp@scan#1-#2@-#3#4\@nil{#3}
2678
2679 \def\pp@scan@a#1\ndash#2@\ndash#3#4\@nil{#3}

\eprintpages   If we have eprint info and pages info and no journal name, the pages information
                is presumably the number of pages in the eprint.
2680 \newcommand{\eprintpages}[1]{%
2681     #1\IfEmptyBibField{eprint}{\IfEmptyBibField{journal}{ pp.}{}}%
2682 }

```



`\PrintThesisType`

```

2683 \def\PrintThesisType#1{%
2684   \thesis@type#1?\@nil{#1}%
2685 }
2686
2687 \def\thesis@type#1#2\@nil#3{%
2688   \ifx p#1%
2689     Ph.D. Thesis%
2690   \else
2691     \ifx m#1%
2692       Master's Thesis%
2693     \else
2694       #3%
2695     \fi
2696   \fi
2697 }

```

`\PrintDOI` Perhaps need to add allowbreak penalties at the parentheses in a DOI. Also what about prohibiting a break after the leading S?

```

2698 \newcommand{\PrintDOI}[1]{%
2699   DOI #1%
2700 }

```

`\PrintDatePV` Print date in different forms depending on DOI and volume information.

```

2701 \newcommand{\PrintDatePV}[1]{%
2702   \IfEmptyBibField{doi}{%
2703     \let\@tempa\PrintDate
2704   }{%
2705     \IfEmptyBibField{volume}{%
2706       \let\@tempa\PrintDatePosted
2707     }{%
2708       \let\@tempa\PrintDate
2709     }%
2710   }%
2711   \@tempa{#1}%
2712 }

```

`\PrintDate` The intent is to handle variations such as 1987, August 1987, 1987-08, and 1987-08-14. If the month is present, print August or Aug. or 08 or nothing, at the behest of the bib style.

We've taken some special care to parse out the date info ahead of time, so this function just discards arg 1 and uses the already-parsed value.

```

2713 \newcommand{\PrintDate}[1]{(\print@date)}

```

`\PrintDateB` The same, but without the parentheses.

```

2714 \newcommand{\PrintDateB}[1]{\print@date}

```

`\PrintDateField`

```

2715 \newcommand{\PrintDateField}[1]{%
2716   \@ifempty{#1}{}{%
2717     \begingroup
2718       \bib@parsedate{#1}%
2719       \print@full@date
2720     \endgroup
2721   }%
2722 }

```

`\print@full@date`

```

2723 \def\print@full@date{%
2724   \ifx\bib@month\@empty
2725   \else
2726     \print@month@day
2727   \fi
2728   \bib@year
2729 }

```

`\print@date`

```

2730 \let\print@date\print@full@date

```

`\print@month@day`

```

2731 \def\print@month@day{%
2732   \bib@monthname
2733   \ifx\@empty\bib@day \else \nobreakspace\number 0\bib@day,\fi
2734   \space
2735 }

```

`\bib@monthname` With the Babel package, month names for a given language are typically available in a macro `\month@language`:

```

\def\month@german{\ifcase\month\or
  Januar\or Februar\or M"arz\or April\or Mai\or Juni\or
  Juli\or August\or September\or Oktober\or November\or Dezember\fi}

```

However this is not true for English.

```

2736 \newcommand{\bib@monthname}{%
2737   \ifcase 0\bib@month
2738   \or January\or February\or March\or April\or May\or June\or
2739     July\or August\or September\or October\or November\or December\or
2740     Winter\or Spring\or Summer\or Fall\else Unknown Month%
2741   \fi
2742 }

```

`\PrintYear` You can use `\PrintYear` if you want to suppress month/day even when supplied in the data.

```

2743 \newcommand{\PrintYear}[1]{\bib@year}

```

`\PrintDatePosted` This one is special for AMS use.

```

2744 \newcommand{\PrintDatePosted}[1]{\unskip, posted on \print@date}

```

\PrintEdition

```
2745 \newcommand{\PrintEdition}[1]{%
2746   \afterassignment\print@edition
2747   \count@ 0#1\relax\@nil
2748 }
```

\print@edition If the number assignment swept up all the contents, produce a cardinal number from \count@.

```
2749 \def\print@edition#1#2\@nil{%
2750   \ifx\relax#1\relax
2751     \ifnum\count@>\z@
2752       \CardinalNumeric\count@
2753     \else
2754       ??th%
2755     \fi
2756   \ \editiontext
2757 \else
2758   \ifnum \count@>\z@ \number\count@ \fi
2759   #1#2\relax
2760 \fi
2761 }
```

\editiontext

```
2762 \newcommand{\editiontext}{ed.}
```

\CardinalNumber

```
2763 \newcommand{\CardinalNumeric}[1]{%
2764   \number#1\relax
2765   \if
2766     \ifnum#1<14
2767       \ifnum#1>\thr@@ T\else F\fi
2768     \else
2769       F%
2770     \fi
2771     T%
2772     th%
2773   \else
2774     \@xp\keep@last@digit\@xp#1\number#1\relax
2775     \ifcase#1th\or st\or nd\or rd\else th\fi
2776   \fi
2777 }
```

\keep@last@digit

```
2778 \def\keep@last@digit#1#2{%
2779   \ifx\relax#2%
2780     \@xp\@gobbletwo
2781   \else
2782     #1=#2\relax
2783   \fi
```

```
2784 \keep@last@digit#1%
2785 }
```

`\SentenceSpace` Note how careful we are here to preserve `\frenchspacing`.

```
2786 \newcommand{\SentenceSpace}{\relax\ifhmode\spacefactor'\. \fi}
```

`\eprint` For now, this does nothing. Could do a url/hyperlink or something.

```
2787 \newcommand{\eprint}[1]{\url{#1}}
```

The [www.arXiv.org](http://www.arXiv.org) recommendations for citing their eprints are found at <http://xxx.lanl.gov/help/faq/references>, including these examples:

```
arXiv:hep-th/9910001
arXiv:math.AT/9910001
arXiv:physics.acc-ph/9911027
```

## 6.25 Bib $\TeX$ support

`\bibliographystyle` Disable `\bibliographystyle` since we're going to handle that behind the scenes.

```
2788 \let\bibliographystyle@gobble
```

`\bibtex@style`

```
2789 \def\bibtex@style{amsrn}
```

```
2790 \AtBeginDocument{
```

```
2791 \if@filesw
```

```
2792 \immediate\write\auxout{\string\bibstyle{\bibtex@style}}%
```

```
2793 \fi
```

```
2794 }
```

## 6.26 Implementing package options

### 6.26.1 The alphabetic option

```
2795 \IfOption{alphabetic}{%
```

```
2796 \def\bibtex@style{amsra}%
```

```
2797 \let\process@citelist\process@citelist@unsorted
```

```
2798 \amsrefs@option@alphabetic
```

```
2799 }{-}
```

### 6.26.2 The shortalphabetic option

```
2800 \IfOption{shortalphabetic}{%
```

```
2801 \def\bibtex@style{amsrs}%
```

```
2802 \let\process@citelist\process@citelist@unsorted
```

```
2803 \amsrefs@option@shortalphabetic
```

```
2804 }{-}
```

### 6.26.3 The backrefs option

Rather than checking for the `backrefs` option *per se*, we check to see if the `backref` package is loaded. This accomodates authors who load the `backref` package explicitly but do not pass the `backrefs` option to `amsrefs`.

```
2805 \AtBeginDocument{%
```

```

2806 \@ifpackageloaded{backref}{%
2807     \let\PrintBackRefs\print@backrefs
2808     \let\BackCite\back@cite

```

The `backref` package uses `\@starttoc` inside `\thebibliography` to open and read the `.brf` file. We could do something similar with `\biblist`, but it seems cleaner to use `\AtBeginDocument`. Unfortunately, `amsart` redefines `\@starttoc` in a way that interacts badly with this use. So, we inline the relevant parts of `\@starttoc` here. (The group and `\makeatletter` are unnecessary at present, but I'll leave them in as future-proofing.)

```

2809     \begingroup
2810         \makeatletter
2811         \@input{\jobname.brf}%
2812         \if@filesw
2813             \newwrite\tf@brf
2814             \immediate\openout\tf@brf \jobname.brf\relax
2815         \fi
2816     \endgroup
2817 }{%
2818 }

```

#### 6.26.4 The citation-order option

```

2819 \IfOption{citation-order}{%
2820     \IfOption{alphabetic}{%
2821         \amsrefs@warning@nl{%
2822             The citation-order and alphabetic options are
2823             incompatible%
2824         }%
2825     }{
2826         \def\bibtex@style{amsru}%
2827     }
2828 }{}

```

#### 6.26.5 The initials option

```

2829 \IfOption{initials}{% TRUE:
2830     \BibSpec{nameLE}{
2831         +{}{}{initials}
2832         +{}{\IfEmptyBibField{initials}{}{ }}{surname}
2833         +{}{ }{jr}
2834     }
2835
2836     \BibSpec{nameBE}{
2837         +{}{}{surname}
2838         +{}{ }{initials}
2839     %   +{}{ }{jr}
2840     }
2841
2842     \BibSpec{nameinverted}{
2843         +{} {} {surname}
2844         +{,}{ } {initials}

```

```

2845     +{,}{ } {jr}
2846   }
2847 }{% initials? FALSE:
2848 %   \let\extract@initials@gobble
2849 } % end conditional code for initials option

```

### 6.26.6 The logical-quotes option

`\deferredquotes`

```
2850 \let\deferredquotes@empty
```

`\deferredquoteslogical`

```

2851 \IfOption{logical-quotes}{%
2852   \def\deferredquoteslogical{\deferredquotes}%
2853 }{%
2854   \let\deferredquoteslogical\relax
2855 }

```

### 6.26.7 The non-compressed-cites option

```

2856 \IfOption{non-compressed-cites}{%
2857   \let\cite@compress\cite@print
2858 }{}

```

### 6.26.8 The non-sorted-cites option

```

2859 \IfOption{non-sorted-cites}{%
2860   \let\process@citelist\process@citelist@unsorted
2861 }{}

```

### 6.26.9 The short-journals option

```

2862 \IfOption{short-journals}{%
2863   \renewcommand{\DefineJournal}[4]{%
2864     \bib*{#1}{periodical}{
2865       issn={#2},
2866       journal={#3},
2867     }%
2868   }
2869 }{}

```

### 6.26.10 The short-publishers option

```

2870 \IfOption{short-publishers}{%
2871   \renewcommand{\DefinePublisher}[4]{%
2872     \bib*{#1}{publisher}{%
2873       publisher={#2},%

```

Maybe `short-publishers` should suppress the `address`? Or is that a separate option? I sense a combinatorial explosion coming on...

```

2874       address={#4},
2875     }%
2876   }%
2877 }{}

```

### 6.26.11 The short-months option

```

2878 \IfOption{short-months}{%
2879   \renewcommand{\bib@monthname}{%
2880     \ifcase 0\bib@month
2881     \or Jan.\or Feb.\or Mar.\or Apr.\or May\or June\or
2882     July\or Aug.\or Sep.\or Oct.\or Nov.\or Dec.\or
2883     Winter\or Spring\or Summer\or Fall\else Unknown Month%
2884   \fi
2885 }%
2886 }{}

```

### 6.26.12 The y2k option

```

2887 \IfOption{y2k}{%
2888   \IfOption{alphabetic}{%
2889     \def\year@short#1\@nil{#1}%
2890     \def\bibtex@style{amsry}%
2891   }{%
2892     \amsrefs@warning@nl{%
2893       The y2k option can only be used with the^^J%
2894       alphabetic option%
2895     }%
2896 }
2897 }{}

```

### 6.26.13 The bibtex-style option

```

2898 \IfOption{bibtex-style}{%
2899   \RequirePackage{amsbst}
2900 }{}

```

### 6.26.14 The msc-links option

```

2901 \IfOption{msc-links}{%

```

Unless you're using pdf<sub>T</sub><sub>E</sub><sub>X</sub>, links cannot be broken across lines, which causes problems for long-form MR numbers such as “MR2149145 (2006d:01012)”. To mitigate the problem, we manually break such numbers into two separate links.

```

2902   \@ifundefined{href}{}{%
2903     \def\parse@MR#1 (#2)#3\@nil{%
2904       \def\MR@url{http://www.ams.org/mathscinet-getitem?mr=#1}%
2905       \def\@tempd{#1}%
2906       \def\@tempe{#2}%
2907     }%
2908     \def\MRhref#1#2{%
2909       \begingroup
2910         \parse@MR#1 ()\@empty\@nil%
2911         \href{\MR@url}{\@tempd\vphantom{()}}%
2912         \ifx\@tempe\@empty
2913         \else
2914           \ \href{\MR@url}{(\@tempe)}%
2915         \fi
2916       \endgroup
2917     }%
2918   }{}

```

2919 }{}

### 6.26.15 The author-year option

Here ends the `amsrefs` package, unless the `author-year` option is in effect; then we want to use some different bibspecs.

2920 \IfOption{author-year}{\PopCatcodes \endinput}

\@biblistsetup

```

2921 \renewcommand{\@biblistsetup}[1]{%
2922   \RestrictedSetKeys{}{biblist}{\the\rsk@toks}{#1}%
2923   \rkvIfEmpty{biblist}{prefix}{}{%
2924     \amsrefs@warning{%
2925       The 'prefix' biblist option cannot be used\MessageBreak
2926       with the author-year option.%
2927     }%
2928   }%
2929   \rkvIfEmpty{biblist}{labels}{}{%
2930     \amsrefs@warning{%
2931       The 'labels' biblist option cannot be used\MessageBreak
2932       with the author-year option.%
2933     }%
2934   }%
2935 }
```

\generate@label

```

2936 \def\generate@label{%
  If the user supplied an explicit label field, we use it. Otherwise, we generate
  our own.
2937   \ifx\bib'label\@empty
2938     \begingroup
  We begin by saving the previous stem and initializing the current stem to the
  empty string.
2939     \global\let\previous@stem\current@stem
2940     \global\let\current@stem\@empty
2941     \global\let\previous@year\current@year
2942     \global\let\current@year\bib@year
  The list of primary contributors is available to us in \current@primary in the
  form
      \name{Last1, First1} \name{Last2, First2} ... \name{Lastn, Firstn}
  We will be executing this list multiple times with various definitions of \name.
  So the first thing we want to do is establish a safe environment and normalize
  the names.
2943     \@apply\auto@protect\amsrefs@textsymbols
2944     \@apply\auto@protect\amsrefs@textaccents
2945     \def\name##1{\@nx\name{\lncan@a##1, \@nil}}%
2946     \auto@protect\etaltext
2947     \normalize@edef\current@stem{\current@primary}%

```



```

2948         \xdef\current@stem{\current@stem}%
    At this point, the \current@stem is complete and we're ready to determine
    what (if any) suffix is needed to disambiguate it from the previous label.
2949         \calc@alpha@suffix
    We have all the pieces now. Arrange to end the current group and then define
    \bib@label in the enclosing group. (This keeps \bib@label from being defined
    outside the group started by \bib@start. This isn't strictly necessary, but it
    provides a bit of compartmentalization.)
2950         \edef\@tempa{%
2951             \def\@nx\cite@label{\current@stem}%
2952             \def\@nx\bib@label@year{%
2953                 \current@year
2954                 \alpha@label@suffix
2955             }%
2956         }
2957         \@xp\endgroup
2958         \@tempa
2959     \fi
2960 }

\lncan@a
2961 \def\lncan@a#1,#2\@nil{#1}

\citesel@author
2962 \def\citesel@author#1#2#3#4#5{\PrintCiteNames{#3}}

\citesel@authoryear
2963 \def\citesel@authoryear#1#2#3#4#5{\PrintCNY{#3}{#4}}

\citesel@object
2964 \def\citesel@object#1#2#3#4#5{\PrintCiteNames{#3} \citeleft#4}

\citesel
2965 \let\citesel\citesel@authoryear

\citeleft
2966 \def\citeleft{()%

\citeright
2967 \def\citeright{)}}%

\@citeleft
2968 \def\@citeleft{\ifx\citesel\citesel@object\else\citeleft\fi}%

\citepunct
2969 \def\citepunct{; }

```

`\BibLabel`

```
2970 \def\BibLabel{%
2971   \Hy@raisedlink{\hyper@anchorstart{cite.\CurrentBib}\relax\hyper@anchorend}%
2972 }
```

`\process@citelist`

```
2973 \let\process@citelist\process@citelist@unsorted
```

`\ycite`

```
2974 \DeclareRobustCommand{\ycite}[1]{%
2975   \star@{\cite@a\citesel@year{#1}}{}}%
2976 }
```

`\ycites`

```
2977 \DeclareRobustCommand{\ycites}[1]{%
2978   \begingroup
2979     \def\citepunct{, }%
2980     \let\citesel\citesel@year
2981     \cites{#1}%
2982   \endgroup
2983 }
```

`\citeyear`

```
2984 \DeclareRobustCommand{\citeyear}[1]{%
2985   \begingroup
2986     \let\citeleft\@empty
2987     \let\citeright\@empty
2988     \star@{\cite@a\citesel@year{#1}}{}}%
2989   \endgroup
2990 }
```

`\ocite`

```
2991 \DeclareRobustCommand{\ocite}[1]{%
2992   \star@{\cite@a\citesel@object{#1}}{}}%
2993 }
```

`\ocites`

```
2994 \DeclareRobustCommand{\ocites}[1]{%
2995   \begingroup
2996     \let\@citelist\@ocitelist
2997     \cites{#1}%
2998   \endgroup
2999 }
```

`\ocitelist`

```
3000 \def\@ocitelist#1{%
3001   \PrintSeries{\InnerCite}%
3002   {\ocite}%
3003   {}{ and \ocite}%
```

For three or more names: print ‘et al’ instead of the last name. Have to putz around with the space factor a bit or the comma between name and year will not be applied.

```

3004      {,}{ \ocite}%
3005      {,}{ and \ocite}%
3006      {}%
3007      {#1}%
3008      {}%
3009 }

\citeauthor
3010 \DeclareRobustCommand{\citeauthor}[1]{%
3011   \star@\cite@a\citesel@author{#1}}{}%
3012 }

\citeauthority
3013 \DeclareRobustCommand{\citeauthority}[1]{%
3014   \citeauthor{#1} \ycite{#1}%
3015 }

\fullcite
3016 \DeclareRobustCommand{\fullcite}[1]{%
3017   \begingroup
3018     \let\print@citenames\CiteNamesFull
3019     \star@\cite@a\citesel@authoryear{#1}}{}%
3020   \endgroup
3021 }

\fullocite
3022 \DeclareRobustCommand{\fullocite}[1]{%
3023   \begingroup
3024     \let\print@citenames\CiteNamesFull
3025     \star@\cite@a\citesel@object{#1}}{}%
3026   \endgroup
3027 }

      Invert the first author's name.
3028 \def\set@firstname#1{%
3029   \set@name{#1}\setbib@nameinverted
3030 }

\PrintCNY
3031 \def\PrintCNY#1#2{%
3032   \PrintCiteNames{#1}%
3033   \@ifnotempty{#2}{\@addpunct{,} #2}%
3034 }
```

`\PrintCiteNames`

```

3035 \def\PrintCiteNames#1{%
3036     \leavevmode
3037     \def\@tempa{#1}%
3038     \ifx\@tempa\prev@names
3039     \else
3040         \gdef\prev@names{#1}%
3041         \@xp\ifx\@car#1.\@nil\CitePrintUndefined
3042             #1\relax
3043     \else
3044         \print@citenames{#1}%
3045     \fi
3046 \fi
3047 }

```

`\CiteNames`

```

3048 \newcommand{\CiteNames}[1]{%
3049     \PrintSeries{\name}%
3050     {}%
3051     {}{ and }%

```

For three or more names: print ‘et al’ instead of the last name. Have to putz around with the space factor a bit or the comma between name and year will not be applied.

```

3052     {}{\@gobble}%
3053     {}{ \etaltext\@gobble}%
3054     {}%
3055     {#1}%
3056     {}%
3057 }

```

`\print@citenames`

```

3058 \let\print@citenames\CiteNames

```

`\CiteNamesFull`

```

3059 \newcommand{\CiteNamesFull}[1]{%
3060     \PrintSeries{\name}%
3061     {}%
3062     {}{ and }%

```

For three or more names: print ‘et al’ instead of the last name. Have to putz around with the space factor a bit or the comma between name and year will not be applied.

```

3063     {,}{ }%
3064     {,}{ and }%
3065     {}%
3066     {#1}%
3067     {}%
3068 }

```

`\PrintDate` No parentheses around the year.

```
3069 \renewcommand{\PrintDate}[1]{\bib@label@year}
```

`\print@date` Only print the year, not the month or day.

```
3070 \def\print@date{%
3071   \IfEmptyBibField{date}{%
3072     \IfEmptyBibField{year}{\BibField{status}}{\bib@year}%
3073   }{%
3074     \bib@year
3075   }%
3076 }

3077 \BibSpec{article}{%
3078   +{ } {\PrintAuthors}           {author}
3079   +{.} { \PrintDate}             {date}
3080   +{.} { \textit}                {title}
3081   +{.} { }                        {part}
3082   +{:} { \textit}                 {subtitle}
3083   +{,} { \PrintContributions}     {contribution}
3084   +{.} { \PrintPartials}          {partial}
3085   +{,} { }                        {journal}
3086   +{ } { \textbf}                 {volume}
3087   +{,} { \issuetext}              {number}
3088   +{,} { \eprintpages}            {pages}
3089   +{,} { }                        {status}
3090   +{,} { \PrintDOI}               {doi}
3091   +{,} { available at \eprint}    {eprint}
3092   +{ } { \parenthesize}           {language}
3093   +{ } { \PrintTranslation}        {translation}
3094   +{;} { \PrintReprint}           {reprint}
3095   +{.} { }                        {note}
3096   +{.} { }                        {transition}
3097   +{ } {\SentenceSpace \PrintReviews} {review}
3098 }
3099
3100 \BibSpec{book}{%
3101   +{ } {\PrintPrimary}            {transition}
3102   +{.} { \PrintDate}              {date}
3103   +{.} { \textit}                 {title}
3104   +{.} { }                        {part}
3105   +{:} { \textit}                 {subtitle}
3106   +{,} { \PrintEdition}            {edition}
3107   +{ } { \PrintEditorsB}           {editor}
3108   +{,} { \PrintTranslatorsC}      {translator}
3109   +{,} { \PrintContributions}     {contribution}
3110   +{,} { }                        {series}
3111   +{,} { \voltext}                 {volume}
3112   +{,} { }                        {publisher}
3113   +{,} { }                        {organization}
3114   +{,} { }                        {address}
```

```

3115   +{,} { }                {status}
3116   +{} { \parenthesize}    {language}
3117   +{} { \PrintTranslation} {translation}
3118   +{;} { \PrintReprint}    {reprint}
3119   +{.} { }                 {note}
3120   +{.} {}                  {transition}
3121   +{} {\SentenceSpace \PrintReviews} {review}
3122 }
3123
3124 \BibSpec{collection.article}{%
3125   +{} {\PrintAuthors}      {author}
3126   +{.} { \PrintDate}       {date}
3127   +{.} { \textit}          {title}
3128   +{.} { }                 {part}
3129   +{:} { \textit}          {subtitle}
3130   +{,} { \PrintContributions} {contribution}
3131   +{,} { \PrintConference}  {conference}
3132   +{} {\PrintBook}         {book}
3133   +{,} { }                 {booktitle}
3134   +{,} { pp.~}             {pages}
3135   +{,} { }                 {status}
3136   +{,} { \PrintDOI}        {doi}
3137   +{,} { available at \eprint} {eprint}
3138   +{} { \parenthesize}     {language}
3139   +{} { \PrintTranslation}  {translation}
3140   +{;} { \PrintReprint}    {reprint}
3141   +{.} { }                 {note}
3142   +{.} {}                  {transition}
3143   +{} {\SentenceSpace \PrintReviews} {review}
3144 }
3145
3146 \BibSpec{report}{%
3147   +{} {\PrintPrimary}      {transition}
3148   +{.} { \PrintDate}       {date}
3149   +{.} { \textit}          {title}
3150   +{.} { }                 {part}
3151   +{:} { \textit}          {subtitle}
3152   +{,} { \PrintEdition}    {edition}
3153   +{,} { \PrintContributions} {contribution}
3154   +{,} { Technical Report } {number}
3155   +{,} { }                 {series}
3156   +{,} { }                 {organization}
3157   +{,} { }                 {address}
3158   +{,} { \eprint}          {eprint}
3159   +{,} { }                 {status}
3160   +{} { \parenthesize}     {language}
3161   +{} { \PrintTranslation}  {translation}
3162   +{;} { \PrintReprint}    {reprint}
3163   +{.} { }                 {note}
3164   +{.} {}                  {transition}

```

```

3165   +{} {\SentenceSpace \PrintReviews} {review}
3166 }
3167
3168 \BibSpec{thesis}{%
3169   +{} {\PrintAuthors}           {author}
3170   +{.} { \PrintDate}           {date}
3171   +{.} { \textit}              {title}
3172   +{:} { \textit}              {subtitle}
3173   +{,} { \PrintThesisType}     {type}
3174   +{,} { }                     {organization}
3175   +{,} { }                     {address}
3176   +{,} { \eprint}              {eprint}
3177   +{,} { }                     {status}
3178   +{} { \parenthesize}         {language}
3179   +{} { \PrintTranslation}     {translation}
3180   +{;} { \PrintReprint}        {reprint}
3181   +{.} { }                     {note}
3182   +{.} {}                      {transition}
3183   +{} {\SentenceSpace \PrintReviews} {review}
3184 }
3185
3186 \BibSpec{webpage}{%
3187   +{} {\PrintAuthors}           {author}
3188   +{.} { \PrintDate}           {date}
3189   +{.} { \emph}                {title}
3190   +{:} { \emph}                {subtitle}
3191   +{,} { \url}                 {url}
3192   +{.} { Accessed \PrintDateField} {accessdate}
3193   +{.} { }                     {note}
3194   +{.} {}                      {transition}
3195 }
3196 \PopCatcodes
3197 </pkg>

```

## 6.27 The amsbst package

```

3198 (*bst)
3199 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3200 \ProvidesPackage{amsbst}[2013/01/16 v2.12]
3201 %\RequirePackage{amsrefs}[2004/03/29]
3202 \BibSpec{article}{%
3203   +{} {\PrintAuthors}           {author}
3204   +{.} { }                     {title}
3205   +{.} { }                     {part}
3206   +{:} { }                     {subtitle}
3207   +{.} { \PrintContributions}  {contribution}
3208   +{.} { \PrintPartials}       {partial}
3209   +{.} { \emph}                {journal}
3210   +{} { }                     {volume}
3211   +{} { \parenthesize}         {number}

```

```

3212   +{:} {}                               {pages}
3213   +{,} { \PrintDateB}                   {date}
3214   +{,} { }                               {status}
3215   +{.} { \PrintTranslation}              {translation}
3216   +{.} { Reprinted in \PrintReprint}    {reprint}
3217   +{.} { }                               {note}
3218   +{.} {}                               {transition}
3219 }
3220
3221 \BibSpec{partial}{%
3222   +{} {}                                  {part}
3223   +{:} { }                               {subtitle}
3224   +{.} { \PrintContributions}            {contribution}
3225   +{.} { \emph}                          {journal}
3226   +{} { }                                 {volume}
3227   +{} { \parenthesize}                   {number}
3228   +{:} {}                                 {pages}
3229   +{,} { \PrintDateB}                   {date}
3230 }
3231
3232 \BibSpec{book}{%
3233   +{} { \PrintPrimary}                   {transition}
3234   +{.} { \emph}                          {title}
3235   +{.} { }                               {part}
3236   +{:} { \emph}                          {subtitle}
3237   +{.} { }                               {series}
3238   +{,} { \voltext}                       {volume}
3239   +{.} { Edited by \PrintNameList}       {editor}
3240   +{.} { Translated by \PrintNameList}   {translator}
3241   +{.} { \PrintContributions}            {contribution}
3242   +{.} { }                               {publisher}
3243   +{.} { }                               {organization}
3244   +{,} { }                               {address}
3245   +{,} { \PrintEdition}                  {edition}
3246   +{,} { \PrintDateB}                    {date}
3247   +{.} { }                               {note}
3248   +{.} {}                               {transition}
3249   +{.} { \PrintTranslation}              {translation}
3250   +{.} { Reprinted in \PrintReprint}    {reprint}
3251   +{.} {}                               {transition}
3252 }
3253
3254 \BibSpec{collection.article}{%
3255   +{} { \PrintAuthors}                   {author}
3256   +{.} { }                               {title}
3257   +{.} { }                               {part}
3258   +{:} { }                               {subtitle}
3259   +{.} { \PrintContributions}            {contribution}
3260   +{.} { \PrintConference}               {conference}
3261   +{.} { \PrintBook}                     {book}

```



```

3262   +{.} { In }                               {booktitle}
3263   +{,} { pages~}                             {pages}
3264   +{.} { \PrintDateB}                       {date}
3265   +{.} { \PrintTranslation}                 {translation}
3266   +{.} { Reprinted in \PrintReprint}       {reprint}
3267   +{.} { }                                   {note}
3268   +{.} {}                                    {transition}
3269 }
3270
3271 \BibSpec{conference}{%
3272   +{} {}                                     {title}
3273   +{} {\PrintConferenceDetails} {transition}
3274 }
3275
3276 \BibSpec{innerbook}{%
3277   +{.} { \emph}                             {title}
3278   +{.} { }                                   {part}
3279   +{:} { \emph}                             {subtitle}
3280   +{.} { }                                   {series}
3281   +{,} { \voltext}                          {volume}
3282   +{.} { Edited by \PrintNameList}          {editor}
3283   +{.} { Translated by \PrintNameList}      {translator}
3284   +{.} { \PrintContributions}              {contribution}
3285   +{.} { }                                   {publisher}
3286   +{.} { }                                   {organization}
3287   +{,} { }                                   {address}
3288   +{,} { \PrintEdition}                    {edition}
3289   +{,} { \PrintDateB}                      {date}
3290   +{.} { }                                   {note}
3291   +{.} {}                                    {transition}
3292 }
3293
3294 \BibSpec{report}{%
3295   +{} {\PrintPrimary}                       {transition}
3296   +{.} { \emph}                             {title}
3297   +{.} { }                                   {part}
3298   +{:} { \emph}                             {subtitle}
3299   +{.} { \PrintContributions}              {contribution}
3300   +{.} { Technical Report }                {number}
3301   +{,} { }                                   {series}
3302   +{.} { }                                   {organization}
3303   +{,} { }                                   {address}
3304   +{,} { \PrintDateB}                      {date}
3305   +{.} { \PrintTranslation}                 {translation}
3306   +{.} { Reprinted in \PrintReprint}       {reprint}
3307   +{.} { }                                   {note}
3308   +{.} {}                                    {transition}
3309 }
3310
3311 \BibSpec{thesis}{%

```

```

3312   +{ } {\PrintAuthors}           {author}
3313   +{,} { \emph}                  {title}
3314   +{:} { \emph}                   {subtitle}
3315   +{.} { \PrintThesisType}        {type}
3316   +{.} { }                         {organization}
3317   +{,} { }                         {address}
3318   +{,} { \PrintDateB}              {date}
3319   +{.} { \PrintTranslation}        {translation}
3320   +{.} { Reprinted in \PrintReprint} {reprint}
3321   +{.} { }                         {note}
3322   +{.} { }                         {transition}
3323 }
3324
3325 \BibSpec{webpage}{%
3326   +{ } {\PrintAuthors}           {author}
3327   +{.} { }                         {title}
3328   +{:} { }                         {subtitle}
3329   +{.} { \PrintDateB}              {date}
3330   +{.} { \url}                     {url}
3331   +{.} { Accessed \PrintDateField} {accessdate}
3332   +{.} { }                         {note}
3333   +{.} { }                         {transition}
3334 }

```

`\PrintEditorsA` When we consider editor names we have to think about some further complications. First, for the case of a book where editor names are listed in place of author names, just copy the same style with a bit of added text at the end.

```

3335 \renewcommand{\PrintEditorsA}[1]{%
3336   \def\current@bibfield{\bib'editor}%
3337   \PrintNames}{, editor\Plural{s}}{#1}%
3338   \erase@field\bib'editor
3339 }

```

`\PrintTranslatorsA`

```

3340 \renewcommand{\PrintTranslatorsA}[1]{%
3341   \def\current@bibfield{\bib'translator}%
3342   \PrintNames}{, translator\Plural{s}}{#1}%
3343   \erase@field\bib'translator
3344 }

```

```
3345 </bst>
```

The usual `\endinput` to ensure that random garbage at the end of the file doesn't get copied by `docstrip`.

```
3346 \endinput
```

## References

- [1] David M. Jones, *User's Guide to the amsrefs Package*. distributed with the `amsrefs` code.

- [2] Ellen Swanson, Arlene O'Sean, and Antoinette Schleyer, *Mathematics into Type*, updated, American Mathematical Society, 1999.

## 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.

| <b>Symbols</b>    |                                     |                                                           |
|-------------------|-------------------------------------|-----------------------------------------------------------|
| \"                | 69, 1786                            | \@ifnotempty <i>45</i> , 239,                             |
| \&                | 25                                  | 717, 741, 1392,                                           |
| \'                | 40, <i>69</i> , 1787                | 1513, 2213, 3033                                          |
| \+                | 24                                  | \@ifpackageloaded                                         |
| \,                | 24, 119, 2553                       | ..... 1763,                                               |
| \-                | 24                                  | 1766, 1772, 2806                                          |
| \.                | 24, 114, 1788, 2786                 | \@ifpackagewith ..                                        |
| .aux              | 9–11, <i>53</i> , 64–67             | ..... 5, 8, 1773                                          |
| .bb1              | 9, <i>31</i> , <i>32</i> , 50       | \@ifsame@patterns                                         |
| .bib              | 9                                   | ..... 1045, <u>1051</u>                                   |
| .brf              | 109                                 | \@ifsamepat . 1052, <u>1054</u>                           |
| .dtx              | 8                                   | \@input@ ..... 1280                                       |
| .ltx              | 31, <i>32</i> , 49                  | \@lappend .....                                           |
| \:                | 24, 117, 684, 700                   | <u>61</u> , 1784, 1813,                                   |
| \;                | 24, 118                             | 1814, 1816, 1817                                          |
| \=                | 24, 1789                            | \@latex@warning ..                                        |
| ? option          | 18                                  | ..... 573, 1499                                           |
| \@@undefined      | 1484                                | \@lbibitem ... 27, 541                                    |
| \@MM              | 2355, 2382, 2404                    | \@let@token .... 78, 85                                   |
| \@addpunct        | 7, <i>15</i> ,                      | \@listctr ... 532, 546                                    |
|                   | <i>16</i> , <u>107</u> , 1447, 3033 | \@m ..... 110, 551, 605                                   |
| \@alph            | 1084, 2271                          | \@marknames . <u>2212</u> , 2224                          |
| \@apply           | 68                                  | \@maxlabelwidth 544,                                      |
| \@arabic          | 1092, 2172                          | <u>577</u> , 580, 581, 589                                |
| \@bibdef          | <i>31</i> , <i>32</i> , 632,        | \@mkboth .... 48, 1204                                    |
|                   | 646, <u>650</u> , 662,              | \@multiauthorlabel                                        |
|                   | 1260, 1263, 1279                    | . 2202, 2206, <u>2242</u>                                 |
| \@bibitem         | 27, 540                             | \@nametoken .. <u>2210</u> ,                              |
| \@biblist         | 501, 506, <u>536</u>                | 2218, 2230, 2232                                          |
| \@biblistsetup    | ...                                 | \@nameuse ..... 72                                        |
|                   | ... 556, <u>558</u> , <u>2921</u>   | \@nil ..... 14,                                           |
| \@bibtitlestyle   | ...                                 | <i>43</i> , <i>74</i> , <i>76</i> , <i>81</i> , <i>82</i> |
|                   | ..... 1222, 1225                    | \@nilgobble .....                                         |
| \@car             | 2208, 2395, 3041                    | 76, 82, 1923, 2223                                        |
| \@carcube         | 2209                                | \@nmbrrlistfalse ..                                       |
| \@chomp           | <u>85</u> , 1981,                   | .. 518, 1086, 1094                                        |
|                   | 1993, 2046, 2047                    | \@nmbrrlisttrue 545, 1078                                 |
| \@chomper         | 15, 88, 102                         | \@noitemerr ..... 573                                     |
| \@citeleft        | .....                               | \@nopunctsfcodes ..                                       |
|                   | 1511, <u>1519</u> , <u>2968</u>     | ... 122, <u>123</u> , 2397                                |
|                   |                                     | \@ocitelist . 2996, 3000                                  |
| \@citelist        | .... 56,                            |                                                           |
|                   | 1434, <u>1435</u> , 2996            |                                                           |
| \@citestyle       | 1438, <u>1455</u>                   |                                                           |
| \@citex           | ..... 58                            |                                                           |
| \@close@bbl@file  | ..... 1276,                         |                                                           |
|                   | 1348, 1357, 1367                    |                                                           |
| \@concat          | 59, 2173                            |                                                           |
| \@count@chars     | 2155, <u>2160</u>                   |                                                           |
| \@currentlabel    | ...                                 |                                                           |
|                   | ..... 1066, 1136                    |                                                           |
| \@currrent        | ..... 135                           |                                                           |
| \@currname        | ..... 137                           |                                                           |
| \@empty           | 15, 20, 22, <i>51</i> , 52          |                                                           |
| \@endbiblist      | .....                               |                                                           |
|                   | ..... 503, 508, <u>571</u>          |                                                           |
| \@extract@initial | ..... 85,                           |                                                           |
|                   | 2121, 2123, <u>2135</u>             |                                                           |
| \@file@und        | 1293,                               |                                                           |
|                   | 1296, 1299, 1313                    |                                                           |
| \@firstofone      | 15,                                 |                                                           |
|                   | <i>43</i> , 1042, 1411,             |                                                           |
|                   | 1433, 1476,                         |                                                           |
|                   | 1764, 1770, 2281                    |                                                           |
| \@firstoftwo      | .....                               |                                                           |
|                   | ... 48, 1056, 2338                  |                                                           |
| \@firstone        | <u>2208</u> , 2210                  |                                                           |
| \@firstthree      | <u>2209</u> , 2232                  |                                                           |
| \@for             | ..... 39                            |                                                           |
| \@gobblefour      | ..... 752                           |                                                           |
| \@gobblethree     | ... 1529                            |                                                           |
| \@ifclassloaded   | ..                                  |                                                           |
|                   | ..... 1200–1202                     |                                                           |
| \@ifempty         | ..... <u>45</u> ,                   |                                                           |
|                   | 268, 1152, 1380,                    |                                                           |
|                   | 1405, 1491,                         |                                                           |
|                   | 1664, 1670,                         |                                                           |
|                   | 1686, 1738, 2716                    |                                                           |
| \@ifnextchar      | ..... 20                            |                                                           |

- \oneauthorlabel . . . . . 2200, 2227
  - \open@bbl@file 675, 1273, 1347, 1363
  - \@pass@ptions . . . . . 134
  - \@prev@cite . . . . . 1641
  - \@secondoftwo . . . . . 48, 1058, 2340
  - \@sptoken . . . . . 1935, 2010, 2060, 2108
  - \@starttoc . . . . . 109
  - \@suffix@format . . . . . 1084, 1092, 2271, 2291, 2302
  - \@tabackkludge . 72, 1843, 1855, 1874
  - \@tempa . . . . . 77
  - \@tempcnta . . . . . 14, 64
  - \@tempcntb . . . . . 64
  - \@tempd . . . . . 2905, 2911
  - \@tempdima . . . . . 579–581
  - \@tempe 2906, 2912, 2914
  - \@tempswafalse . . . . . 635, 1127, 2273
  - \@tempswatrue . 631, 1145, 1148, 2277
  - \@temptokena . . . . . 77
  - \@testdef . . . . . 47
  - \@threeauthors . . . . . 2238, 2252
  - \@uclclist . . . . . 71
  - \@urlbreak . . . . . 25, 27
  - \@urlfix . . . . . 24, 32
  - \@urlsetup . . . . . 20, 22
  - \@xifempty . . . . . 45, 47
  - \^ . . . . . 106, 1790
  - \‘ . . . . . 1791
  - \~ . . . . . 29, 1792
  - \\_ . . . . . 1432, 2756, 2914
- A**
- \A . . . . . 603
  - abbrv BIB<sub>T</sub>E<sub>X</sub> style . . 16
  - \add@toks@ . . . . . 60
  - address field 4, 49, 110
  - \addto@defer@list . . . . . 763, 767
  - \addto@hook . . . . . 60, 880
  - \AE . . . . . 1822
  - \ae . . . . . 71, 1822
  - \after@deleting@token . . . . . 41, 1942, 1952, 2020, 2058, 2074, 2117
  - \afterassignment 13, 42, 82, 774, 2746
  - \aftergroup . . . . . 22, 291
  - alpha BIB<sub>T</sub>E<sub>X</sub> style . . 16
  - \alpha@label . . . . . 1067, 1076, 1081, 1089, 1098
  - \alpha@label@ . . . . . 1068, 1081, 1089
  - \alpha@label@suffix . . . . . 2191, 2267, 2291, 2298, 2302, 2954
  - \alpha@sufffix 2269, 2270, 2290–2292, 2299, 2302
  - alphabetic option . . . . . 16, 87, 90, 108
  - amsart class . . . . . 109
  - amsbook class . . . . . 55
  - amsbst package . . . . . 119
  - amsgen package . . . . . 15
  - amsproc class . . . . . 55
  - amsrdoc.tex . . . . . 3
  - amsrefs package . . . . . 1, 3–6, 8, 10, 18, 27, 49, 66, 94, 108, 112, 122
  - \amsrefs@@lbibitem 514
  - \amsrefs@bibcite . . . . . 1709, 1714, 1718
  - \amsrefs@bibitem . . . . . 514, 540
  - \amsrefs@error . . . . . 105, 258, 308, 655
  - \amsrefs@label@prefix . . . . . 561, 2171, 2172, 2178
  - \amsrefs@lbibitem . . . . . 514, 541
  - \amsrefs@option@alphabeticauth field . . . . . 1080, 2798
  - \amsrefs@option@numeric . . . . . 1075
  - \amsrefs@option@shortalphabetic . . . . . 1088, 2803
  - \amsrefs@textaccents . . . . . 70, 910, 1126, 1782, 1784, 1842, 1856, 1873, 1974, 2180, 2944
  - \amsrefs@textsymbols . . . . . 72, 909, 1125, 1811, 1813, 1814, 1816, 1817, 1841, 1857, 1871, 1973, 2179, 2943
  - \amsrefs@warning . . . . . 104, 565, 953, 954, 1159, 1173, 1178, 1284, 1310, 1388, 1393, 1398, 1581, 1967, 2039, 2467, 2924, 2930
  - \amsrefs@warning@nl . . . . . 4, 6, 9, 157, 167, 2821, 2892
  - amsxport BIB<sub>T</sub>E<sub>X</sub> style . . . . . 86, 87
  - \append@label@year . . . . . 1085, 1093, 2186
  - \append@label@year@ . . . . . 1085, 2257
  - \append@to@stem . . . . . 2173, 2234, 2240, 2250, 2260
  - \ar@hyperlink . . . . . 1512, 1758, 1767
  - \ar@SK@cite . . . . . 1510, 1761, 1774, 1777
  - article entry type . 5, 7
  - \AtBeginDocument . . . . . 109, 650, 651, 1710, 1762, 2790, 2805
  - \AtEndDocument . . . . . 1196
  - author-year option . . . . . 56, 90, 97, 112
  - \AuthorList . . . . . 8

- `\auto@protect` .. 13,  
   57, 73, 909,  
   910, 1107,  
   1124–1126,  
   1841, 1856,  
   1864, 1871,  
   1970, 1971,  
   1973, 1974,  
   2179–2182,  
   2943, 2944, 2946
- B**
- `\b` ..... 1793  
`\b@` 45, 58, 59, 62, 66, 67  
`\b@*` ..... 1571  
`\b@moo` ..... 9, 10, 53  
`\b@whatever` . 10, 11, 59  
 babel package ..... 43  
`\back@cite` . 1746, 2808  
`\BackCite` ... 1464,  
   1543, 1745, 2808  
 backref package 108, 109  
 backrefs option 10, 108  
`\bbl@copy` ... 680, 734  
`\bbl@copy@a` .. 738, 740  
`\bbl@out` .... 1345,  
   1355, 1356,  
   1358, 1366, 1369  
`\bbl@write` .....  
   677, 681, 744,  
   1346, 1358, 1368  
`\bblname` 1267, 1289, 1356  
`\begin` ..... 9, 10  
`\bgroup` .....  
   89, 1892, 1960,  
   2023, 2077, 2120  
`\bi@` ..... 35  
`\bib` .... 3–5, 8, 11,  
   27, 30, 43, 49,  
   51–53, 57, 72, 628  
`\bib'` ..... 39  
`\bib'bar` ..... 21  
`\bib'date` ..... 42  
`\bib'given` .... 75, 76  
`\bib'initials` .... 76  
`\bib'jr` ..... 76  
`\bib'language` ..... 42  
`\bib'pages` ..... 21  
`\bib'surname` ..... 76  
`\bib*` ..... 31, 32
- `\bib@append` .... 21,  
   22, 246, 259, 2582  
`\bib@author` ..... 45  
`\bib@bar` ..... 21  
`\bib@cite` 45, 786, 1096  
`\bib@cite@a` . 1101, 1104  
`\bib@cite@check` ..  
   ..... 1116, 1119  
`\bib@citeyear` 2259, 2263  
`\bib@day` .... 1013,  
   1020, 1021, 2733  
`\bib@dbfile` .....  
   .. 1290, 1293,  
   1296, 1299,  
   1318, 1321, 1324  
`\bib@div@mark` ....  
   . 1199, 1214, 1227  
`\bib@end` .. 37, 803, 820  
`\bib@exec` .. 12, 32,  
   35, 651, 659, 664  
`\bib@field@patches`  
   .... 782, 809,  
   964, 2565, 2579  
`\bib@label` ... 88, 113  
`\bib@label@year` ..  
   .. 1110, 1143,  
   1157, 1164,  
   1171, 2952, 3069  
`\bib@language` ....  
   .... 42, 1034,  
   1036, 1039,  
   1041, 1042,  
   1045–1047, 1061  
`\bib@language@fixup`  
   ..... 1009, 1031  
`\bib@mklab` .. 547, 578  
`\bib@month` .. 1012,  
   1019, 1022,  
   2724, 2737, 2880  
`\bib@monthname` ...  
   . 2732, 2736, 2879  
`\bib@parsedate` 991,  
   994, 1014, 2718  
`\bib@parsedate@a` .  
   ..... 1015, 1017  
`\bib@print` .....  
   .. 32, 36, 651, 777  
`\bib@print@inner` .  
   ... 805, 2511,  
   2530, 2556, 2610
- `\bib@reprint` 2559, 2561  
`\bib@reset` ..... 914  
`\bib@resolve@xrefs`  
   .... 781, 808,  
   828, 2564, 2578  
`\bib@selectlanguage`  
   ..... 783, 1044  
`\bib@start` .....  
   . 88, 113, 778, 816  
`\bib@store` 32, 664, 773  
`\bib@translation` .  
   ..... 2573, 2575  
`\bib@year` ..... 45,  
   989, 998, 1011,  
   1018, 1023,  
   2258, 2261,  
   2728, 2743,  
   2942, 3072, 3074
- `\BibAbbrevWarning`  
   .... 862, 897, 953  
 bibchapter environ-  
   ment ..... 1209  
`\bibchapter` ..... 1234  
`\bibcite` ..... 9–  
   11, 47, 65, 66,  
   522, 532, 1194,  
   1196, 1714, 1718  
`\bibcite@a` . 1709, 1720  
`\bibcite@b` . 1725, 1729  
`\bibcite@write` ...  
   ..... 1102, 1184  
`\bibdata` ..... 9  
`\bibdefer@list` 766,  
   770, 1277, 1278  
 bibdiv environment 1231  
`\bibdiv` ..... 1237  
`\BibField` . 8, 184, 3072  
`\BibItem` . 633, 636, 639  
`\bibitem` .... 9, 10, 27  
`\BibLabel` .....  
   542, 613, 619, 2970  
`\biblanguagedefault`  
   ..... 1034, 1061  
`\biblanguageEnglish`  
   ..... 1061  
`\bibliography` ..... 9  
`\bibliographystyle`  
   .... 9, 108, 2788  
 biblist environment  
   ..... 1, 27, 499

- \biblist . 109, 561, 564, 565, 567, 1238
  - biblist\* environment 505
  - \biblistfont . 510, 538
  - \bibname 48, 1197, 1209
  - \bibquotes 3, 6, 7, 2631
  - bibsection environ-  
ment . . . . . 1218
  - \bibsection . . . . . 1232
  - \BibSelect . . . . .  
. 1261, 1264, 1267
  - \bibselect . . . . .  
31, 34, 35, 60,  
1258, 1285, 1286
  - \bibselect\* . . . . . 32, 60
  - \bibselect@msg . . 1270
  - \BibSpec 4–6, 8, 20, 237
  - \biblespec@scan . . . .  
. . . . . 20, 240, 245
  - \biblespec@scan@error  
. . . . . 254, 258
  - \BibSpecAlias . . . .  
. 23, 305, 473–481
  - \bibstyle . . . . . 9, 2792
  - bibtex-style option 111
  - \bibtex@style 2789,  
2792, 2796,  
2801, 2826, 2890
  - book class . . . . . 49
  - \bracket@stack 2616,  
2619, 2624, 2629
  - bst . . . . . 3
  - \bud . . . . . 1836
  - \bysame . . 8, 2477, 2478
- C**
- \c . . . . . 1794
  - \c@bib . . . . . 2172
  - \c@bib@env 589, 594, 597
  - \calc@alpha@suffix  
. 2187, 2272, 2949
  - \calc@author@part  
. 1083, 1091, 2185
  - \calc@author@part@  
. . . . . 1083, 2198
  - \calc@author@part@short  
. . . . . 1091, 2205
  - \CardinalNumber . 2763
  - \CardinalNumeric .  
. . . . . 8, 2752, 2763
  - \catcode . . . . . 30
  - \cdprime . . . . . 1835
  - \ch@ck . . . . . 1360
  - \chapter\* . . . . . 48
  - \check@mathfonts . . 23
  - \chomp@ . . . . . 95, 100
  - \citation . . 9–11, 1522
  - citation-order op-  
tion . . . . . 87, 109
  - cite package . . . . . 55
  - \cite . . . . . 9–11,  
45, 53, 55, 59, 1378
  - \cite@a . . . . . 56,  
1462, 1463,  
1569, 2975,  
2988, 2992,  
3011, 3019, 3025
  - \cite@b . 56, 1473, 1480
  - \cite@bc 57, 1481, 1483
  - \cite@begingroup .  
. . 1440, 1456,  
1465, 1542, 1549
  - \cite@cb . . . . . 1509,  
1520, 1558, 1645
  - \cite@cj 57, 63, 64, 1506
  - \cite@cjs . . 1593, 1654
  - \cite@compress . . .  
. . 1594, 1662,  
1682, 1706, 2857
  - \cite@compress@a .  
. 1666, 1669, 1699
  - \cite@compress@b .  
. . . . . 1675, 1685
  - \cite@dash 64, 1603,  
1657, 1660,  
1690, 1694, 1702
  - \cite@dest . 1730, 1731
  - \cite@dest@b 1735, 1737
  - \cite@endgroup 1445,  
1457, 1517,  
1544, 1550, 1564
  - \cite@label . . . . .  
. . 1066, 1110,  
1136, 1137,  
1139, 1154,  
1157, 1164, 2951
  - \cite@nobib@test .  
. . . . . 1486, 1490
  - \cite@print . . 1644,  
1651, 1663,  
1674, 1677,  
1681, 1692,  
1697, 1705, 2857
  - \cite@single 1379, 1403
  - \cite@sort . 1597, 1607
  - \cite@sort@a 1611, 1614
  - \cite@sort@b 1620, 1630
  - \cite@sorted@e . . .  
. . . 63, 1574, 1600
  - \cite@sorted@s . . .  
. . . 63, 1574, 1590
  - \CiteAltPunct . . . . . 55
  - \citeAltPunct 1432, 1443
  - \citeauthor . . . . .  
. 1538, 3010, 3014
  - \citeauthorhy 1539, 3013
  - \citedest . . . . .  
. 10, 66, 67, 1730
  - \citeform . . 1433, 1757
  - \citeleft . . . 1428,  
1439, 1447,  
1469, 1519,  
1552, 2964,  
2966, 2968, 2986
  - \citelist . . . . .  
59, 1393, 1414,  
1434, 1516, 1545
  - \citimid 55, 1428, 1513
  - \citen . . . . . 1531
  - \CiteNames . 3048, 3058
  - \CiteNamesFull . . .  
. 3018, 3024, 3059
  - \CitePrintUndefined  
. . 1128, 1495,  
1502, 1505,  
1592, 1653, 3041
  - \citepunct 55, 1428,  
1439, 1443,  
1447, 2969, 2979
  - \citeright . . 1428,  
1448, 1452,  
1470, 1514,  
1553, 2967, 2987
  - \cites . . 1389, 1530,  
1533, 1535,  
1546, 2981, 2997
  - \cites@a . . . . 1384,  
1405, 1406, 1530
  - \cites@b . . . 1408, 1417
  - \cites@c . . . 1421, 1423

- \cites@init .. 1446, 295, 2473, 2490, 1458, 1466, 1551
  - \cites@o ... 1381, 1404
  - \cites@oo .. 1404, 1405
  - \CITSEL ..... 57
  - \citesel ..... 45, 53, 57, 59, 66, 67, 1376, 2965
  - \citesel@author .. 1468, 2962, 3011
  - \citesel@authoryear .. 2963, 2965, 3019
  - \citesel@number .. 1374, 1376, 1573
  - \citesel@object .. 2968, 2992, 3025
  - \citesel@other ... 1569, 1570
  - \citesel@update .. 1371, 1525
  - \citesel@write ... 1187, 1192
  - \citesel@year 1375, 2975, 2980, 2988
  - \citeyear ..... 2984
  - \CloseBBLFile 1357, 1365
  - \closein ..... 1318
  - \clubpenalty . 552, 553
  - cmr font ..... 56
  - cmsy font ..... 7
  - collection entry type 7
  - \compare@stems ... 2274, 2307
  - \copy@bibdef ..... 31, 60, 667, 757, 1260, 1561
  - \copy@bibdef@a ... 671, 674, 720
  - \count ..... 14, 1352
  - \count@ ..... 107
  - \count@chars 2143, 2152
  - \cprime ..... 1834
  - \CPU@normal . 1505, 1653
  - \CPU@sort .. 1589, 1592
  - \cs ..... 72, 73
  - \CurLine 51, 52, 1324, 1325, 1331, 1342
  - \current@bibfield ..... 22, 267, 295, 2473, 2490, 2597, 3336, 3341
  - \current@citekey . 814, 818, 825, 2300
  - \current@primary 87, 98, 112, 955, 961, 967, 970, 974, 2183, 2460, 2463, 2470, 2487, 2594, 2947
  - \current@stem .. 88, 113, 1179, 2168, 2173, 2177, 2178, 2190, 2274, 2282, 2283, 2939, 2940, 2947, 2948, 2951
  - \current@year .... 90, 2170, 2276, 2941, 2942, 2953
  - \CurrentBib ... 516, 529, 612, 615, 822, 1070, 1097, 1175, 1194, 2971
  - \CurrentBibType 32, 653
  - \cydot ..... 1837
- ### D
- \d ..... 1795
  - \DashPages ... 7, 2673
  - \dateLANGUAGE .... 43
  - \Dbar ..... 1831
  - \dbar ..... 1831
  - \DeclareBooleanOption 127–132, 142–147
  - \DeclareExclusiveOptions 125, 126, 139–141
  - \DeclareNameAccent . 1783, 1786–1810
  - \DeclareNameSymbol . 1812, 1820–1838
  - \DeclareOption ... 133
  - \DeclareRobustCommand ... 18, 1434, 1502, 1530, 2974, 2977, 2984, 2991, 2994, 3010, 3013, 3016, 3022
  - \def ..... 23, 58, 59
  - \defer@bibdef .... 669, 759, 1315
  - \deferredquotes .. 2403, 2633, 2634, 2850, 2852
  - \deferredquoteslogical ..... 2392, 2851
  - \DefineAdditiveKey ..... 188–195
  - \DefineDummyKey .. 234
  - \DefineJournal 4, 17, 38, 49, 1246, 2863
  - \DefineName ..... 4, 37, 49, 52, 1243, 2605, 2606
  - \DefinePublisher 17, 38, 49, 1252, 2871
  - \DefineSimpleKey . 196–233, 235, 236, 497, 498
  - \DH ..... 1825
  - \dh ..... 1825
  - \DJ ..... 1826
  - \dj ..... 1826
  - \do ..... 14, 19, 69
  - docstrip ..... 122
  - \dospecials ..... 14
  - \DSK@append ..... 912
  - \DSK@def ..... 911
  - \dudot ..... 1804
  - \DuplicateBibKeyWarning . 1121, 1172, 1177
  - \DuplicateBibLabelWarning ..... 1177, 2285
- ### E
- \edef ..... 72
  - \editiontext ..... 7, 8, 2756, 2762
  - editor field ..... 22
  - \emph 464, 465, 3189, 3190, 3209, 3225, 3234, 3236, 3277, 3279, 3296, 3298, 3313, 3314
  - \empty@cite . 1485, 1489
  - \EmptyNameWarning ..... 1917, 1967

- `\EmptyPrimaryWarning` ..... 2461, [2466](#)  
`\endbibchapter` .. 1234  
`\endbibdiv` ..... 1241  
`\endbiblist` ..... 1240  
`\endbibsection` .. 1232  
`\endinput` .... 52, [122](#)  
`\endlist` ..... 575  
environments:  
  `bibchapter` ... [1209](#)  
  `bibdiv` ..... [1231](#)  
  `biblist` . 1, [27](#), [499](#)  
  `biblist*` ..... [505](#)  
  `bibsection` ... [1218](#)  
  `tabbing` ..... [72](#)  
  `thebibliography`  
  ..... [1236](#)  
`\eprint` 329, 391, 434,  
  452, [2787](#), 3091,  
  3137, 3158, 3176  
`\eprintpages` .. 326,  
  346, [2680](#), 3088  
`\erase@field` .....  
  [22](#), [290](#), 2493,  
  2600, 3338, 3343  
`\errmessage` ..... 263  
`\etalchar` ..... 56,  
  70, 72, 1809,  
  2240, 2245, [2255](#)  
`\etaltext` ..... 70,  
  1107, 1124,  
  1810, 2182,  
  2244, 2349,  
  2605, 2606,  
  [2607](#), 2946, 3053  
`\ExecuteOptions` .. 148  
export-bibtex pack-  
  age ..... [32](#)  
`\extr@cite` .. 1159,  
  1495, 1500,  
  1522, [1529](#),  
  1748, 1759, 1761  
`\extract@initial` .  
  ..... 2095,  
  [2100](#), 2117, 2132  
`\extract@initial@a`  
  ... [81](#), 2102, [2104](#)  
`\extract@initial@b`  
  ..... 2109, [2116](#)  
`\extract@initial@c`  
  ..... 2111, [2119](#)  
`\extract@initial@d`  
  ..... 2126, [2128](#)  
`\extract@initials`  
  . 1921, [1968](#), 2848  
`\extract@surnames`  
  . [2222](#), 2229, 2248
- F**
- `\fld@elt` ..... [186](#),  
  191–195, 2500,  
  2658, 2662, 2669  
`\font` ..... 23  
`\fontseries` ..... 1503  
`\fontshape` ..... 1503  
`\foo` ..... [72](#)  
`\format@cite` .....  
  . [1757](#), 1759, 1767  
`\frenchspacing` ...  
  . [95](#), [108](#), 113, 554  
`\fsa@n` ..... [81](#)  
`\fullcite` .. [1536](#), [3016](#)  
`\fullocite` . [1537](#), [3022](#)  
`\futurelet` ..... [76](#)
- G**
- `\G@refundefinedtrue`  
  ..... 1492  
`\g@undef` . 58, 724, 1277  
`\generate@alphalabel`  
  . 1082, 1090, [2174](#)  
`\generate@label` 784,  
  [1065](#), 1077,  
  1082, 1090, [2936](#)  
`\get@current@properties`  
  ... 276, [293](#), 2449  
`\get@namepart` 76, 77,  
  81, 1908, 1911,  
  1914, [1925](#), 2223  
`\get@namepart@a` ..  
  ..... 77, [81](#),  
  1928, [1930](#), 1942  
`\get@namepart@b` ..  
  77, [81](#), 1932, [1934](#)  
`\get@namepart@c` ..  
  ... [82](#), 1936, [1941](#)  
`\get@namepart@d` ..  
  ... 76, 1938, [1944](#)
- H**
- `\H` ..... 1796  
`har.bst` ..... [9](#)  
`\hbox` ..... 2479  
`\hfil` ..... 583  
`\hfill` ..... 614  
`hij.bib` ..... [9](#)  
`\href` ..... 2911, 2914  
`\hrulefill` ..... 2479  
`\Hy@backout` ..... 1750  
`\Hy@raisedlink` ...  
  ... 615, 1770, 2971  
`\hyper@link` .... 1759  
`\hyper@anchorend` .  
  ... 615, 1769, 2971  
`\hyper@anchorstart`  
  ... 615, 1768, 2971  
hyperref package [66](#), [68](#)  
`\hyph@to@space` ...  
  ..... [2211](#), 2224  
`\hyphenpenalty` .... 30
- I**
- `\i` .. 71, [73](#), 1820, 1875  
`\if@tempwa` ... 668,  
  678, 1151, 2289  
`\ifBR@verbose` [1744](#), 1747  
`\IfEmptyBibField` .  
  ..... [8](#), [20](#),  
  [185](#), 2413, 2567,  
  2583, 2584,  
  2671, 2681,  
  2702, 2705,  
  2832, 3071, 3072  
`\IfFileExists` ....  
  . 11, 154, 164,  
  1292, 1295, 1298  
`\ifin@` ..... 730,  
  934, 1129, 1442,  
  2140, 2508, 2519  
ifoption package ... [16](#)



- `\ifsame@stems` 2275, [2306](#)  
`\ifx` ..... 13  
`\in@` ..... 729,  
     933, 1128, 1441,  
     2139, 2507, 2518  
`\ini@hyphen` .....  
     .. 1970, [1987](#),  
     1993, 2024,  
     2026, 2136, 2137  
initials option 76, [109](#)  
`\InnerCite` .. 1419,  
     1424, 1449,  
     [1462](#), 1554, 3001  
inputenc package ... 72  
`\interlinepenalty` 551  
inverted attribute .. 76  
isbn field ..... 8  
`\issuetext` . 7, 325,  
     345, [2672](#), 3087  
`\itemindent` ..... 623
- J**
- `\j` .. 71, 73, 1821, 1876  
journal field ..... 38
- K**
- `\k` ..... 1797  
`\keep@last@digit` .  
     ..... 2774, [2778](#)  
`\kern` ..... 787,  
     2480, 2543, 2584  
klm.bib ..... 9
- L**
- `\L` ..... 1827  
`\l` ..... 1827  
label field .... 87, [112](#)  
`\labelsep` 539, 599, 625  
`\labelwidth` .....  
     597, 598, 619–621  
`\languagename` 1045, [1061](#)  
`\lasp` ..... 1832  
`\lastkern` 269, 2385,  
     2393, 2405, 2543  
`\lastpenalty` 2386, 2388  
`\lc@do` ..... 1857, [1862](#)  
`\lc@edef` [1853](#), 2311, 2312  
`\lcode` ..... 29  
`\lcsymbol` ..... 71
- `\leftmargin` 598, 599,  
     620, 622, 623, 625  
`\let` ..... 23, 58, 59  
`\lfhook` ..... 1805  
`\linebreak` ..... 5  
`\list` ..... 542  
lite option ..... 18  
`\lncan@a` .. 2945, [2961](#)  
logical-quotes op-  
     tion .... 7, [110](#)  
`\lowercase` .....  
     . 29, 710, 856,  
     893, 1041, 1858  
ltoutenc.dtx ..... 72
- M**
- `\m@ne` ..... 266, 1055  
`\m@th` ..... 1455  
`\macrotext` .... [52](#), 55  
`\makeatletter` .... 109  
`\makelabel` ..... 547  
`\MakeUppercase` .. 1204  
`\mathchar` ..... 30  
`\mathcode` ..... 28, 33  
mathscinet package  
     ..... 17, 70, 71  
`\mathsurround` ..... 56  
`\mdash` ..... 8, [2638](#)  
`\mddefault` ..... 1503  
`\mdseries` ..... 1455  
`\meaning` ..... 20, 52  
`\MessageBreak` ....  
     ... 106, 1390,  
     1399, 2925, 2931  
`\MessageBreakNS` [106](#),  
     1175, 1179–1181  
`\modify@xref@fields`  
     ..... 686, 704  
`\ModifyBibLabel` ..  
     ..... 2294, [2321](#)  
`\month@` ..... 106  
`\moo` ..... 21  
`\MR` ..... [2648](#)  
`\MR@url` 2904, 2911, 2914  
`\MRhref` 2652, [2656](#), 2908  
msc-links option .. 111  
`\MultipleBibSelectWarning` ■ ..... 1281, [1283](#)  
     ..... 650, [652](#), 665, 1279  
`\MultipleCiteKeyWarning`  
     ..... 1379, [1387](#)
- N**
- name entry type .... 4  
`\name` ..... 87, [112](#),  
     186, 188–190,  
     843, 859, 863,  
     869, 886, 2181,  
     2184, 2228,  
     2238, 2239,  
     2243, 2246,  
     2281, 2430,  
     2945, 3049, 3060  
`\name@split` .....  
     ... 76, [1907](#), 2444  
`\name@split@finish`  
     ..... 1914, [1916](#)  
`\name@split@given`  
     ... 76, 1908, [1910](#)  
`\name@split@jr` ...  
     ..... 1911, [1913](#)  
`\NameCheck` ..... 1919  
`\ndash` ..... 8, [104](#),  
     [2638](#), 2674, 2679  
`\NeedsTeXFormat` 2, 3199  
`\newcount` .. 2269, 2324  
`\newcounter` .. 496, 585  
`\newdimen` ..... 577  
`\newif` ..... 2306  
`\newlabel` ..... 589  
`\newread` ..... 1290  
`\NG` ..... 1828  
`\ng` ..... 1828  
`\NoBibDBFile` 1302, [1309](#)  
`\nobreak` ..... 80  
`\nobreakspace` 1971, 2733  
nobysame option ... [128](#)  
`\nocite` .... 60, [1540](#)  
non-compressed-cites  
     option .... 110  
non-sorted-cites op-  
     tion .... 17, [110](#)  
`\NonNumericCiteWarning`  
     ..... 1576, [1580](#)  
`\nopunct` ... 7, [122](#),  
     1439, 1659, 2567  
`\normal@bibdef` ...  
     ..... 31, 32,  
     650, [652](#), 665, 1279  
`\NormalCatcodes` ... 37  
`\normalfont` ..... 511

- \normalize@edef ..  
     ..... 1839,  
     1975, 2183, 2947
- \number .....
- 30, 2404, 2733,  
     2758, 2764, 2774
- O**
- \O .....
- \o .....
- \ObsoleteCiteOptionWarning .....
- \ocite .....
- \ocitelist .....
- \ocites ..
- \OE .....
- \oe .....
- \on@line .....
- \OpenBBLFile .....
- \openin .....
- \or .....
- organization field ..
- \OtherCite ..
- \othercitelist ..
- \othercites ..
- \output@inner@xref@ .....
- \output@xref@ .....
- \output@xref@a ..
- P**
- \PackageError .....
- \PackageInfo .....
- \PackageWarning ..
- \PackageWarningNoLine .....
- \parenthesize ..
- \parse@MR ..
- partial field .....
- pcatcode package ...
- place field .....
- \Plural .....
- \polhk .....
- \pop@bracket .....
- \PopCatcodes .....
- \pp@scan ...
- \pp@scan@a ..
- \prev@cite ....
- \prev@cite@cb ....
- \prev@citekey ....
- \prev@names .....
- \previous@primary .....
- \previous@stem .....
- \previous@year ..
- \prime .....
- \print@backrefs ..
- \print@citenames ..
- \print@contribution ..
- \print@date ..
- \print@edition ...
- \print@full@date ..
- \print@month@day ..
- \print@one@dash ..
- \print@partial ...
- \print@primary ...
- \PrintAuthors ..
- \PrintBackRefs ...
- \PrintBook ...
- \PrintCiteNames ..
- \PrintCNY ..
- \PrintConference ..
- \PrintConferenceDetails ..
- \PrintConferenceDetails@ ..
- \PrintContributions .....
- \PrintDate .....
- \PrintDateB ...

- \PrintDateField 468, 2715, 3192, 3331
  - \PrintDatePosted . . . . . 2706, 2744
  - \PrintDatePV . . . . . 324, 344, 2701
  - \PrintDOI 328, 390, 2698, 3090, 3136
  - \PrintEdition . . . . . 7, 8, 359, 409, 427, 2745, 3106, 3152, 3245, 3288
  - \PrintEditorsA . . . . . 7, 8, 97, 98, 971, 2486, 3335
  - \PrintEditorsB . . . . . 7, 360, 410, 2495, 3107
  - \PrintEditorsC 7, 2483
  - \PrintSeries@a . 2334
  - \PrintISBNs . . . . . 2668
  - \PrintNameList 351, 2482, 3239, 3240, 3282, 3283
  - \PrintNames . . . . . 7, 2426, 2474, 2482, 2484, 2491, 2496, 2591, 2598, 2603, 3337, 3342
  - \PrintNames@a 2427, 2429
  - \PrintPartials 321, 2660, 3084, 3208
  - \PrintPrimary . . 7, 355, 423, 2459, 2581, 3101, 3147, 3233, 3295
  - \printref . . . . . 1756
  - \PrintReprint . 332, 372, 394, 438, 456, 2558, 3094, 3118, 3140, 3162, 3180, 3216, 3250, 3266, 3306, 3320
  - \PrintReviews . . 8, 335, 375, 397, 441, 459, 2657, 3097, 3121, 3143, 3165, 3183
  - \PrintSeries 92, 93, 97, 2325, 2430, 2499, 2658, 2661, 2669, 3001, 3049, 3060
  - \PrintSeries@a . . . . . 93, 2332, 2334
  - \PrintThesisType . . . . . 448, 2683, 3173, 3315
  - \PrintTranslation . . . . . 331, 371, 393, 437, 455, 2572, 3093, 3117, 3139, 3161, 3179, 3215, 3249, 3265, 3305, 3319
  - \PrintTranslatorsA . . . . . 7, 98, 968, 2593, 3340
  - \PrintTranslatorsB . . . . . 7, 2602
  - \PrintTranslatorsC . . . . . 7, 361, 411, 2590, 3108
  - \PrintYear 8, 106, 2743
  - proceedings entry type . . . . . 7
  - \process@bare@dot . . . . . 2058, 2068
  - \process@citelist . . . . . 1450, 1588, 2797, 2802, 2860, 2973
  - \process@citelist@sorted . . . . . 1572, 1588
  - \process@citelist@unsorted . . . . . 1577, 1585, 2797, 2802, 2860, 2973
  - \process@dots . . . . . 80, 84, 1979, 2041
  - \process@hyphens 80, 82, 84, 1978, 1988
  - \process@names . . . . . 1980, 2092
  - \process@one@dot . . . . . 83, 2044, 2052, 2074, 2089
  - \process@one@dot@a . . . . . 2054, 2056
  - \process@one@dot@b . . . . . 2061, 2073
  - \process@one@dot@c . . . . . 2063, 2076
  - \process@one@dot@d . . . . . 2071, 2083, 2085
  - \process@one@hyphen . . . . . 81, 1991, 1998, 2020, 2035
  - \process@one@hyphen@a . . . . . 83, 84, 2000, 2002
  - \process@one@hyphen@b . . . . . 2008, 2011, 2019
  - \process@one@hyphen@c . . . . . 2013, 2022
  - \process@one@hyphen@d . . . . . 81, 2029, 2031
  - \process@xrefs 676, 702
  - \ProcessExclusiveOptions . . . . . 152
  - \ProcessOptions . . 150
  - \prop . . . . . 285, 287, 2452
  - \prop@reset . . . . . 298
  - \protected@edef . . . . . 1108, 1162, 1211, 1220
  - \ProvidesFile . . . 1313
  - \ProvidesPackage 3, 3200
  - publisher field . . . . . 38
  - \purge@accent 1873, 1904
  - \purge@edef . . . . . 1869, 2309, 2310
  - \purge@edef@ . . . . . 1879, 1884, 1894, 1895, 1899
  - \purge@edef@a . . . . . 74, 1886, 1888
  - \push@bracket 2613, 2616
  - \PushCatcodes . . . . . 37
- R**
- \r . . . . . 1798
  - \rasp . . . . . 1833
  - \read . . . . . 1324
  - \ReadBibData 1274, 1291
  - \ReadBibData@a . . . . . 1303, 1307, 1312

- \ReadBibLoop . . . . . 1631, 1665,  
     . 1316, 1320, 1334  
 \ReadBibLoop@a . . . . .  
     . . . . . 1325, 1329  
 \ReadBibLoop@b . . . . .  
     . . . . . 1336, 1340  
 \ReadBibLoop@e . . . . . 1329  
 \refname *48*, 1198, 1218  
 \RequirePackage 5,  
     12, 36, 124, 155,  
     165, 178, 181,  
     182, 2899, 3201  
 \reset@nth@property 881  
 \resetbiblist . . . . .  
     . . . . . 595, 618, 1238  
 \ResetCapSFCodes . . . . .  
     . . . . . 555, 602  
 \resolve@inner 2517,  
     2530, 2556,  
     2559, 2573, 2610  
 \restore@labelwidth . . . . .  
     . . . . . 543, 593  
 \RestrictedSetKeys . . . . .  
     . . . . . 12,  
     22, 297, 559,  
     658, 680, 704,  
     2510, 2520, 2922  
 review field . . . . . *6*, *8*  
 rkeyval package . . . . .  
     . . . . . *6*, *11*, *22*, *81*  
 \rkv@DSAK . . . . . *22*  
 \rkv@setter . . . . . 929, 949  
 \rkvIfAdditive . . . . . 275  
 \rkvIfEmpty . . . . .  
     . . . . . 18, 185, 560,  
     563, 2923, 2929  
 \romannumeral . . . . . 106  
 \rsk@resume . . . . . 746  
 \rsk@set . . . . . *33*, 680  
 \rsk@set@a . . . . . *22*  
 \rsk@set@b . . . . . *22*  
 \rsk@toks . . . . . *11*,  
     *12*, *32*, *36*, 298,  
     559, 659, 704,  
     2511, 2520, 2922  
  
     **S**  
 \safe@set . . . . . 81,  
     1608, 1615,
- 1672, 1688, 2258  
 \same@stemsfalse 2316  
 \same@stemstrue . 2314  
 \sameauthors . . . . .  
     . . . . . 7, 8, 2471,  
     2477, 2488, 2595  
 \save@labelwidth . . . . .  
     . . . . . 572, 586  
 \save@primary 824, 957  
 \section\* . . . . . *48*  
 \selbibdef@a . 749, 751  
 \selbibdef@b . 753, 756  
 \select@auxlanguage . . . . .  
     . . . . . 277, 284, 2450  
 \selective@bibdef . . . . .  
     . . . . . *31*,  
     *60*, 748, 1263, 1561  
 \selectlanguage . . . . .  
     . . . . . *42*, *43*, 287, 1047  
 \SentenceSpace 335,  
     375, 397, 441,  
     459, 1753, 2786,  
     3097, 3121,  
     3143, 3165, 3183  
 \series@add . 2343, 2347  
 \series@add@. . . . . *93*  
 \series@add@a 2327, 2359  
 \series@add@b 2328, 2362  
 \series@add@c . . . . .  
     . . . . . 2329, 2364, 2371  
 \series@add@d 2330, 2368  
 \series@add@e 2331, 2353  
 \series@index . . . . .  
     . . . . . *22*, 266,  
     295, 842, 854,  
     881, 2324, 2342,  
     2348, 2350,  
     2355, 2357,  
     2367, 2370, 2509  
 \series@total . . . . .  
     . . . . . 2336, 2337,  
     2361, 2367, 2370  
 \set . . . . . 687–697  
 \set@firstname . . . . .  
     . . . . . 2427, 2437, 3028  
 \set@name . . . . . 2438,  
     2441, 2443, 3029  
 \set@name@a . . . . .  
     . . . . . 98, 2445, 2447  
 \set@namea . . . . . 2447  
 \set@othername 2427,  
     2432–2434, 2440  
 \setbib@ *31*, *32*, *34–36*  
 \setbib@@ 779, 788,  
     793, 796, 799,  
     802, 1001, 2587  
 \setbib@article . . . . .  
     . . . . . 788, 1001  
 \setbib@book . . . . . 2569  
 \setbib@conference 2530  
 \setbib@contribution . . . . .  
     . . . . . 2511  
 \setbib@incollection . . . . .  
     . . . . . 482, 793, 796, 799  
 \setbib@innerbook 2556  
 \setbib@inproceedings . . . . .  
     . . . . . 485  
 \setbib@misc . . . . . 656  
 \setbib@nameBE . . . . .  
     . . . . . 2416, 2453  
 \setbib@nameinverted . . . . .  
     . . . . . 2421, 3029  
 \setbib@nameLE . . . . .  
     . . . . . 2411, 2438, 2441  
 \setbib@partial . 2610  
 \setbib@TYPE . . . . . *20*  
 \setcounter . . . . . 500  
 \settowidth . . . . . 579, 619  
 \sfcode *95*, 114–119,  
     605, 2395, 2553  
 \shade@cite . . . . .  
     . . . . . 1756, 1757, 1764  
 shaderef package . . . . . *68*  
 short-journals op-  
     tion . . . . . *17*, *110*  
 short-months option *110*  
 short-publishers op-  
     tion . . . . . *17*, *110*  
 shortalphabetic op-  
     tion . . . . . *108*  
 showkeys package *54*, *68*  
 \SingularPlural . . . . .  
     . . . . . *8*, *93*, *96*,  
     2338, 2340, 2409  
 \sixt@n . . . . . 1354, 1360  
 \SK@ . . . . . 1100, 1761, 1779  
 \SK@label . . . . . 1100, 1778  
 \SK@ref . . . . . 1761  
 \soft . . . . . 1808

- sorted option . . . . . 16  
\space . . . . . 81  
\spacefactor . . 110,  
122, 2395, 2397,  
2553, 2649, 2786  
\SS . . . . . 1829  
\ss . . . . . 1829  
\star@ 556, 738, 1404,  
1462, 2975,  
2988, 2992,  
3011, 3019, 3025  
\star@bibdef . . . 31,  
632, 663, 720, 1315  
\stepcounter . . . . . 537  
\string . . . . . 72  
\strip@MRprefix . .  
. . . . . 2640, 2651  
\strip@prefix . . 20, 52  
\SubEtal 2351, 2353, 2607  
\surround@names . .  
. . . . . 2213, 2215  
\SwapBreak . . . . 94,  
95, 271, 2328–  
2331, 2380,  
2496, 2550, 2603  
\symbol . . . . . 71
- T**
- \t . . . . . 1799  
tabbing environment 72  
\textbf 5, 323, 343, 3086  
textcmds package . .  
. . . . . 8, 17, 103  
\textemdash . . . . 8, 2638  
\textendash 8, 1659, 2639  
\textfont . . . . . 23  
\textit 317, 319, 340,  
356, 358, 380,  
382, 424, 426,  
446, 447, 3080,  
3082, 3103,  
3105, 3127,  
3129, 3149,  
3151, 3171, 3172  
\textquotedblleft 2632  
\textquotedblright 2635  
\textsc . . . . . 92  
\textsubscript . . . . 7
- \textsuperscript . . . 7  
\texttt . . . . . 20  
\tf@brf . . . . . 2813, 2814  
\TH . . . . . 1830  
\th . . . . . 1830  
\the . . . . . 32, 36  
\thebib . . . . . 517,  
616, 619, 1070,  
1072, 1098, 2172  
thebibliography envi-  
ronment . . . 1236  
\thebibliography . . 109  
\thesis@type 2684, 2687  
\thinspace . . . . . 2479  
\thr@@ . . . . . 2361, 2767  
\toks@ . . . . . 20, 77  
\tprime . . . . . 7  
\TrailingHyphenWarning  
. . . . . 2007, 2038  
transition field 6, 20, 37  
\tsub . . . . . 7  
\tsup . . . . . 7  
\ttfamily . . . . . 1503  
\tw@ . . . . . 2292, 2350  
\typeout 172, 1269, 1271
- U**
- \u . . . . . 1800  
\uarc . . . . . 1803  
\udot . . . . . 1806  
\undefined . . . . . 1222  
\UndefinedCiteWarning  
. . . . . 1493, 1498  
\unkern . . . . . 2385  
\unpenalty . . . . . 2389  
\unskip . . . . . 80, 1987,  
2384, 2649, 2744  
\uparrow . . . . . 1753  
\upn 102, 2613, 2615, 2628  
upref package . . . . 102  
\upshape . . . . . 1455  
\url . . . . . 18, 467,  
2787, 3191, 3330  
\use@accent . . . . .  
. . . . . 1843, 1849, 1855  
\utilde . . . . . 1802
- V**
- \v . . . . . 1801  
\value . . . . . 532  
\vdef 53, 640, 1409, 1474  
\voltext . . . . . 7,  
364, 414, 2671,  
3111, 3238, 3281  
\vphantom . . . . . 2911
- W**
- \widowpenalty . . . . 553  
\wrap@accent . . . . .  
. . . . . 74, 1842, 1850  
wrapper macros, defi-  
nition . . . . . 13
- X**
- xref field . . . . . 39  
\xref@add@toks 911, 926  
\xref@append . . . . 912, 947  
\xref@check@a . . . .  
. . . . . 830–832, 836  
\xref@check@aa 843, 853  
\xref@check@ab 865, 872  
\xref@check@b . . . .  
. . . . . 833, 834, 889  
\xref@check@c . . . .  
. . . . . 39, 829, 900, 905  
\xref@ID . . . . . 915–917, 919  
\XRefWarning . . . . .  
. . . . . 917, 954, 2523  
\XrefWarning . . . . . 954  
xyz.aux . . . . . 9  
xyz.bbl . . . . . 9  
xyz.tex . . . . . 9
- Y**
- y2k option . . . . . 86, 111, 127  
\ycite . . . . . 59, 1532,  
1534, 1538,  
1539, 2974, 3014  
\ycites . . . . . 1533, 2977  
\year@short . . . . .  
. . . . . 2256, 2263, 2889
- Z**
- \Z . . . . . 607  
\zap@space . . . . .  
. . . . . 642, 1411, 1476