

The `amsrefs` package

Michael Downes and David M. Jones
American Mathematical Society

Version 2.14, 2013/03/07

Contents

1	Introduction	2
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	4
5	Miscellaneous commands provided by the <code>amsrefs</code> package	5
6	Implementation	8
6.1	Overview	8
6.1.1	Normal L ^A T _E X processing of cites	8
6.2	How cites are processed by <code>amsrefs</code>	9
6.3	Data structures	10
6.4	Preliminaries	11
6.5	Utilities	12
6.6	Declaring package options	14
6.6.1	The <code>?</code> option	16
6.7	Loading auxiliary packages	16
6.7.1	The <code>lite</code> option	16
6.8	Key-value setup	16
6.8.1	Standard field names (the <code>bib</code> group)	17
6.8.2	Auxiliary properties (the <code>prop</code> group)	18
6.9	Bibliography type specifications	18
6.10	The standard bibliography types	20
6.11	The <code>\biblist</code> environment	24
6.12	Processing bibliography entries	27
6.12.1	<code>\@bibdef</code> Implementations	28
6.12.2	<code>\bib@exec</code> Implementations	31
6.12.3	Resolving cross-references	32
6.12.4	Bib field preprocessing	35
6.12.5	Date setup	36
6.12.6	Language setup	37
6.12.7	Citation label setup	38
6.12.8	Printing the bibliography	41
6.13	Name, journal and publisher abbreviations	42
6.14	Processing <code>.ltb</code> files	42
6.15	Citation processing	45
6.15.1	The <code>\citesel</code> structure	45
6.15.2	The basic <code>\cite</code> command	46
6.15.3	Fancier <code>\cite</code> commands	50
6.15.4	The <code>\nocite</code> command	51
6.15.5	Citation sorting	52

6.15.6	Range compression	53
6.15.7	Munging the <code>.aux</code> file	56
6.15.8	Back references	57
6.15.9	<code>hyperref</code> , <code>showkeys</code> and <code>shaderef</code> support	58
6.16	Lexical structure of names	58
6.16.1	Text accents	59
6.16.2	Text symbols	60
6.16.3	<code>\edef</code> -like macros for names	61
6.17	Name parsing	63
6.18	Extracting initials	67
6.18.1	The algorithm	67
6.18.2	The implementation	68
6.19	Generating alphabetic labels	73
6.19.1	The algorithm	73
6.19.2	The implementation	74
6.20	Generating short alphabetic labels	78
6.21	Formatting series	78
6.22	Formatting names and series of names	81
6.23	The <code>partial</code> field	87
6.24	Special formatting for other fields	87
6.25	<code>BibTeX</code> support	92
6.26	Implementing package options	92
6.26.1	The <code>alphabetic</code> option	92
6.26.2	The <code>shortalphabetic</code> option	92
6.26.3	The <code>backrefs</code> option	92
6.26.4	The <code>citation-order</code> option	92
6.26.5	The <code>initials</code> option	93
6.26.6	The <code>logical-quotes</code> option	93
6.26.7	The <code>non-compressed-cites</code> option	93
6.26.8	The <code>non-sorted-cites</code> option	93
6.26.9	The <code>short-journals</code> option	93
6.26.10	The <code>short-publishers</code> option	94
6.26.11	The <code>short-months</code> option	94
6.26.12	The <code>y2k</code> option	94
6.26.13	The <code>bibtex-style</code> option	94
6.26.14	The <code>msc-links</code> option	94
6.26.15	The <code>author-year</code> option	95
6.27	The <code>amsbst</code> package	101

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 L^AT_EX 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 L^AT_EX knowledge All bibliography setup can be done in L^AT_EX; learning another programming language (such as the one used in BibT_EX `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 BibT_EX.

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 BibT_EX to me, so I have re-instated `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 \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, \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 \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 `LATEX`, these do not use math mode at all.¹

`\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.

¹There 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.

`\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`.

`\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 `\textemdash` 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 $\text{Bib}\text{\TeX}$.

1. A `\cite{moo}` command writes one line to the `.aux` file: `\citation{moo}`. This indicates to $\text{Bib}\text{\TeX}$ that it should include 'moo' in the list of cited items to be searched for. The `\cite` command also checks to see if `\b@{moo}` 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 $\text{Bib}\text{\TeX}$ 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 $\text{Bib}\text{\TeX}$ 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.

$\text{Bib}\text{\TeX}$ pass For a document named `xyz.tex`, the command `bibtex xyz` is used to invoke $\text{Bib}\text{\TeX}$. It looks in `xyz.aux` to find the citation information written there by \LaTeX . For each `\citation` line, $\text{Bib}\text{\TeX}$ 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: `\bibcite{moo}{9}`, where 9 is the current item number.
4. A `\bibitem[Moody]{moo}` command writes one line to the `.aux` file: `\bibcite{moo}{Moody}`, using the supplied label instead of a number.

Third \LaTeX pass Now the `.aux` file contains some `\bibcite` commands. Once again, \LaTeX reads the `.aux` file when it reaches `\begin{document}`.

1. A `\bibcite{moo}{Moody}` causes \LaTeX to define `\b@moo` with ‘Moody’ as the replacement text.
2. If two `\bibcite` commands have the same citation key, \LaTeX 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 `\bibcite` command, \LaTeX will give an ‘Undefined citation’ warning. This often happens if the `.aux` file is incomplete due to a \TeX 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 `\bibcite` 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, `\bibcite` 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 `\bibcite` in the `.aux` file for a given key, then the info is

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

If there was neither `\citation` nor `\bibcite` 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}[2013/03/07 v2.14]

\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{pcatcode}\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.

<code>\after@deleting@token</code>	Similar in concept to <code>\afterassignment</code> , except it deletes the next token in the stream before putting its argument back into the input. Useful for skipping past tokens during parsing.
	<pre> 41 \def\after@deleting@token#1{% 42 \afterassignment#1% 43 \let\@let@token= % Don't delete this space! 44 }</pre>
<code>\@ifempty</code>	Some frequently used tests for empty arguments. Note that an argument consisting entirely of spaces (e.g., <code>\@ifempty{\ }</code>) counts as empty.
<code>\@ifnotempty</code>	<pre> 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}{}}</pre>
<code>\macrotext</code>	<pre> 52 \def\macrotext{\expandafter\strip@prefix\meaning}</pre>
<code>\vdef</code>	“Verbatim” def.
	<pre> 53 \def\vdef#1#2{% 54 \def#1{#2}% 55 \edef#1{\macrotext#1}% 56 }</pre>
<code>\auto@protect</code>	Sometimes it’s convenient to render a given control sequence unexpandable for a time. <code>\auto@protect</code> provides a way to do that. ²
	An earlier version of this code read simply <code>\let#1\relax</code> but that had the disadvantage of making all <code>\auto@protected</code> macros compare equal via <code>\ifx</code> . This version allows macros to keep their identities under comparisons.
	<pre> 57 \def\auto@protect#1{\def#1{\@nx#1}}</pre>
<code>\auto@stringify</code>	<pre> 58 \def\auto@stringify#1{\def#1{\string#1}}</pre>
<code>\g@undef</code>	Globally undefine a control sequence.
	<pre> 59 \def\g@undef#1{\global\let#1\relax}</pre>
<code>\@concat</code>	Concatenate onto the end of a token list. Expands everything.
	<pre> 60 \def\@concat#1#2{\edef#1{#1#2}}</pre>
<code>\add@toks@</code>	This saves a few tokens of main memory and a lot of typing.
	<pre> 61 \def\add@toks@{\addto@hook\toks@}</pre>
<code>\@lappend</code>	Append an element to a <code>\do</code> -delimited list. As long as the element to be appended (<code>#2</code>) is a single token, nothing is expanded. If it contains multiple tokens, all tokens after the first will be expanded.
	<pre> 62 \def\@lappend#1#2{%</pre>

²There 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.

```

63   \begingroup
64     \def\do{\@nx\do\@nx}%
65     \edef\@tempa{\def\@nx#1{#1\do#2}}%
66   \@xp\endgroup
67   \@tempa
68 }

```

\@apply Apply a macro to each element of a \do-delimited list.

```

69 \def\@apply#1#2{%
70   \let\do#1%
71   #2%
72 }

```

\get@numberof This is a generic macro for counting the number of elements in a L^AT_EX-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.

```

73 \def\get@numberof#1#2#3{%
74   \begingroup
75     \def#2{\advance\@tempcnta\@ne \gobble}%
76     \@tempcnta\z@
77     #3\relax
78     \edef\@tempb{#1=the\@tempcnta\relax}%
79   \@xp\endgroup
80   \@tempb
81 }

```

\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.)

```

82 \def\safe@set#1#2{%
83   \afterassignment\@nilgobble
84   #1=0#2\relax\@nil
85 }

```

\@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.

```

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

```

```

98    \@xp\endgroup
99    \@tempa
100 }

```

`\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.

```

101 \def\chomp@{%
102     \futurelet\@let@token
103     \@chomper
104 }

```

`\amsrefs@warning`

```

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

```

`\amsrefs@error`

```

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

```

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

```

107 \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.

```

108 \@ifundefined{@addpunct}{%
109     \def\@addpunct#1{%
110         \relax\ifhmode
111             \ifnum\spacefactor>\@m \else#1\fi
112         \fi
113     }
114     \def\frenchspacing{%
115         \sfcode`.1006
116         \sfcode`?.1005
117         \sfcode`!.1004
118         \sfcode`\:1003
119         \sfcode`;1002
120         \sfcode`\,1001\relax
121     }
122 }{}

```

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

```

123 \providecommand{\nopunct}{\spacefactor \@nopunctsfcode}

```

`\@nopunctsfcode`

```

124 \def\@nopunctsfcode{1007 }

```

6.6 Declaring package options

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

```

125 \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.

```

126 \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 `LlYY` for this option. Numeric is the default since it is commoner in AMS publications.

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

`y2k`

```
128 \DeclareBooleanOption{y2k}
```

`nobysame`

```
129 \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.

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

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

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

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

```
133 \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.

```
134 \DeclareOption{abbrev}{%
135   \@pass@options
136   \@currentt
137   {initials,short-months,short-journals,short-publishers}%
138   \@currname
139 }
```

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.

```
140 \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.

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

```
142 \DeclareExclusiveOptions{non-compressed-cites,compressed-cites}
```

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

```
143 \DeclareBooleanOption{backrefs}
```

Option for giving information about the available options:

```
144 \DeclareBooleanOption{?}
```

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

```
145 \DeclareBooleanOption{lite}
```

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

```
146 \DeclareBooleanOption{beta}
```

```

147 \DeclareBooleanOption{bibtex-style}
148 \DeclareBooleanOption{msc-links}
149 \ExecuteOptions{numeric,traditional-quotes,sorted-cites,compressed-cites}
150
151 \ProcessOptions\relax
152
153 \ProcessExclusiveOptions
154 \IfOption{backrefs}{%
155   \IfFileExists{backref.sty}{%
156     \RequirePackage{backref}[1999/05/30]
157   }{%
158     \amsrefs@warning@nl{The backrefs option cannot be used^^J%
159       unless the backref package is also installed.^^J%
160       (backref is part of the hyperref package)}%
161   }%
162 }{}
163
164 \IfOption{msc-links}{%
165   \IfFileExists{hyperref.sty}{%
166     \RequirePackage{hyperref}[1999/07/08]
167   }{
168     \amsrefs@warning@nl{The msc-links option cannot be used^^J%
169       unless the hyperref package is installed}%
170   }%
171 }{}

```

6.6.1 The ? option

```

172 \IfOption{?}{%
173   \typeout{^^J%
174     Documentation for the amsrefs package is found in amsrdoc.dvi^^J%
175     (or .pdf or .tex).
176     ^^J%
177   }%
178 }{}%

```

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.

```

179 \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.

```

180 \IfOption{lite}{% True? Then don't load the next two packages.
181 }{% False? OK, let's load them:
182   \RequirePackage{textcmds}[2001/12/14]
183   \RequirePackage{mathscinet}[2002/01/01]
184 }

```

6.8 Key-value setup

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

```

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

```

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

```

186 \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
\name processing is done. (It's not clear to me there's any real benefit to using these
instead of just using \do.—dmj)
187 \let\fld@elt=?
188 \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.

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

197 \DefineSimpleKey{bib}{accessdate}
198 \DefineSimpleKey{bib}{address}
199 \DefineSimpleKey{bib}{book}
200 \DefineSimpleKey{bib}{booktitle}
201 \DefineSimpleKey{bib}{conference}
202 %\DefineSimpleKey{bib}{contributor}
203 \DefineSimpleKey{bib}{copula}
204 \DefineSimpleKey{bib}{date}
205 \DefineSimpleKey{bib}{doi}
206 \DefineSimpleKey{bib}{edition}
207 \DefineSimpleKey{bib}{eprint}
208 \DefineSimpleKey{bib}{fulljournal}
209 \DefineSimpleKey{bib}{hyphenation}
210 \DefineSimpleKey{bib}{institution}
211 \DefineSimpleKey{bib}{journal}
212 \DefineSimpleKey{bib}{label}
213 \DefineSimpleKey{bib}{language}
214 \DefineSimpleKey{bib}{name}
215 \DefineSimpleKey{bib}{note}
216 \DefineSimpleKey{bib}{number}
217 \DefineSimpleKey{bib}{organization}
218 \DefineSimpleKey{bib}{pages}
219 \DefineSimpleKey{bib}{part}
220 \DefineSimpleKey{bib}{place}
221 \DefineSimpleKey{bib}{publisher}
222 \DefineSimpleKey{bib}{reprint}
223 \DefineSimpleKey{bib}{school}
224 \DefineSimpleKey{bib}{series}
225 \DefineSimpleKey{bib}{setup}
226 \DefineSimpleKey{bib}{status}
227 \DefineSimpleKey{bib}{subtitle}
228 \DefineSimpleKey{bib}{title}
229 \DefineSimpleKey{bib}{translation}
230 \DefineSimpleKey{bib}{type}
231 \DefineSimpleKey{bib}{url}
232 \DefineSimpleKey{bib}{volume}
```

```

233 \DefineSimpleKey{bib}{xref}
234 \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.

```
235 \DefineDummyKey{bib}{transition}
```

6.8.2 Auxiliary properties (the `prop` group)

```

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

```

6.9 Bibliography type specifications

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

```

238 \newcommand{\BibSpec}[2]{%
239   \toks@{\@emptytoks}
240   \@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 `\@empty`s appended to the specification (`#2`) below ensure that this is so.

```

241     \@ifnextchar{+}{\@xp\bibspeg@scan\@gobble}{\bibspeg@scan}%
242     #2\@empty\@empty\@empty\@empty
243   }%
244   \@xp\edef\csname setbib@#1\endcsname{\the\toks@}%
245 }

```

`\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.

```

246 \def\bibspeg@scan#1#2#3#4{%
247   \add@toks@{\bib@append{#1}{#2}}%
248   \edef\@tempa{%
249     \toks@{\the\toks@ \@xp\@nx\csname bib'#3\endcsname}%
250   }%
251   \@tempa
252   \ifx\@empty#4%
253     \@xp\@gobble % end the recursion
254   \else
255     \ifx +#4\else\bibspeg@scan@error\fi
256   \fi
257   \bibspeg@scan
258 }

```

`\bibspeg@scan@error`

```
259 \def\bibspeg@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`.

```
260 \def\bib@append#1#2#3{%
261     \ifx\@empty#3%
262     \else
        Known bug: Need better error message here.
263     \ifx\relax#3%
264         \errmessage{#3=\relax}%
265     \else
266         \begingroup
267             \series@index\m@ne
268             \def\current@bibfield{#3}%
269             \@ifempty{#1}{%
270                 \@temptokena{\ifnum\lastkern=\@ne\ignorespaces\fi #2}%
271             }{%
272                 \@temptokena{\SwapBreak{#1}#2}%
273             }%
274             \toks@\xp{#3}%
275             \edef\@tempa{\the\@temptokena\the\toks@}%
276             \rkvIfAdditive#3{}{%
277                 \get@current@properties
278                 \select@auxlanguage
279             }%
280             \@tempa
281         \endgroup
282     \fi
283 \fi
284 }

\select@auxlanguage

285 \def\select@auxlanguage{%
286     \ifx\prop'language\@empty
287     \else
288         \@xp\selectlanguage\@xp{\prop'language}%
289     \fi
290 }
```

\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 `\rkval@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).

```
291 \def\erase@field#1{%
292   \aftergroup\let\aftergroup#1\aftergroup\@empty
293 }
```

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

```
294 \def\get@current@properties{%
295   \begingroup
296     \exp\get@nth@property\exp\@tempa\current@bibfield\series@index
297     \edef\@tempa{%
298       \nx\RestrictedSetKeys{}\prop}\%
299       \def\@nx\@tempa{\nx\prop@reset \nx\the\@nx\rsk@toks}%
300     }\@tempa}%
301   }%
302   \@tempa
303   \exp\endgroup
304   \@tempa
305 }
```

`\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.

```
306 \newcommand{\BibSpecAlias}[2]{%
307   \exp\def\exp\@tempa\exp{\csname setbib@#1\exp\endcsname}%
308   \exp\ifx\csname setbib@#2\endcsname\@tempa
309     \amsrefs@error{%
310       Mirror alias #1->#2 not allowed (infinite loop)}\@ehc
311   \else
312     \exp\def\csname setbib@#1\exp\endcsname
313       \exp{\csname setbib@#2\endcsname}%
314   \fi
315 }
```

6.10 The standard bibliography types

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

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

```
325   +{} { \PrintDatePV}           {date}
```

```

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

```

```

385   +{,} { \PrintConference}           {conference}
386   +{}  {\PrintBook}                  {book}
387   +{,} { }                          {booktitle}
388   +{,} { \PrintDateB}                {date}
389   +{,} { pp.~}                      {pages}
390   +{,} { }                          {status}
391   +{,} { \PrintDOI}                  {doi}
392   +{,} { available at \eprint}       {eprint}
393   +{}  { \parenthesize}              {language}
394   +{}  { \PrintTranslation}          {translation}
395   +{;} { \PrintReprint}              {reprint}
396   +{.} { }                          {note}
397   +{.} {}                          {transition}
398   +{}  {\SentenceSpace \PrintReviews} {review}
399 }
400
401 \BibSpec{conference}{%
402   +{}  {}                          {title}
403   +{}  {\PrintConferenceDetails} {transition}
404 }
405
406 \BibSpec{innerbook}{%
407   +{,} { }                          {title}
408   +{.} { }                          {part}
409   +{:} { }                          {subtitle}
410   +{,} { \PrintEdition}             {edition}
411   +{}  { \PrintEditorsB}            {editor}
412   +{,} { \PrintTranslatorsC}       {translator}
413   +{,} { \PrintContributions}      {contribution}
414   +{,} { }                          {series}
415   +{,} { \voltext}                 {volume}
416   +{,} { }                          {publisher}
417   +{,} { }                          {organization}
418   +{,} { }                          {address}
419   +{,} { \PrintDateB}              {date}
420   +{.} { }                          {note}
421 }
422
423 \BibSpec{report}{%
424   +{}  {\PrintPrimary}              {transition}
425   +{,} { \textit}                  {title}
426   +{.} { }                          {part}
427   +{:} { \textit}                  {subtitle}
428   +{,} { \PrintEdition}            {edition}
429   +{,} { \PrintContributions}      {contribution}
430   +{,} { Technical Report }        {number}
431   +{,} { }                          {series}
432   +{,} { }                          {organization}
433   +{,} { }                          {address}
434   +{,} { \PrintDateB}              {date}
435   +{,} { \eprint}                  {eprint}
436   +{,} { }                          {status}
437   +{}  { \parenthesize}            {language}
438   +{}  { \PrintTranslation}        {translation}
439   +{;} { \PrintReprint}            {reprint}
440   +{.} { }                          {note}
441   +{.} {}                          {transition}
442   +{}  {\SentenceSpace \PrintReviews} {review}
443 }

```

```

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

```

\setbib@incollection

```

483 \edef\setbib@incollection{%
484   \@xp\@nx\csname setbib@collection.article\endcsname
485 }

```

\setbib@inproceedings

```

486 \edef\setbib@inproceedings{%
487   \@xp\@nx\csname setbib@collection.article\endcsname
488 }

```

Some more entry types for implementing abbreviations.

```

489 \BibSpec{name}{%
490   +{} {\PrintAuthors}    {name}
491 }
492
493 \BibSpec{publisher}{%
494   +{,} { } {publisher}
495   +{,} { } {address}
496 }

```

6.11 The biblist environment

The `biblist` environment can be used with a section or chapter heading.

Use a standard \LaTeX counter for numbering bibliography items.

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

`biblist`

```
500 \newenvironment{biblist}{%
501   \setcounter{bib}\z@
502   \@biblist
503 }{%
504   \@endbiblist
505 }
```

`biblist*`

```
506 \newenvironment{biblist*}{%
507   \@biblist
508 }{%
509   \@endbiblist
510 }
```

`\biblistfont`

```
511 \newcommand{\biblistfont}{%
512   \normalfont
513   \footnotesize
514 }
```

`\amsrefs@lbibitem`

`\amsrefs@bibitem`

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`.

```
515 \def\amsrefs@lbibitem[#1]#2{%
516   \begingroup
517     \def\CurrentBib{#2}%
518     \def\thebib{#1}%
519     \@nmbrlistfalse
520     \item\leavevmode
521     \if@filesw
522       {\let\protect\noexpand
523        \immediate\write\@auxout{\string\bibcite{#2}{\thebib}}}
524     \fi
525   \endgroup
526   \ignorespaces
527 }
528
529 \def\amsrefs@bibitem#1{%
530   \def\CurrentBib{#1}%
531   \item
532   \if@filesw
533     \immediate\write\@auxout{\string\bibcite{#1}{\thevalue{\@listctr}}}%
534   \fi
535   \ignorespaces
536 }
```

`\@biblist`

```
537 \newcommand\@biblist[1] []{%
```



```

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

```

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

```

552 \interlinepenalty\@m
553 \clubpenalty\@M
554 \widowpenalty\clubpenalty
555 \frenchspacing
556 \ResetCapSFCodes
557 \@ifstar{\@biblistsetup}{}%
558 }

```

\@biblistsetup

```

559 \newcommand{\@biblistsetup}[1]{%
560     \RestrictedSetKeys{\biblist}{\the\rsk@toks}{#1}%
561     \rkvIfEmpty{biblist}{prefix}{}%
562     \let\amsrefs@label@prefix\biblist'prefix
563     }%
564     \rkvIfEmpty{biblist}{labels}{}%
565     \@ifundefined{amsrefs@option@biblist'labels}{%
566         \amsrefs@warning{Invalid label style `biblist'labels`}%
567     }{%
568         \csname amsrefs@option@biblist'labels\endcsname
569     }%
570     }%
571 }

```

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

```

572 \def\@endbiblist{%
573     \save@labelwidth
574     \def\@noitemerr{\@latex@warning{Empty bibliography list}}%
575     \global\let\previous@primary\@empty
576     \endlist
577 }

```

\@maxlabelwidth

```

578 \newdimen\@maxlabelwidth

```

\bib@mklab

```

579 \def\bib@mklab#1{%
580     \settowidth\@tempdima{#1}%
581     \ifdim \@tempdima > \@maxlabelwidth
582         \global\@maxlabelwidth\@tempdima
583     \fi
584     #1\hfil
585 }

```

```

586 \newcounter{bib@env}

\save@labelwidth
587 \def\save@labelwidth{%
588   \if@filesw
589     \immediate\write\@auxout{%
590       \string\newlabel{[bibenv:\the\c@bib@env]}{\the\@maxlabelwidth}%
591     }%
592   \fi
593 }

\restore@labelwidth
594 \def\restore@labelwidth{%
595   \@xp\ifx \csname r@[bibenv:\the\c@bib@env]\endcsname \relax
596     \resetbiblist{00}%
597   \else
598     \@xp\labelwidth\csname r@[bibenv:\the\c@bib@env]\endcsname
599     \leftmargin\labelwidth
600     \advance\leftmargin\labelsep
601   \fi
602 }

\ResetCapSFCodes Presumably this is here because there has been a problem in the past with
packages that change the \catcodes of capital letters.
603 \providecommand{\ResetCapSFCodes}{%
604   \count@=-\A
605   \def\@tempa{%
606     \sfcode\count@=\@m
607     \advance\count@\@ne
608     \ifnum\count@>\Z\relax \expandafter\@gobble \fi
609   \@tempa
610 }%
611 \@tempa
612 }

\CurrentBib In case this is undefined sometimes.
613 \def\CurrentBib{??}

\BibLabel
614 \newcommand{\BibLabel}{%
615   \hfill
616   \Hy@raisedlink{\hyper@anchorstart{cite.\CurrentBib}\hyper@anchorend}%
617   [\thebib]%
618 }

\resetbiblist
619 \newcommand{\resetbiblist}[1]{%
620   \settowidth\labelwidth{\def\thebib{#1}\BibLabel}%
621   \leftmargin\labelwidth
622   \ifdim\labelwidth=\z@
623     \leftmargin=1em
624     \itemindent=-\leftmargin
625   \else
626     \advance\leftmargin\labelsep
627   \fi
628 }

```

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.

```

629 \newcommand{\bib}{%
630     \begingroup
631     \ifstar{%
632         \@tempswatrue
633         \let\@bibdef\star@bibdef
634         \BibItem
635     }{%
636         \@tempswafalse
637         \BibItem
638     }%
639 }
```

`\BibItem` *Arguments:*

```

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

```

640 \newcommand{\BibItem}[2]{%
641     \vdef\@tempa{#1}%
642     \edef\@tempa{%
643         \edef\@nx\@tempa{\@nx\@xp\@nx\zap@space\@tempa\space\@nx\@empty}%
644     }%
645     \@tempa
646     \edef\@tempb{%
647         \@nx\@bibdef\@xp\@nx\csname setbib@#2\endcsname{#2}{\@tempa}%
648     }%
649     \@tempb
650 }
```

`\@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.
```

```

651 \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.
```

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

6.12.1 \@bibdef Implementations

`\normal@bibdef` *Arguments:*

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

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

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

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

`\star@bibdef` *Arguments:*

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

```
664 \def\star@bibdef{%
665   \let\bib@exec\bib@store
666   \normal@bibdef
667 }
```

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

```
668 \def\copy@bibdef{%
669   \if@tempwa
670     \xp\defer@bibdef
671   \else
672     \xp\copy@bibdef@a
673   \fi
674 }
```

`\copy@bibdef@a`

```
675 \def\copy@bibdef@a#1#2#3#4{%
676   \@open@bbl@file
677   \process@xrefs{#4}%
678   \bbl@write{%
679     \string\bib\if@tempwa*\fi{#3}{#2}\string{\iffalse}\fi
680   }%
```

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.

```
681   \RestrictedSetKeys{\global\let\rsk@set\bbl@copy}\@empty
```

```

682      {\bbl@write{\iffalse{\fi\string}^^J}%
683      \endgroup}{#4}%
684 }

685 \catcode`\:=11
686
687 \def\modify@xref@fields{%
688   \let\set:bib'author\output@xref@a
689   \let\set:bib'editor\output@xref@a
690   \let\set:bib'translator\output@xref@a
691   \let\set:bib'journal\output@xref@a
692   \let\set:bib'publisher\output@xref@a
693   \def\set:bib'xref##1##2{\output@xref@{##1}\@empty}%
694   \def\set:bib'book##1##2{\output@inner@xref@{##1}\@empty}%
695   \let\set:bib'conference\set:bib'book
696   \let\set:bib'partial\set:bib'book
697   \let\set:bib'reprint\set:bib'book
698   \let\set:bib'translation\set:bib'book
699 }
700
701 \catcode`\:=12
702
703 \def\process@xrefs#1{%
704   \begingroup
705     \RestrictedSetKeys{\modify@xref@fields}{bib}{\the\rsk@toks}{#1}%
706   \endgroup
707 }
708
709 \def\output@xref@a#1#2{%
710   \def\@tempa{#1}%
711   \lowercase{\def\@tempb{#1}}%
712   \ifx\@tempa\@tempb
713     \output@xref@{#1}%
714   \fi
715 }
716
717 \def\output@xref@#1{%
718   \@ifnotempty{#1}{%
719     \@ifundefined{bi@#1}{%
720       \begingroup
721         \let\star@bibdef\copy@bibdef@a
722         \csname bi@#1\endcsname
723       \endgroup
724     }%
725     \@xp\g@undef\csname bi@#1\endcsname
726   }%
727 }
728
729 \def\output@inner@xref@#1{%
730   \in@={#1}%
731   \ifin@
732     \output@xref@{#1}%
733   \fi
734 }

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

```

```

739      \star@{\bbl@copy@a}{}%
740 }

```

\bbl@copy@a

```

741 \def\bbl@copy@a#1{%
742     \@ifnotempty{#1}{%
743         \add@toks@{*{#1}}%
744     }%
745     \bbl@write{ \space\@tempa=\the\toks@,}%
746     \endgroup
747     \rsk@resume
748 }

```

\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.

```

```

749 \def\selective@bibdef#1#2#3{%
750     \@xp\selbibdef@a\csname b@#3\endcsname{#1}{#2}{#3}%
751 }

```

\selbibdef@a

```

752 \def\selbibdef@a#1{%
753     \def\@tempa{\endgroup\@gobblefour}%
754     \ifx\relax#1\else \@xp\selbibdef@b#1\@nil \fi
755     \@tempa
756 }

```

\selbibdef@b

```

757 \def\selbibdef@b#1#2#3\@nil{%
758     \ifx 1#2\let\@tempa\copy@bibdef\fi
759 }

```

\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.

```

```

760 \def\defer@bibdef#1#2#3#4{%
761     \@xp\gdef\csname bi@#3\endcsname{%
762         \bib*{#3}{#2}{#4}%
763     }%
764     \@xp\addto@defer@list \csname bi@#3\endcsname
765     \endgroup
766 }

```

\bibdefer@list

```

767 \let\bibdefer@list\@empty

```

\addto@defer@list

```

768 \def\addto@defer@list#1{%
769     \begingroup

```

```

770      \def\do{\@nx\do\@nx}%
771      \xdef\bibdefer@list{\bibdefer@list\do#1}%
772  \endgroup
773 }

```

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`.

```

774 \def\bib@store#1{%
775   \afterassignment\@gobble
776   \@xp\xdef\csname bi@#1\endcsname
777 }

```

\bib@print *Arguments:*

```

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

778 \def\bib@print#1#2#3{%
779   \bib@start{#1}%
780   \let\setbib@@#3%
781   #2\relax      % execute definitions locally
782   \bib@resolve@xrefs
783   \bib@field@patches
784   \bib@selectlanguage
785   \generate@label
786   \bib'setup
787   \bib@cite{#1}%
788   \kern\@ne sp
789   \ifx\setbib@@\setbib@article
790     \ifx\bib'booktitle\@empty
791       \ifx\bib'book\@empty
792         \ifx\bib'conference\@empty
793           \else
794             \let\setbib@@\setbib@incollection
795           \fi
796         \else
797           \let\setbib@@\setbib@incollection
798         \fi
799       \else
800         \let\setbib@@\setbib@incollection
801       \fi
802     \fi
803     \setbib@@
804   \bib@end
805 }

```

\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.

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

```

```

813     \endgroup
814 }

\current@citekey
815 \let\current@citekey\@empty

\prev@citekey
816 \let\prev@citekey\@empty

\bib@start There used to be more to it.
817 \def\bib@start#1{%
818     \begingroup
819     \def\current@citekey{#1}%
820 }

\bib@end Instead of being handled by \bib@end, ending punctuation is normally handled
via the transition field (q.v.)
821 \def\bib@end{%
822     \relax
823     \@xp\PrintBackRefs\@xp{\CurrentBib}%
824     \par
825     \save@primary
826     \global\let\prev@citekey\current@citekey
827     \endgroup
828 }

```

6.12.3 Resolving cross-references

```

\bib@resolve@xrefs
829 \def\bib@resolve@xrefs{%
830     \xref@check@c\bib'xref
831     \xref@check@a\bib'author
832     \xref@check@a\bib'editor
833     \xref@check@a\bib'translator
834     \xref@check@b\bib'journal
835     \xref@check@b\bib'publisher
836 }

\xref@check@a Resolve a contributor (typically a \DefineName) alias. Requires rebuilding the
list.
837 \def\xref@check@a#1{%
838     \ifx\@empty#1\relax
839     \else
840         \begingroup
841             \toks@\@emptytoks
842             \@temptokenb\@emptytoks
843             \series@index\z@
844             \def\name{\xref@check@aa#1}%
845             #1\relax
846             \edef\@tempa{%
847                 \def\@nx#1{\the\toks@}%
848                 \the\@temptokenb
849             }%
850             \@xp\endgroup
851             \@tempa
852         \fi
853 }

```


\xref@check@aa

```

854 \def\xref@check@aa#1#2{%
855   \advance\series@index\@ne
856   \def\@tempa{#2}%
857   \lowercase{\def\@tempb{#2}}%
858   \ifx\@tempa\@tempb
859     \ifx\@tempa\@empty
860       \add@toks@{\name{}}%
861     \else
862       \ifundefined{bi@#2}{%
863         \BibAbbrevWarning{#2}%
864         \add@toks@{\name{#2}}%
865       }{%
866         \xref@check@ab#1{#2}%
867       }%
868     \fi
869   \else
870     \add@toks@{\name{#2}}%
871   \fi
872 }
```

\xref@check@ab

```

873 \def\xref@check@ab#1#2{%
874   \csname bi@#2\endcsname
875   \ifx\@empty\bib'name
876     \@temptokena{#2}%
877   \else
878     \@temptokena\exp{\bib'name}%
879     \get@property\@tempa\bib'name
880     \edef\@tempa{%
881       \@nx\addto@hook\@temptokenb{%
882         \@nx\reset@nth@property\@nx#1\the\series@index{\@tempa}%
883       }%
884     }%
885     \@tempa
886   \fi
887   \edef\@tempa{\@nx\add@toks@{\@nx\name{\the\@temptokena}}}%
888   \@tempa
889 }
```

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

```

890 \def\xref@check@b#1{%
891   \ifx\@empty#1%
892   \else
893     \toks@{\xp{#1}%
894     \edef\@tempb{\lowercase{\def\@nx\@tempa{\the\toks@}}}%
895     \@tempb
896     \ifx\@tempa#1\relax % all lowercase
897       \ifundefined{bi@#1}{%
898         \BibAbbrevWarning{#1}%
899       }{%
```

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.

```

900       \let#1\@empty
901       \xref@check@c\@tempa
902     }%
```

```

903     \fi
904   \fi
905 }

```

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

```

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

```

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

```

916       \@for\xref@ID:=#1\do{%
917         \ifundefined{bi@\xref@ID}{%
918           \XRefWarning{\xref@ID}%
919         }{%
920           \csname bi@\xref@ID\endcsname
921         }%
922       }%
923       \edef\@tempa{\endgroup\the\toks@}%
924       \@tempa
925     \fi
926 }

```

`\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.

```

```

927 \def\xref@add@toks#1#2#3{%
928   \ifx#1\@empty
929     \edef\@tempa{%
930       \@nx\add@toks@{\@xp\@nx\csname\rkv@setter#1\endcsname{#2}{#3}}%
931     }%
932     \@tempa
933   \else
934     \in@\bib'title{#1}%
935     \ifin@
936       \ifx\bib'booktitle\@empty
937         \edef\@tempa{%
938           \@nx\add@toks@{%
939             \@xp\@nx\csname set:bib'booktitle\endcsname
940           }%
941         }%
942         \@tempa
943         \add@toks@{#2}{#3}%
944       \fi
945     \fi
946   \fi
947 }

948 \def\xref@append#1#2#3#4{%
949   \edef\@tempa{%

```

```

950      \@nx\add@toks@{\@xp\@nx\csname\rkv@setter#2\endcsname{#3}{#4}}%
951    }%
952    \@tempa
953 }

```

\BibAbbrevWarning

```

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

```

\XrefWarning

```

955 \def\XrefWarning#1{\amsrefs@warning{Xref '#1' undefined}}

```

6.12.4 Bib field preprocessing

\current@primary

```

956 \let\current@primary\@empty

```

\previous@primary

```

957 \let\previous@primary\@empty

```

\save@primary

```

958 \IfOption{nobysame}{%
959   \let\save@primary\@empty
960 }{%
961   \def\save@primary{%
962     \global\let\previous@primary\current@primary
963   }%
964 }

```

\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.

```

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

```

```

993     \fi
994   \else
995     \bib@parsedate\bib'date
996   \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.

```

997   \def\@tempa{year}%
998   \ifx\bib'volume\@tempa
999     \let\bib'volume\bib@year
1000   \let\bib'date\@empty
1001   \fi

```

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

```

1002   \ifx\setbib@@\setbib@article
1003     \ifx\bib'volume\@empty
1004       \ifx\bib'number\@empty\else
1005         \let\bib'volume\bib'number
1006         \let\bib'number\@empty
1007       \fi
1008     \fi
1009   \fi

```

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

```

1010   \bib@language@fixup
1011 }

```

6.12.5 Date setup

`\bib@year`

```

1012 \let\bib@year\@empty

```

`\bib@month`

```

1013 \let\bib@month\@empty

```

`\bib@day`

```

1014 \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.

```

1015 \def\bib@parsedate#1{%
1016   \@xp\bib@parsedate@a#1---\@nil
1017 }

```

`\bib@parsedate@a`

```

1018 \def\bib@parsedate@a#1-#2-#3-#4\@nil{%
1019   \def\bib@year{#1}%
1020   \def\bib@month{#2}%
1021   \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.

```

1022   \ifx\@empty\bib@day
1023     \ifx\@empty\bib@month
1024       \let\bib'date\bib@year
1025     \else
1026       \def\bib'date{#1-#2}%

```

```

1027     \fi
1028   \else
1029     \def\bib'date{#1-#2-#3}%
1030   \fi
1031 }

```

6.12.6 Language setup

`\bib@language@fixup`

```

1032 \def\bib@language@fixup{%
1033   \ifx\bib'hyphenation\@empty
1034     \ifx\bib'language\@empty
1035       \let\bib@language\biblanguagedefault
1036     \else
1037       \let\bib@language\bib'language
1038     \fi
1039   \else
1040     \let\bib@language\bib'hyphenation
1041   \fi
1042   \def\@tempa##1 ##2\@nil{\lowercase{\def\bib@language{##1}}}%

```

The mysterious `\@firstofone` here is to preserve the space before the `\@nil`.

```

1043   \@firstofone{\@xp\@tempa\bib@language} \@nil
1044 }

```

`\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.

```

1045 \def\bib@selectlanguage{%
1046   \@ifsame@patterns{\language}{\bib@language}{-}{-}%
1047   \ifundefined{date\bib@language}{-}{-}%
1048   \@xp\selectlanguage\@xp{\bib@language}%
1049   }%
1050 }%
1051 }

```

`\@ifsame@patterns`

```

1052 \def\@ifsame@patterns#1#2{%
1053   \@xp\@ifsamepat\csname l@#1\@xp\endcsname\csname l@#2\endcsname
1054 }

```

`\@ifsamepat`

```

1055 \def\@ifsamepat#1#2{%
1056   \ifnum \ifx\relax#1\m@ne\else#1\fi = \ifx\relax#2\m@ne\else#2\fi
1057   \@xp\@firstoftwo
1058   \else
1059     \@xp\@secondoftwo
1060   \fi
1061 }

```

`\language`

`\biblanguageEnglish`

`\biblanguagedefault`

`\bib@language`

```

1062 \providecommand{\language}{english}
1063 \def\biblanguageEnglish{english}
1064 \let\biblanguagedefault\biblanguageEnglish
1065 \let\bib@language\@empty

```

6.12.7 Citation label setup`\generate@label``1066 \let\generate@label\relax``\cite@label``1067 \def\cite@label{\@currentlabel}``\alpha@label``1068 \let\alpha@label\relax``\alpha@label@`

```

1069 \def\alpha@label@{%
1070     \ifx\@empty\bib'label
1071         \def\thebib{\CurrentBib}%
1072     \else
1073         \let\thebib\bib'label
1074     \fi
1075 }%
```

`\amsrefs@option@numeric`

```

1076 \def\amsrefs@option@numeric{%
1077     \let\alpha@label\relax
1078     \let\generate@label\relax
1079     % \nmbrlisttrue
1080 }
```

`\amsrefs@option@alphabetic`

```

1081 \def\amsrefs@option@alphabetic{%
1082     \let\alpha@label\alpha@label@
1083     \let\generate@label\generate@alphalabel
1084     \let\calc@author@part\calc@author@part@
1085     \let\@suffix@format\@alph
1086     \let\append@label@year\append@label@year@
1087     % \nmbrlistfalse
1088 }
```

`\amsrefs@option@shortalphabetic`

```

1089 \def\amsrefs@option@shortalphabetic{%
1090     \let\alpha@label\alpha@label@
1091     \let\generate@label\generate@alphalabel
1092     \let\calc@author@part\calc@author@part@short
1093     \let\@suffix@format\@arabic
1094     \let\append@label@year\@empty
1095     % \nmbrlistfalse
1096 }
```

`\bib@cite` When `\bib@cite` is called, author name and year are available in `\bib@author` and `\bib@year`.

Arguments:

`#1` <- *citekey*.

```

1097 \def\bib@cite#1{%
1098     \def\CurrentBib{#1}%
1099     \alpha@label           % modify \thebib if necessary
1100     \item\leavevmode
1101     \SK@\SK@@label{#1}%
1102     \@xp\bib@cite@a\csname b@#1\endcsname
1103     \bibcite@write{#1}%
1104 }
```

```

1105 \def\bib@cite@a#1{%
1106     \ifx\relax#1%
1107         \begingroup
1108             \auto@protect\etaltext
1109             \protected@edef\@tempa{%
1110                 \gdef\@nx#1{%
1111                     \@nx\citesel 01{\cite@label}{\bib@label@year}{}%
1112                 }%
1113             }%
1114         \@xp@endgroup
1115     \@tempa
1116 \else
1117     \@xp\bib@cite@check\@xp#1#1\@empty\@empty\@empty\@empty\@empty
1118 \fi
1119 }

```

`\bib@cite@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.

```

```

1120 \def\bib@cite@check#1#2#3#4#5#6#7{%
1121     \ifx 1#4\relax
1122         \DuplicateBibKeyWarning
1123     \else

```

This has gotten *way* out of hand.

```

1124     \begingroup
1125         \auto@protect\etaltext
1126         \@apply\auto@protect\amsrefs@textsymbols
1127         \@apply\auto@protect\amsrefs@textaccents
1128         \@tempswafalse
1129         \in@CitePrintUndefined{#5}%
1130         \ifin@
1131             \let\@tempa\@empty
1132         \else
1133             \def\@tempa{#5}%
1134         \fi
1135         \ifx\@tempa\@empty
1136         \else
1137             \@xp\ifx\@xp\@currentlabel\cite@label
1138                 \edef\@tempb{\cite@label}%
1139             \else
1140                 \let\@tempb\cite@label
1141             \fi
1142             \ifx\@tempa\@tempb
1143                 \def\@tempa{#6}%

```

```

1144             \ifx\@tempa\bib@label@year
1145             \else
1146                 \@tempswatrue
1147             \fi
1148         \else
1149             \@tempswatrue
1150         \fi
1151     \fi
1152     \if@tempswa
1153         \@ifempty{#6}{%
1154             \def\@tempa{#5}%
1155             \let\@tempb\cite@label
1156         }{%
1157             \def\@tempa{#5, #6}%
1158             \def\@tempb{\cite@label, \bib@label@year}%
1159         }%
1160         \amsrefs@warning{Citation label for \extr@cite#1 is
1161             changing from \@tempa ' to \@tempb '%}
1162     \fi
1163     \protected@edef\@tempa{%
1164         \gdef\@nx#1{%
1165             \@nx\citesel #31{\cite@label}{\bib@label@year}{#7}%
1166         }%
1167     }%
1168     \@xp\endgroup
1169     \@tempa
1170 \fi
1171 }

```

\bib@label@year

```
1172 \let\bib@label@year\@empty
```

\DuplicateBibKeyWarning

```

1173 \def\DuplicateBibKeyWarning{%
1174     \amsrefs@warning{%
1175         Duplicate \protect\bib\space key
1176         \@CurrentBib ' detected\MessageBreakNS}%
1177 }

```

\DuplicateBibLabelWarning

```

1178 \def\DuplicateBibLabelWarning{%
1179     \amsrefs@warning{%
1180         Duplicate biblabel stem \@current@stem ' detected.\MessageBreakNS
1181         This usually means the order of the bibitems\MessageBreakNS
1182         is incompatible with the style of labels\MessageBreakNS
1183         you are using}%
1184 }

```

\bibcite@write

```

1185 \def\bibcite@write#1{%
1186     \if@filesw
1187         \begingroup
1188             \let\citesel\citesel@write
1189             \csname b@#1\endcsname
1190         \endgroup
1191     \fi
1192 }

```

\citesel@write


```

1193 \def\citesel@write#1#2#3#4#5{%
1194     \toks@{{#3}{#4}}}%
1195     \immediate\write\@auxout{\string\bibcite{\CurrentBib}{\the\toks@}}%
1196 }

```

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

```
1197 \AtEndDocument{\let\bibcite\gobbletwo}
```

6.12.8 Printing the bibliography

`\bibname`

```
1198 \providecommand{\bibname}{Bibliography}
```

`\refname`

```
1199 \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.

```
1200 \let\bib@div@mark\gobble
```

This is verbose, but probably safer than any alternative.

```

1201 \@ifclassloaded{amsbook}{}{%
1202     \@ifclassloaded{amsart}{}{%
1203         \@ifclassloaded{amsproc}{}{%
1204             \def\bib@div@mark#1{%
1205                 \@mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}%
1206             }%
1207         }%
1208     }%
1209 }

```

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

```

1210 \newenvironment{bibchapter}[1][\bibname]{%
1211     \begingroup
1212         \protected@edef\@{%
1213             \endgroup
1214             \protect\chapter*{#1}%
1215             \protect\bib@div@mark{#1}%
1216         }%
1217     \@
1218 }{\par}

```

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

```

1219 \newenvironment{bibsection}[1][\refname]{%
1220     \begingroup
1221         \protected@edef\@{%
1222             \endgroup
1223             \ifx\@bibtitlestyle\undefined
1224                 \protect\section*{#1}%
1225             \else
1226                 \protect\@bibtitlestyle
1227             \fi
1228             \protect\bib@div@mark{#1}%
1229         }%
1230     \@
1231 }{\par}

```

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

```

1232 \@ifundefined{chapter}{%
1233   \newenvironment{bibdiv}{\bibsection}{\endbibsection}
1234 }{%
1235   \newenvironment{bibdiv}{\bibchapter}{\endbibchapter}
1236 }
```

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

```

1237 \renewenvironment{thebibliography}[1]{%
1238   \bibdiv
1239   \biblist[\resetbiblist{#1}]%
1240 }{%
1241   \endbiblist
1242   \endbibdiv
1243 }
```

6.13 Name, journal and publisher abbreviations

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

\DefineName

```

1244 \newcommand{\DefineName}[2]{%
1245   \bib*{#1}{name}{name={#2}}%
1246 }
```

\DefineJournal

```

1247 \newcommand{\DefineJournal}[4]{%
1248   \bib*{#1}{periodical}{
1249     issn={#2},
1250     journal={#4}
1251   }%
1252 }
```

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

```

1253 \newcommand{\DefinePublisher}[4]{%
1254   \bib*{#1}{publisher}{%
1255     publisher={#3},
1256     address={#4}
1257   }%
1258 }
```

6.14 Processing .ltb 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

```

1259 \newcommand{\bibselect}{%
1260   \@ifstar{%
1261     \let\@bibdef\copy@bibdef
1262     \BibSelect
1263   }{%
1264     \let\@bibdef\selective@bibdef
1265     \BibSelect
1266   }%
1267 }
```

\BibSelect

```

1268 \newcommand{\BibSelect}[2][\bblname]{%
1269   \if@files
1270     \typeout{Trying to create bbl file `#1.bbl' ...}%
1271     \def\bibselect@msg{%
1272       \typeout{ ... rats. Unable to create bbl file.}%
1273     }%
1274     \let\@open@bbl@file\OpenBBLFile
1275     \@for\@tempa:=#2\do{\ReadBibData{\@tempa}}%
1276   \fi
1277   \@close@bbl@file
1278   \@apply\g@undef\bibdefer@list
1279   \global\let\bibdefer@list\@empty
  Now read the .bbl file we just created.
1280   \let\@bibdef\normal@bibdef
1281   \@input{#1.bbl}%
1282   \let\BibSelect\MultipleBibSelectWarning
1283 }
```

\MultipleBibSelectWarning

```

1284 \newcommand\MultipleBibSelectWarning[2][{%
1285   \amsrefs@warning{%
1286     Multiple \string\bibselect 's found (only one
1287     \string\bibselect\space per biblist environment is allowed)%
1288   }%
1289 }
```

\bblname

```

1290 \def\bblname{\jobname}
```

\bib@dbfile

```

1291 \newread\bib@dbfile
```

\ReadBibData

```

1292 \newcommand{\ReadBibData}[1]{%
1293   \IfFileExists{#1.ltb}{%
1294     \openin\bib@dbfile=\@filef@und \relax
1295   }{%
1296     \IfFileExists{#1.ltx}{%
1297       \openin\bib@dbfile=\@filef@und \relax
1298     }{%
1299       \IfFileExists{#1.tex}{%
1300         \openin\bib@dbfile=\@filef@und \relax
1301       }{%
1302         \begingroup
1303           \NoBibDBFile{#1}%
1304           \let\ReadBibData@a\endgroup

```

```

1305         }%
1306     }%
1307 }%
1308 \ReadBibData@a
1309 }

```

\NoBibDBFile

```

1310 \def\NoBibDBFile#1{%
1311     \amsrefs@warning{No data file #1.ltb (.ltx, .tex) found}%
1312 }

```

\ReadBibData@a

```

1313 \def\ReadBibData@a{%
1314     \ProvidesFile{\@filef@und}\relax
1315     \begingroup
1316         \let\star@bibdef\defer@bibdef
1317         \ReadBibLoop
1318     \endgroup
1319     \closein\bib@dbfile
1320 }

```

\ReadBibLoop

```

1321 \def\ReadBibLoop{%
1322     \ifeof\bib@dbfile
1323         \@xp\@gobble
1324     \else
1325         \read\bib@dbfile to\CurLine

```

The \@empty is in case \CurLine is empty.

```

1326         \@xp\ReadBibLoop@a\CurLine\@empty\@nil
1327     \fi
1328     \ReadBibLoop
1329 }

```

\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.

```

1330 \long\def\ReadBibLoop@a#1#2\@nil{%
1331     \ifx\bib#1%
1332         \CurLine % just exec it
1333     \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.

```

1334         \ifx\endinput#1%
1335             \let\ReadBibLoop\@empty
1336         \else

```

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

```

1337         \@xp\ReadBibLoop@b#1#2\@empty\@nil
1338     \fi
1339 \fi
1340 }

```

`\ReadBibLoop@b`

```

1341 \long\def\ReadBibLoop@b#1#2\@nil{%
1342     \ifx\bib#1%
1343         \CurLine % just exec it
1344     \fi
1345 }

1346 \let\bbl@out=\relax
1347 \let\bbl@write\@gobble
1348 \let\@open@bbl@file\relax
1349 \let\@close@bbl@file\relax

```

`\OpenBBLFile`

```

1350 \def\OpenBBLFile{%
1351     \if@filesw
1352         % Just use the next unused output stream
1353         \count@\count17
1354         \advance\count@\@ne
1355         \ifnum\count@<\sist@\@n
1356             \global\chardef\bbl@out=\count@
1357             \immediate\openout\bbl@out=\bblname.bbl\relax
1358             \global\let\@close@bbl@file\CloseBBLFile
1359             \gdef\bbl@write{\immediate\write\bbl@out}%
1360         \else
1361             \ch@ck\count@\sist@\@n\write
1362         \fi
1363     \fi
1364     \global\let\@open@bbl@file\relax
1365 }

```

`\CloseBBLFile`

```

1366 \def\CloseBBLFile{%
1367     \immediate\closeout\bbl@out\relax
1368     \global\let\@close@bbl@file\relax
1369     \global\let\bbl@write\@gobble
1370     \global\let\bbl@out\relax
1371 }

```

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 \LaTeX 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
1372 \def\citesel@update#1#2#3#4#5#6{%
1373   \gdef#6{\citesel 1#2#{3}-{#4}-{#5}}%
1374 }
```

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

```
\citesel@year
1376 \def\citesel@year#1#2#3#4#5#{#4}
```

```
\citesel
1377 \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^AT_EX definition of \cite_l prevents certain problems that could arise with the showkeys package.

```
1378 \expandafter\let\csname cite \endcsname\relax
```

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

```
1379 \renewcommand{\cite}[2][ ]{%
1380   \if\cite@single#2,\@gobble \else\MultipleCiteKeyWarning{#2}{#1}\fi
1381   \@ifempty{#1}{%
1382     \cites@o{#2}%
1383   }{%
1384     \ObsoleteCiteOptionWarning
1385     \cites@a{*{#1}}{#2}%
1386   }%
1387 }
```

\MultipleCiteKeyWarning

```
1388 \def\MultipleCiteKeyWarning#1#2{%
1389   \amsrefs@warning{%
1390     Use of \string\cites\space is recommended instead of %
1391     \string\cite\space\MessageBreak
1392     for multiple cites '#1'%
1393   \@ifnotempty{#2}{%
1394     \amsrefs@warning{Star option requires \string\citelist\space here}%
1395   }%
1396   \global\let\MultipleCiteKeyWarning\@gobbletwo
1397 }
```

\ObsoleteCiteOptionWarning

```
1398 \def\ObsoleteCiteOptionWarning{%
1399   \amsrefs@warning{%
1400     The form \string\cite{...}*{...} is recommended\MessageBreak
1401     instead of \string\cite[...]{...}%
1402   \global\let\ObsoleteCiteOptionWarning\@empty
1403 }
```

```

\cite@single
1404 \edef\cite@single#1,#2{\iffalse{\fi\iffalse{\fi\string}#2.\string}}

\cites@o
1405 \def\cites@o#1{\star@\{\cites@oo{#1}\}\{}}

\cites@oo
1406 \def\cites@oo#1#2{\@ifempty{#2}{\cites@a{#1}}{\cites@a{#{#2}}{#1}}}

\cites@a
1407 \def\cites@a#1#2{%
1408     \begingroup
1409     \toks@{\endgroup \cites@b{#1}}%
1410     \vdef\@tempa{#2}%
1411     \edef\@tempa{%
1412         \the\toks@ \@firstofone{\@xp\zap@space\@tempa} \@empty
1413     }%
1414     \@tempa,\@empty
1415     \edef\@tempa{\endgroup\@nx\citelist{\the\toks@}}%
1416     \@tempa
1417 }

\cites@b
1418 \def\cites@b#1#2,#3{%
1419     \begingroup
1420     \toks@{\InnerCite{#2}#1}%
1421     \ifx\@empty#3\@xp@gobble\fi
1422     \cites@c#3%
1423 }

\cites@c
1424 \def\cites@c#1,#2{%
1425     \add@toks@{\InnerCite{#1}}%
1426     \ifx\@empty#2\@xp@gobble\fi
1427     \cites@c#2%
1428 }

\citeleft These variables are named to follow the precedent set by Arseneau’s cite pack-
\citeright age. \citimid is used to separate a citation label from additional information
\citimid such as “Theorem 4.9”. \citepunct is used to separate multiple cites, unless one
\citepunct of the cites has additional associated information, in which case \CiteAltPunct
is used.
1429 \def\citeleft{[]}
1430 \def\citeright{[]}
1431 \def\citimid{\penalty9999 \space}
1432 \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 \citimid by \CiteAltPunct.
1433 \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 for-
matting of \cite’s optional argument. This is probably a bug.
1434 \providecommand{\citeform}{\@firstofone}

```

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

```
1435 \DeclareRobustCommand{\citelist}{\@citelist}
```

`\@citelist`

```
1436 \def\@citelist#1{%
1437   \leavevmode
1438   \begingroup
1439     \@citestyle
1440     \citeleft\nopunct    % suppress first \citepunct
1441     \cite@begingroup
1442       \in@*{#1}%
1443       \ifin@
1444         \let\citepunct\citeAltPunct
1445       \fi
1446       \let\cite@endgroup\@empty
1447       \cites@init
1448       \def\citeleft{\@addpunct{\citepunct}}%
1449       \let\citeright\ignorespaces
1450       \def\cite{\InnerCite}%
1451       \process@citelist{#1}%
1452     \endgroup
1453     \citeright
1454   \endgroup
1455 }
```

`\@citestyle` 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).

```
1456 \providecommand{\@citestyle}{\m@th\upshape\mdseries}
```

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

```
1457 \def\cite@begingroup{\begingroup\let\cite@begingroup\relax}
```

`\cite@endgroup`

```
1458 \let\cite@endgroup\endgroup
```

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

```
1459 \def\cites@init{%
1460   \gdef\prev@names{???}%
1461   \let\cites@init\@empty
1462 }
```

`\InnerCite`

```
1463 \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`.

```
1464 \def\cite@a#1#2{%
1465   \BackCite{#2}%
1466   \cite@begingroup
1467     \cites@init
1468     \let\citesel#1\relax
1469     \ifx\citesel\citesel@author
```



```

1470         \let\citeleft\@empty
1471         \let\citeright\@empty
1472     \fi
1473     \begingroup
1474         \toks@{\endgroup \cite@b}%
1475         \vdef\@tempa{#2}%
1476         \edef\@tempa{%
1477             \the\toks@{\@firstofone{\@xp\zap@space\@tempa} \@empty}}%
1478         }%
1479     \@tempa
1480 }

```

`\cite@b` *Arguments:*

```

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

```

```

1481 \def\cite@b#1#2{%
1482     \@xp\cite@bc\csname b@#1\@xp\endcsname {#1}{#2}%
1483 }

```

`\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`.

```

1484 \def\cite@bc#1#2{%
1485     \ifx#1\@undefined \global\let#1\relax \fi
1486     \ifx#1\relax \global\let#1\empty@cite \fi
1487     \@xp\cite@nobib@test#1{}{}{}{}\@nil#1%
1488     \cite@cj#1%
1489 }

```

`\empty@cite`

```

1490 \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.

```

```

1491 \def\cite@nobib@test#1#2#3#4#5\@nil#6{%
1492     \@ifempty{#4}{%
1493         \G@refundefinedtrue
1494         \UndefinedCiteWarning#6%
1495         \xdef#6{\@nx\citesel #2#3{%
1496             \@nx\CitePrintUndefined{\extr@cite#6}}{}{}}%
1497     }{}%
1498 }

```

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

```

1499 \def\UndefinedCiteWarning#1{%
1500     \@latex@warning{%
1501         Citation `~\extr@cite#1' on page \thepage\space undefined}%
1502 }

```

```

\CitePrintUndefined
1503 \DeclareRobustCommand{\CitePrintUndefined}[1]{%
1504     \begingroup\fontshape{n}\fontseries\mddefault \ttfamily ?#1\endgroup
1505 }

\CPU@normal This has to be a \let, not a \def.
1506 \let\CPU@normal\CitePrintUndefined

\cite@cj Arguments:
        #1 <- \b@citekey.
        #2 <- star-optional-arg.

1507 \def\cite@cj#1#2{%
1508     \leavevmode
1509     \begingroup
1510         \cite@cb#1% write info to aux file
1511         \ar@SK@cite#1%
1512         \@citeleft
1513         \ar@hyperlink{#1}%
1514         \@ifnotempty{#2}{\citelink{#2}}%
1515         \citeright
1516     \endgroup
1517     \ignorespaces % ignore spaces inside \citelist
1518     \cite@endgroup
1519 }

\@citeleft The following definition provides some indirection that helps to deal with
author-year object cites.
1520 \def\@citeleft{\citeleft}

\cite@cb
1521 \def\cite@cb#1{%
1522     \if@filesw
1523         \immediate\write\@auxout{\string\citation{\extr@cite#1}}%
1524     \fi
    Define \citesel to make \b@whatever update itself.
1525     \begingroup
1526         \let\citesel\citesel@update
1527         #1#1%
1528     \endgroup
1529 }

\extr@cite Extract citekey from \b@citekey.
1530 \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.)

```
1531 \DeclareRobustCommand{\cites}{\cites@a{}}
```

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

```
1532 \providecommand{\citen}{\ocite}
```

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

```
1533 \def\ycite{\cite}
```

\ycites

1534 \let\ycites\cites

\ocite

1535 \let\ocite\ycite

\ocites

1536 \let\ocites\cites

\fullcite

1537 \let\fullcite\cite

\fullocite

1538 \let\fullocite\ocite

\citeauthor

1539 \let\citeauthor\ycite

\citeauthory

1540 \let\citeauthory\ycite

6.15.4 The \nocite command

\nocite

1541 \renewcommand{\nocite}[1]{\othercites{#1}}

\othercites

```
1542 \newcommand{\othercites}[1]{%
1543   \cite@begingroup
1544     \let\BackCite\@gobble
1545     \let\cite@endgroup\@empty
1546     \def\citelist{\othercitelist}%
1547     \cites{#1}%
1548 }
```

\othercitelist

```
1549 \newcommand{\othercitelist}[1]{%
1550   \cite@begingroup
1551     \let\cite@endgroup\@empty
1552     \cites@init
1553     \let\citeleft\relax
1554     \let\citeright\ignorespaces
1555     \def\InnerCite{\OtherCite}%
1556     \def\cite@cj ##1##2{%
1557       \begingroup
1558         \@xp\citesel##1%
1559         \cite@cb ##1%
1560       \endgroup
```

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

```
1561     \@xp\ifx\csname b@*\endcsname ##1%
1562     \global\let\selective@bibdef\copy@bibdef
1563     \fi
1564     \ignorespaces
1565     \cite@endgroup
1566   }%
1567   #1\relax
1568 \endgroup
1569 }
```

`\OtherCite`

```
1570 \def\OtherCite#1{\cite@a\citesel@other{#1}{}}{}}
```

`\citesel@other`

```
1571 \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.

```
1572 \@namedef{b@*}{\citesel 11{*}{*}{*}}
```

6.15.5 Citation sorting

`\process@citelist@sorted`

```
1573 \def\process@citelist@sorted#1{%
1574   \ifx\citesel\citesel@number
1575     \cite@sorted@s #1\cite@sorted@e
1576   \else
1577     \NonNumericCiteWarning
1578     \process@citelist@unsorted{#1}%
1579   \fi
1580 }
```

`\NonNumericCiteWarning`

```
1581 \def\NonNumericCiteWarning{%
1582   \amsrefs@warning{%
1583     Unable to confirm that cite keys are numeric: not sorting%
1584   }%
1585 }
```

`\process@citelist@unsorted`

```
1586 \def\process@citelist@unsorted#1{%
1587   \ignorespaces#1\relax
1588 }
```

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

```
1589 \let\process@citelist\process@citelist@sorted
```

`\CPU@sort` By defining this as \TeX 's `maxint`, undefined cites migrate to the end of a sorted list.

```
1590 \def\CPU@sort#1{2147483647}
```

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

```
1591 \def\cite@sorted@s{%
1592   \begingroup
1593     \let\CitePrintUndefined\CPU@sort
1594     \let\cite@cjs\cite@cj
1595     \let\cite@cj\cite@compress
1596   \begingroup
1597     \toks@{@emptytoks
1598     \let\cite@cj\cite@sort
1599     \ignorespaces
1600 }
```

`\cite@sorted@e`

```
1601 \def\cite@sorted@e{%
1602   \@xp\endgroup
1603   \the\toks@
1604   \cite@dash
```

```

1605      \prev@cite
1606  \endgroup
1607 }

```

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

Arguments:

```

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

```

```

1608 \def\cite@sort#1#2{%
1609   \safe@set\@tempcnta#1% highest number so far
1610   \toks@{\cite@cj#1{#2}}%
1611   \@temptokena\toks@
1612   \let\cite@cj\cite@sort@a
1613   \ignorespaces
1614 }

```

\cite@sort@a

```

1615 \def\cite@sort@a#1#2{%
1616   \safe@set\@tempcntb#1%
1617   \ifnum\@tempcntb > \@tempcnta
1618     \add@toks@{\cite@cj#1{#2}}%
1619     \@tempcnta\@tempcntb
1620   \else
1621     \let\cite@cj\cite@sort@b
1622     \toks@\@emptytoks
1623     \def\@tempb{\add@toks@{\cite@cj#1{#2}}}%
1624     \the\@temptokena
1625     \@tempb
1626     \let\cite@cj\cite@sort@a
1627   \fi
1628   \@temptokena\toks@
1629   \ignorespaces
1630 }

```

\cite@sort@b

```

1631 \def\cite@sort@b#1#2{%
1632   \safe@set\count@#1%
1633   \ifnum\@tempcntb < \count@
1634     \@tempb
1635     \let\@tempb\@empty
1636   \fi
1637   \add@toks@{\cite@cj#1{#2}}%
1638   \ignorespaces
1639 }

```

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

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

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`

1640 `\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).

```

1641 \def\prev@cite@cb{%
1642   \ifx\@prev@cite\@empty
1643   \else
1644     \begingroup
1645       \def\cite@print##1##2{%
1646         \cite@cb##1%
1647       }%
1648       \prev@cite
1649     \endgroup
1650   \fi
1651 }
```

`\cite@print`

```
1652 \def\cite@print#1#2{%
1653     \begingroup
1654         \let\CitePrintUndefined\CPU@normal
1655         \cite@cjs#1{#2}%
1656     \endgroup
1657 }
```

`\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.

```
1658 \let\cite@dash\@empty
```

`\print@one@dash`

```
1659 \def\print@one@dash{%
1660     \textendash \nopunct
1661     \let\cite@dash\@empty
1662 }
```

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:

```
1663 \def\cite@compress#1#2{%
1664     \cite@print#1{#2}%
1665     \@ifempty{#2}{%
1666         \safe@set\@tempcnta#1%
1667         \let\cite@cj\cite@compress@a
1668     }{}%
1669 }
```

`\cite@compress@a` State 1:

```
1670 \def\cite@compress@a#1#2{%
1671     \@ifempty{#2}{%
1672         \advance\@tempcnta\@ne
1673         \safe@set\@tempcntb#1%
1674         \ifnum\@tempcnta=\@tempcntb
1675             \def\prevcite{\cite@print#1{}}%
1676             \let\cite@cj\cite@compress@b
1677         \else
1678             \cite@print#1{}%
1679             \@tempcnta\@tempcntb
1680         \fi
1681     }{%
1682         \cite@print#1{#2}%
1683         \let\cite@cj\cite@compress
1684     }%
1685 }
```

`\cite@compress@b` State 2:

```
1686 \def\cite@compress@b#1#2{%
1687     \@ifempty{#2}{%
1688         \advance\@tempcnta\@ne
1689         \safe@set\@tempcntb#1%
```

```

1690      \ifnum\@tempcnta=\@tempcntb
1691      \let\cite@dash\print@one@dash
1692      \prev@cite@cb
1693      \def\prev@cite{\cite@print#1{}}%
1694      \else
1695      \cite@dash
1696      \prev@cite
1697      \let\prev@cite\@empty
1698      \cite@print#1{}%
1699      \@tempcnta\@tempcntb
1700      \let\cite@cj\cite@compress@a
1701      \fi
1702  }{%
1703      \cite@dash
1704      \prev@cite
1705      \let\prev@cite\@empty
1706      \cite@print#1{#2}%
1707      \let\cite@cj\cite@compress
1708  }%
1709 }

```

6.15.7 Munging the .aux file

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

```

1710 \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.

```

1711 \AtBeginDocument{%
1712   \if@filesw
1713     \immediate\write\@auxout{%
1714       \string\@ifundefined{amsrefs@bibcite}{}{}%
1715       \string\let\string\bibcite\string\amsrefs@bibcite
1716     }%
1717   }%
1718   \fi

```

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

```

1719   \let\bibcite\amsrefs@bibcite
1720 }

```

`\bibcite@a` *Arguments:*

```

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

```

```

1721 \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.

```

1722   \ifx\relax#1%
1723     \gdef#1{\citesel 00#2{}}%
1724   \else
1725     \begingroup
1726       \@xp\bibcite@b\@xp#1#1{#2}%
1727     \endgroup
1728   \fi
1729 }

```



```

\cite@b Arguments:
    #1 <- \b@citekey.
    #2 <- \citesel.
    #3 <- cited?.
    #4 <- used?.
    #5 <- label.
    #6 <- year.
    #7 <- backrefs.
    #8 <- {newlabel}{newyear}.
1730 \def\cite@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 sup-
port.
1731 \newcommand{\citedest}[1]{\@xp\cite@dest\csname b@#1\endcsname}

\cite@dest
1732 \def\cite@dest#1{%
1733     \ifx\relax#1%
1734         \gdef#1{\citesel 00{}{}{}}%
1735     \fi
1736     \@xp\cite@dest@b\@xp#1#1%
1737 }

\cite@dest@b Arguments:
    #1 <- \b@citekey.
    #2 <- \citesel.
    #3 <- cited?.
    #4 <- used?.
    #5 <- label.
    #6 <- year.
    #7 <- backrefs.
    #8 <- {more backrefs}.
1738 \def\cite@dest@b#1#2#3#4#5#6#7#8{%
1739     \@ifempty{#7}{%
1740         \def#1{\citesel #3#4{#5}{#6}{#8}}%
1741     }{%
1742         \gdef#1{\citesel #3#4{#5}{#6}{#7,#8}}%
1743     }%
1744 }

```

6.15.8 Back references

```

\ifBR@verbose
1745 \@ifundefined{ifBR@verbose}{\let\ifBR@verbose\iffalse \let\fi\fi}{%

\BackCite
1746 \let\BackCite\@gobble

\back@cite
1747 \def\back@cite#1{%
1748     \ifBR@verbose
1749         \PackageInfo{backref}{back cite \string `extr@cite#1'}%
1750     \fi
1751     \Hy@backout{#1}%
1752 }

```

`\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.

```
1753 \def\print@backrefs#1{%
1754     \space\SentenceSpace$\uparrow$\csname br@#1\endcsname
1755 }
```

`\PrintBackRefs`

```
1756 \let\PrintBackRefs\@gobble
```

6.15.9 hyperref, showkeys and shaderef support

`\shade@cite`

```
1757 \newcommand{\shade@cite}{\printref}
```

`\format@cite`

```
1758 \def\format@cite#1{\shade@cite{\citeform{#1}}}
```

`\ar@hyperlink`

```
1759 \def\ar@hyperlink#1{%
1760     \hyper@link[cite]{\cite.\extr@cite#1}{\format@cite{#1}}%
1761 }
```

`\ar@SK@cite`

```
1762 \def\ar@SK@cite#1{\@bsphack\@xp\SK@\@xp\SK@@ref\@xp{\extr@cite#1}\@esphack}
```

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

```
1763 \AtBeginDocument{%
1764     \@ifpackageloaded{shaderef}{\@firstofone
1765         \let\shade@cite\@firstofone
1766     }%
1767     \@ifpackageloaded{hyperref}{\@firstofone
1768         \def\ar@hyperlink{\format@cite}%
1769         \let\hyper@anchorstart\@gobble
1770         \let\hyper@anchorend\relax
1771         \let\Hy@raisedlink\@firstofone
1772     }%
1773     \@ifpackageloaded{showkeys}{\@firstofone
1774         \@ifpackagewith{showkeys}{notcite}{\@firstofone
1775             \let\ar@SK@cite\@gobble
1776         }{}
1777     }{}%
1778     \let\ar@SK@cite\@gobble
1779     \let\SK@@label\@gobble
1780     \let\SK@\@gobbletwo
1781 }%
1782 }
```

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 `\` or `\k`,

- text symbol macros, such as `\i`, `\ae` or `\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 73).³

`\amsrefs@textaccents` This will contain a list of accent commands in standard L^AT_EX format (i.e., separated by the token `\do`). For example, after registering the `\"` and `\'` accents, it will contain

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

```
1783 \let\amsrefs@textaccents\@empty
```

`\DeclareNameAccent` *Arguments:*

```
#1 <- accent.
```

```
1784 \def\DeclareNameAccent{%
1785     \@lappend\amsrefs@textaccents
1786 }
```

Here are all the standard L^AT_EX accents, as well as a few nonstandard accents from the `mathscinet` package.

```
1787 \DeclareNameAccent\"
1788 \DeclareNameAccent\'
1789 \DeclareNameAccent\
1790 \DeclareNameAccent\=
1791 \DeclareNameAccent\^
1792 \DeclareNameAccent\`
1793 \DeclareNameAccent\~%
1794 \DeclareNameAccent\b
1795 \DeclareNameAccent\c
```

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

```

1796 \DeclareNameAccent\d
1797 \DeclareNameAccent\H
1798 \DeclareNameAccent\k
1799 \DeclareNameAccent\r
1800 \DeclareNameAccent\t
1801 \DeclareNameAccent\u
1802 \DeclareNameAccent\v

```

From mathscinet:

```

1803 \DeclareNameAccent\utilde
1804 \DeclareNameAccent\uarc
1805 \DeclareNameAccent\dudot
1806 \DeclareNameAccent\lfhook
1807 \DeclareNameAccent\udot
1808 \DeclareNameAccent\polhk
1809 \DeclareNameAccent\soft

```

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

```

1810 \DeclareNameAccent\etalchar
1811 %\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.⁴

`\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

```
\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`.

```
1812 \let\amsrefs@textsymbols\@empty
```

`\DeclareNameSymbol` *Arguments:*

```

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

```

```

1813 \def\DeclareNameSymbol#1#2{%
1814   \@lappend\amsrefs@textsymbols#1%
1815   \@lappend\amsrefs@textsymbols#2%
1816   \ifx#1#2\else
1817     \@lappend\amsrefs@textsymbols#2%
1818     \@lappend\amsrefs@textsymbols#2%
1819   \fi
1820 }

```

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

Here are the standard \LaTeX and `mathscinet` text symbols.

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

```
1821 \DeclareNameSymbol\i\i
1822 \DeclareNameSymbol\j\j
1823 \DeclareNameSymbol\AE\ae
1824 \DeclareNameSymbol\OE\oe
1825 \DeclareNameSymbol\O\o
1826 \DeclareNameSymbol\DH\dh
1827 \DeclareNameSymbol\DJ\dj
1828 \DeclareNameSymbol\L\l
1829 \DeclareNameSymbol\NG\ng
1830 \DeclareNameSymbol\SS\ss
1831 \DeclareNameSymbol\TH\th
```

From `mathscinet`:

```
1832 \DeclareNameSymbol\Dbar\dbar
1833 \DeclareNameSymbol\lasp\lasp
1834 \DeclareNameSymbol\rasp\rasp
1835 \DeclareNameSymbol\cprime\cprime
1836 \DeclareNameSymbol\cdprime\cdprime
1837 \DeclareNameSymbol\bud\bud
1838 \DeclareNameSymbol\cydot\cydot
```

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

```
1839 \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{'o}lya}
```

`\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.

```
1840 \def\normalize@edef#1#2{%
1841   \begingroup
1842     \@apply\auto@protect\amsrefs@textsymbols
1843     \@apply\wrap@accent\amsrefs@textaccents
```

Redefine `\@tabacckludge` in case someone wants to use this with the `inputenc` package.

```
1844     \let\@tabacckludge\use@accent
1845     \let~\space
1846     \edef\@tempa{\def\@nx#1{#2}}%
1847   \@xp\endgroup
1848   \@tempa
1849 }
```

`\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

```
1850 \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}`.

```
1851 \def\wrap@accent#1{%
1852   \def#1##1{{\@nx#1##1}}%
1853 }
```

`\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.

```
1854 \def\lc@edef#1#2{%
1855   \begingroup
1856     \let\@tabacckludge\use@accent %%??
1857     \@apply\auto@protect\amsrefs@textaccents
1858     \@apply\lc@do\amsrefs@textsymbols
1859     \edef\@tempa{\lowercase{\def\@nx#1{#2}}}%
1860   \@xp\endgroup
1861   \@tempa
1862 }
```

`\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.

```
1863 \def\lc@do#1\do#2{%
1864   \ifx#1#2%
1865     \auto@protect#1%
1866   \else
1867     \def#1{#2}%
1868   \fi
1869 }
```

`\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`.

```
1870 \def\purge@edef#1#2{%
1871   \begingroup
1872     \@apply\auto@protect\amsrefs@textsymbols
1873     \let-\space
1874     \@apply\purge@accent\amsrefs@textaccents
1875     \let\@tabacckludge\@gobble
```

As mentioned above (page 61), `\i` and `\j` are semantically like text accents; hence, they require special treatment here.

```
1876   \def\i{i}%
1877   \def\j{j}%
```

```

1878      \edef\@tempa{#2}%
1879      \toks@\@emptytoks
1880      \@xp\purge@edef@ \@tempa \@nil
1881      \edef\@tempa{\def\@nx#1{\the\toks@}}%
1882      \@xp\endgroup
1883      \@tempa
1884 }

```

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

```

1885 \def\purge@edef@{%
1886     \futurelet\@let@token
1887     \purge@edef@a
1888 }

```

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

```

1889 \def\purge@edef@a#1{%

```

If we’ve run into the `\@nil` terminator, we’re done.

```

1890     \ifx\@let@token\@nil
1891         \let\@tempa\@empty
1892     \else

```

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

```

1893         \ifx\@let@token\bgroup
1894             \def\@tempa{%
1895                 \purge@edef@ #1\@nil
1896                 \purge@edef@
1897             }%
1898         \else

```

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

```

1899             \add@toks@{#1}%
1900             \let\@tempa\purge@edef@
1901         \fi
1902     \fi
1903     \@tempa
1904 }

```

`\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).

```

1905 \def\purge@accent#1{%
1906     \def##1{##1}%
1907 }

```

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 59), 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.)

```
1908 \def\name@split{%
1909     \get@namepart\bib'surname\name@split@given
1910 }
```

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

```
1911 \def\name@split@given{%
1912     \get@namepart\bib'given\name@split@jr
1913 }
```

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

```
1914 \def\name@split@jr{%
1915     \get@namepart\bib'jr\name@split@finish
1916 }
```

`\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.

```
1917 \def\name@split@finish{%
1918     \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.

```
1919 %%     \ifx\@empty\bib'given
1920 %%         \NameCheck \bib'surname ??\@nil
1921 %%     \else
1922         \extract@initials\bib'given
1923 %%     \fi
1924     \@nilgobble
1925 }
```

`\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.

```
1926 \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.

```
1927     \toks@{#1}%
1928     \@temptokena{#2}%
1929     \get@namepart@a
1930 }
```

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

```
1931 \def\get@namepart@a{%
1932     \futurelet\@let@token
1933     \get@namepart@b
1934 }
```

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

```
1935 \def\get@namepart@b{%
1936     \ifx\@let@token\@sptoken
1937         \@xp\get@namepart@c
1938     \else
1939         \@xp\get@namepart@d
1940     \fi
1941 }
```

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

```
1942 \def\get@namepart@c{%
1943     \after@deleting@token\get@namepart@a
1944 }
```

`\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.

```
1945 \def\get@namepart@d{%
1946     \futurelet\@let@token
1947     \get@namepart@e
1948 }
```

`\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.

```
1949 \def\get@namepart@e{%
1950     \ifx\@let@token,%
1951         \@xp\let\the\toks@\@empty
1952         \edef\@tempa{%
1953             \@nx\after@deleting@token\the\@temptokena
1954         }%
1955         \@xp\@tempa
1956     \else
1957         \@xp\get@namepart@f
1958     \fi
1959 }
```

`\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.⁵

Note that the arguments are not expanded.

⁵More or less. If the second argument is brace-delimited, the braces will be lost. But as mentioned above (page 64), we don't really care.

```

1960 \def\get@namepart@f#1#2,{%
1961     \ifx\@let@token\bgroup
1962         \@xp\def\the\toks@{{#1}#2}%
1963     \else
1964         \@xp\def\the\toks@{{#1#2}%
1965     \fi
1966     \the\@temptokena
1967 }

```

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

```

1968 \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.}
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”.⁶

6.18.2 The implementation

`\extract@initials` This is pretty straightforward.

```
1969 \def\extract@initials#1{%
1970     \begingroup
1971     \auto@protect\ini@hyphen
1972     \auto@protect\nobreakspace
1973     \let~\relax
1974     \@apply\auto@protect\amsrefs@textsymbols
1975     \@apply\auto@protect\amsrefs@textaccents
1976     \normalize@edef\@tempa{#1}%
1977     \ifx\@tempa\empty
1978     \else
1979         \process@hyphens\@tempa
1980         \process@dots\@tempa
1981         \process@names\@tempa
1982         \chomp\@tempa{-}%
1983     \fi
1984     \edef\@tempa{\def\nx\bib'initials{\@tempa}}%
1985     \exp\endgroup
1986     \@tempa
1987 }
```

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.-”)

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

```
1988 \def\ini@hyphen{\unskip-\nobreak}
```

⁶Tying 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.

```
1989 \def\process@hyphens#1{%
1990     \begingroup
1991     \toks@{\@emptytoks
1992     \@xp\process@one@hyphen #1-\@nil
1993     \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.)

```
1994     \@chomp\@tempa{ \ini@hyphen\space}%
1995     \edef\@tempa{\def\@nx#1{\@tempa}}%
1996     \@xp\endgroup
1997     \@tempa
1998 }
```

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

```
1999 \def\process@one@hyphen{%
2000     \futurelet\@let@token
2001     \process@one@hyphen@a
2002 }
```

`\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 `\fsa@n` from `rkeyval` to keep track of the appropriate next action.

```
2003 \def\process@one@hyphen@a{%
2004     \ifx\@let@token\@nil
2005         \let\fsa@n\@gobble
2006     \else
2007         \ifx\@let@token -%
2008             \TrailingHyphenWarning
2009             \let\fsa@n\process@one@hyphen@b
2010         \else
2011             \ifx\@let@token\@sptoken
2012                 \let\fsa@n\process@one@hyphen@b
2013             \else
2014                 \let\fsa@n\process@one@hyphen@c
2015             \fi
2016         \fi
2017     \fi
2018     \fsa@n
2019 }
```

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

```
2020 \def\process@one@hyphen@b{%
2021     \after\deleting@token\process@one@hyphen
2022 }
```

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

```

2023 \def\process@one@hyphen@c#1#2-{%
2024     \ifx\bgroup\@let@token
2025         \add@toks@{#1}#2 \ini@hyphen\space}%
2026     \else
2027         \add@toks@{#1#2 \ini@hyphen\space}%
2028     \fi
2029     \futurelet\@let@token
2030     \process@one@hyphen@d
2031 }
```

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

```

2032 \def\process@one@hyphen@d{%
2033     \ifx\@let@token\@nil
2034         \@xp\@gobble
2035     \else
2036         \@xp\process@one@hyphen
2037     \fi
2038 }
```

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

```

2039 \def\TrailingHyphenWarning{%
2040     \amsrefs@warning{Trailing hyphen deleted from name}%
2041 }
```

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

```

2042 \def\process@dots#1{%
2043     \begingroup
2044         \toks@{\emptytoks
2045             \@xp\process@one@dot #1.\@nil
2046             \edef\@tempa{\the\toks@}%
2047             \@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.

```

2048         \@chomp\@tempa{ }%
2049         \edef\@tempa{\def\@nx#1{\@tempa}}%
2050     \@xp\endgroup
2051     \@tempa
2052 }
```

`\process@one@dot`

```

2053 \def\process@one@dot{%
2054     \futurelet\@let@token
2055     \process@one@dot@a
2056 }
```

`\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`.

```

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

```

```

2064         \let\fsa@n\process@one@dot@c
2065     \fi
2066 \fi
2067 \fsa@n
2068 }

\process@bare@dot
2069 \def\process@bare@dot{%
2070     \add@toks@{. }%
2071     \futurelet\@let@token
2072     \process@one@dot@d
2073 }

\process@one@dot@b
2074 \def\process@one@dot@b{%
2075     \after@deleting@token\process@one@dot
2076 }

\process@one@dot@c
2077 \def\process@one@dot@c#1#2.{%
2078     \ifx\bgroup\@let@token
2079         \add@toks@{{#1}#2. }%
2080     \else
2081         \add@toks@{#1#2. }%
2082     \fi
2083     \futurelet\@let@token
2084     \process@one@dot@d
2085 }

\process@one@dot@d
2086 \def\process@one@dot@d{%
2087     \ifx\@let@token\@nil
2088         \@xp\@gobble
2089     \else
2090         \@xp\process@one@dot
2091     \fi
2092 }

\process@names This is very similar to \process@hyphens and \process@dots, but with a cou-
                ple of twists, as noted below.
2093 \def\process@names#1{%
2094     \begingroup
2095         \toks@\@emptytoks
2096         \@xp\extract@initial #1 \@nil
2097         \edef\@tempa{\def\@nx#1{\the\toks@}}%
2098     \@xp\endgroup
2099     \@tempa
2100 }

\extract@initial Scan through the token stream replacing words by their initials until we hit the
                terminating '11
2101 \def\extract@initial{%
2102     \futurelet\@let@token
2103     \extract@initial@a
2104 }

\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.

```

2105 \def\extract@initial@a{%
2106     \ifx\@let@token\@nil
2107         \let\fsa@n\@gobble
2108     \else
2109         \ifx\@let@token\@sptoken
2110             \let\fsa@n\extract@initial@b
2111         \else
2112             \let\fsa@n\extract@initial@c
2113         \fi
2114     \fi
2115     \fsa@n
2116 }
```

`\extract@initial@b`

```

2117 \def\extract@initial@b{%
2118     \after@deleting@token\extract@initial
2119 }
```

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

```

2120 \def\extract@initial@c#1#2 {%
2121     \ifx\@let@token\bgroup
```

Note that we double-brace the first argument to avoid having to test `\@let@token` again inside `\@extract@initial`.

```

2122         \@extract@initial {{#1}}#2\@nil
2123     \else
2124         \@extract@initial #1#2\@nil
2125     \fi
2126     \futurelet\@let@token
2127     \extract@initial@d
2128 }
```

`\extract@initial@d`

```

2129 \def\extract@initial@d{%
2130     \ifx\@let@token\@nil
2131         \xp\@gobble
2132     \else
2133         \xp\extract@initial
2134     \fi
2135 }
```

`\@extract@initial` This handles the four cases mentioned on page 68.

```

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



```

2151    \fi
2152 }

```

`\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.

```

2153 \def\count@chars#1#2{%
2154     \begingroup
2155         \@tempcnta\z@
2156         \@count@chars#2\@nil
2157     \edef\@tempb{#1=\the\@tempcnta\relax}%
2158     \xp@endgroup
2159     \@tempb
2160 }

```

`\@count@chars`

```

2161 \def\@count@chars#1{%
2162     \ifx #1\@nil
2163     \else
2164         \advance\@tempcnta\@ne
2165         \@xp\@count@chars
2166     \fi
2167 }

```

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.⁷ 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`,

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

but the document author is responsible for ensuring the appropriate order if `amsxport` is not used. This is why it's an error to mix the `alphabetic` and `citation-order` options.

6.19.2 The implementation

```
2168 \let\previous@stem\@empty
```

```
2169 \let\current@stem\@empty
```

```
2170 \let\previous@year\@empty
```

```
2171 \let\current@year\@empty
```

```
\amsrefs@label@prefix
```

```
2172 \let\amsrefs@label@prefix\@empty
```

```
\thebib
```

```
2173 \def\thebib{\amsrefs@label@prefix\@arabic\c@bib}
```

```
\append@to@stem
```

```
2174 \def\append@to@stem{\global\@concat\current@stem}
```

```
\generate@alphalabel
```

```
2175 \def\generate@alphalabel{%
```

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

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

```
2177 \begingroup
```

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

```
2178 \global\let\previous@stem\current@stem
```

```
2179 \global\let\current@stem\amsrefs@label@prefix
```

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.

```
2180 \@apply\auto@protect\amsrefs@textsymbols
```

```
2181 \@apply\auto@protect\amsrefs@textaccents
```

```
2182 \auto@protect\name
```

```
2183 \auto@protect\etaltext
```

```
2184 \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.

```
2185 \get@numberof\@tempcnta\name\@tempa
```

```
2186 \calc@author@part
```

Next append the year part.

```
2187 \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.

```
2188 \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.)

```

2189         \edef\@tempa{%
2190             \def\@nx\bib'label{%
2191                 \current@stem
2192                 \alpha@label@suffix
2193             }%
2194         }%
2195         \@xp\endgroup
2196         \@tempa
2197     \fi
2198 }

```

\calc@author@part@

```

2199 \def\calc@author@part@{%
2200     \ifnum \@tempcnta = 1
2201         \@xp\oneauthorlabel\@xp{\@tempa}%
2202     \else
2203         \@xp\multiauthorlabel\@xp{\@tempa}%
2204     \fi
2205 }

```

\calc@author@part@short

```

2206 \def\calc@author@part@short{%
2207     \@xp\multiauthorlabel\@xp{\@tempa}%
2208 }%

```

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

```
2209 \def\@firstone#1{\@car#1\@empty\@nil}
```

\@firstthree And this extracts the first three characters.

```
2210 \def\@firstthree#1{\@carcube#1\@empty\@empty\@empty\@nil}
```

\@nametoken

```
2211 \let\@nametoken\@firstone
```

\hyph@to@space

```
2212 \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).

```

2213 \def\@marknames#1{%
2214     \@ifnotempty{#1}{\surround@names#1 ' }%
2215 }

```

\surround@names

```

2216 \def\surround@names#1 {%
2217     \ifx '#1%
2218     \else
2219         \@nx\@nametoken{#1}%
2220         \@xp\surround@names
2221     \fi
2222 }

```

\extract@surnames

```

2223 \def\extract@surnames#1#2{%
2224     \get@namepart\@tempb\@nilgobble #2,\@nil
2225     \edef\@tempb{\@nx\@marknames{\@xp\hyph@to@space\@tempb\@gobble-}}%
2226     \edef#1{\@tempb}%
2227 }

```

`\@oneauthorlabel` This is the easy case.

```

2228 \newcommand{\@oneauthorlabel}[1]{%
2229     \def\name##1{%
2230         \extract@surnames\@tempa{##1}%
2231         \get@numberof\@tempcnta\@nametoken\@tempa
2232         \ifnum \@tempcnta = 1
2233             \let\@nametoken\@firstthree
2234         \fi
2235         \append@to@stem{\@tempa}%
2236     }%
2237     #1%
2238 }
```

`\@threeauthors`

```

2239 \def\@threeauthors\name#1\name#2\name#3#4\@empty{%
2240     \name{#1}\name{#2}\name{#3}%
2241     \append@to@stem{\etalchar{+}}%
2242 }
```

`\@multiauthorlabel`

```

2243 \newcommand{\@multiauthorlabel}[1]{%
2244     \def\name##1{%
2245         \ifx\etaltext ##1%
2246             \def\@tempa{\@nx\etalchar{+}}%
2247             \let\name\@gobble
2248         \else
2249             \extract@surnames\@tempa{##1}%
2250         \fi
2251         \append@to@stem{\@tempa}%
2252     }%
2253     \ifnum \@tempcnta > 4 \@xp \@threeauthors \fi
2254     #1\@empty
2255 }
```

`\etalchar`

```

2256 \newcommand{\etalchar}[1]{${\sim}^{#1}$}
```

`\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.

```

2257 \def\year@short#1#2#3#4\@nil{#3#4}
```

`\append@label@year@`

```

2258 \def\append@label@year@{%
2259     \safe@set\@tempcnta\bib@year
2260     \edef\bib@citeyear{the\@tempcnta}%
2261     \append@to@stem{%
2262         \ifx\bib@year\@empty
2263         \else
2264             \@xp\year@short \bib@citeyear \@nil
2265         \fi
2266     }%
2267 }

2268 \let\alpha@label@suffix\@empty
2269
2270 \newcount\alpha@suffix
2271 \alpha@suffix\@ne
2272 \let\@suffix@format\@alph
```

\calc@alpha@suffix

```

2273 \def\calc@alpha@suffix{%
2274     \@tempwafalse
2275     \compare@stems\previous@stem\current@stem
2276     \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.

```

2277     \ifx\previous@year\current@year
2278         \@tempwattrue
2279     \fi
2280 \else
2281     \begingroup
2282         \let\name\@firstofone
2283         \@apply\auto@stringify\amsrefs@textsymbols
2284         \@apply\auto@stringify\amsrefs@textaccents
2285         \ifundefined{amsrefs@stem\current@stem}{%
2286             \expandafter\gdef\csname amsrefs@stem\current@stem\endcsname{}}%
2287         }{%
2288             \DuplicateBibLabelWarning
2289         }%
2290     \endgroup
2291 \fi
2292 \if@tempswa
2293     \global\advance\alpha@suffix\@ne
2294     \edef\alpha@label@suffix{\@suffix@format\alpha@suffix}%
2295     \ifnum\alpha@suffix=\tw@
2296         \immediate\write\@auxout{%
2297             \string\ModifyBibLabel{\prev@citekey}%
2298         }%
2299     \fi
2300 \else
2301     \let\alpha@label@suffix\@empty
2302     \global\alpha@suffix\@ne
2303     \@xp\ifx \csname b@\current@citekey @suffix\endcsname \relax
2304     \else
2305         \edef\alpha@label@suffix{\@suffix@format\alpha@suffix}%
2306     \fi
2307 \fi
2308 }

```

\ifsame@stems

```

2309 \newif\ifsame@stems

```

\compare@stems

```

2310 \def\compare@stems#1#2{%
2311     \begingroup
2312         \purge@edef\@tempa{#1}%
2313         \purge@edef\@tempb{#2}%
2314         \lc@edef\@tempa{\@tempa}%
2315         \lc@edef\@tempb{\@tempb}%
2316         \ifx\@tempa\@tempb
2317             \def\@tempa{\same@stemstrue}%
2318         \else
2319             \def\@tempa{\same@stemsfalse}%
2320         \fi
2321     \@xp\endgroup
2322     \@tempa

```

```

2323 }

\ModifyBibLabel
2324 \def\ModifyBibLabel#1{%
2325   \global\@xp\let\csname b@#1@suffix\endcsname\@empty
2326 }

```

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 92 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_1, \dots, i_3 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_1 is inserted between them; otherwise, i_2 is inserted between each pair of items except for the last pair, where i_3 is inserted. Thus,

n	output
1	$S T_1 E$
2	$S T_1 i_1 T_2 E$
3	$S T_1 i_2 T_2 i_3 T_3 E$
4	$S T_1 i_2 T_2 i_2 T_3 i_3 T_4 E$

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_n is a macro taking one argument,
- p_n is punctuation—material that must precede a line break if one occurs at this boundary,

- i_n 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:

```
\PrintSeries{\m} {f_0} {p_1}{i_1f_1} {p_2}{i_2f_2} {p_3}{i_3f_3}
      {S} {\m{T_1}...\m{T_n}} {E}
```

and our comma-separated list example becomes

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

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

```
2327 \newcount\sseries@index
```

\PrintSeries

```
2328 \def\PrintSeries#1#2#3#4#5#6#7#8{%
2329   \begingroup
2330     \def\sseries@add@a{#2}%
2331     \def\sseries@add@b{\SwapBreak{#3}#4}%
2332     \def\sseries@add@c{\SwapBreak{#5}#6}%
2333     \def\sseries@add@d{\SwapBreak{#7}#8}%
2334     \def\sseries@add@e{\SwapBreak{#7}}%
2335     \PrintSeries@a{#1}%
2336 }
```

\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**.

```
2337 \def\PrintSeries@a#1#2#3#4{%
2338   \get@numberof\@tempcnta#1{#3}%
2339   \chardef\sseries@total=\@tempcnta
2340   \ifnum\sseries@total=\@ne
2341     \let\SingularPlural\@firstoftwo
2342   \else
2343     \let\SingularPlural\@secondoftwo
2344   \fi
2345   \series@index=\z@
2346   \let#1\sseries@add
2347   #2#3#4\relax
2348   \endgroup
2349 }
```

\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**.

```
2350 \def\sseries@add#1{%
2351   \advance\sseries@index\@ne
2352   \ifx\etaltext#1\relax
2353     \ifnum\sseries@index=\tw@
2354       \def\@tempa{\space\SubEtal}%
2355     \else
2356       \def\@tempa{\sseries@add@e\space\SubEtal}%
2357     \fi
```

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

```

2358      \series@index\@MM
2359    \else
2360      \ifcase\series@index
2361      \or
Material before name 1:
2362      \let\@tempa\series@add@a
2363    \or
Material before name 2:
2364      \ifnum\series@total<\thr@@
2365      \let\@tempa\series@add@b
2366    \else
2367      \let\@tempa\series@add@c
2368    \fi
2369  \else
Material before names 3, 4, 5,...
2370      \ifnum\series@index=\series@total
2371      \let\@tempa\series@add@d
2372    \else
2373      \ifnum\series@index<\series@total
2374      \let\@tempa\series@add@c
2375    \else
2376      \let\@tempa\@gobble
2377    \fi
2378  \fi
2379 \fi
2380 \fi
2381 \@tempa{#1}%
2382 }
```

\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.

```

2383 \def\SwapBreak#1{%
2384   \relax\ifvmode\leavevmode\fi
2385   \@tempcnta\@MM
2386   \toks@{#1}%

```

First, remove any preceding glue. (There usually shouldn't be any of this.)

```

2387   \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.

```
2388 \ifnum \lastkern>\@ne \unkern \fi
```

And now look for a penalty and stash it in a safe place.

```
2389 \ifnum\lastpenalty=\z@
2390 \else
2391 \tempcnta\lastpenalty
2392 \unpenalty
2393 \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 1sp, 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`.

```
2394 \edef\@tempa{%
2395 \@nx\deferredquoteslogical
2396 \ifnum\lastkern=\@ne
2397 \else
2398 \ifnum\spacefactor=\sfcode\@xp\@xp\@xp\@xp\@car\string#1)\@nil
2399 \else
2400 \ifnum\spacefactor=\@nopunctsfcode
2401 \else
2402 \the\toks@
2403 \fi
2404 \fi
2405 \fi
2406 \@nx\deferredquotes
2407 \ifnum\@tempcnta=\@MM \else \penalty\number\@tempcnta\space \fi
2408 \ifnum\lastkern=\@ne \ignorespaces \fi
2409 }%
2410 \@tempa
2411 }
```

`\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.

```
2412 \newcommand{\SingularPlural}[2]{#1}
2413 \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.)

```
2414 \BibSpec{nameLE}{
2415 +{ }{ }{given}
2416 +{ }{\IfEmptyBibField{given}{ }{ }}{surname}
2417 +{ }{ }{jr}
2418 }
```

`\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.”

```
2419 \BibSpec{nameBE}{
2420     +{}{}{surname}
2421     +{}{ }{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.

```
2422 %     +{}{ }{jr}
2423 }
```

`\setbib@nameinverted` Inverted western-style names: “Doe, John, Jr.”

```
2424 \BibSpec{nameinverted}{
2425     +{} {} {surname}
2426     +{,}{ } {given}
2427     +{,}{ } {jr}
2428 }
```

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.

```
2429 \newcommand{\PrintNames}{%
2430     \@ifstar{\PrintNames@a\set@othername}{\PrintNames@a\set@firstname}%
2431 }
```

`\PrintNames@a`

```
2432 \newcommand{\PrintNames@a}[4]{%
2433     \PrintSeries{name}
2434     {#1}
2435     {}{ and \set@othername}
2436     {,}{ \set@othername}
2437     {,}{ and \set@othername}
2438     {#2}{#4}{#3}%
2439 }
```

`\set@firstname` By default, the first name is formatted in little-endian format. The `author-year` option changes this to inverted order.

```
2440 \def\set@firstname#1{%
2441     \set@name{#1}\setbib@nameLE
2442 }
```

`\set@othername` The rest of the names are set in little-endian format by default.

```
2443 \def\set@othername#1{%
2444     \set@name{#1}\setbib@nameLE
2445 }
```

`\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.

```
2446 \def\set@name#1{%
2447     \name@split#1,,,\@nil
2448     \set@name@a
2449 }
```

`\set@namea` Use the requested format unless the `order` property has been set to “inverted.”

```
2450 \def\set@name@a#1{%
2451     \begingroup
2452         \get@current@properties
2453         \select@auxlanguage
2454         \def\@tempa{yes}%
2455         \ifx\@tempa\prop'inverted
2456             \setbib@nameBE
2457         \else
2458             #1%
2459         \fi
2460     \endgroup
2461 }
```

`\PrintPrimary`

```
2462 \def\PrintPrimary{%
2463     \ifx\current@primary\@empty
2464         \EmptyPrimaryWarning
2465     \else
2466         \print@primary\current@primary
2467     \fi
2468 }
```

`\EmptyPrimaryWarning`

```
2469 \def\EmptyPrimaryWarning{%
2470     \amsrefs@warning{No authors, editors or translators}%
2471 }
```

`\PrintAuthors` The comparison of `\previous@primary` and `\current@primary` doesn't look at auxiliary properties (see also `\PrintEditorsA` and `\PrintTranslatorsA`). This is probably ok.

```
2472 \newcommand{\PrintAuthors}[1]{%
2473     \ifx\previous@primary\current@primary
2474         \sameauthors\@empty
2475     \else
2476         \def\current@bibfield{\bib'author}%
2477         \PrintNames{}-{}{#1}%
2478     \fi
2479 }
```

```

\sameauthors
2480 \newcommand{\sameauthors}[1]{\bysame#1}

\bysame
2481 \def\bysame{%
2482     \leavevmode\hbox to3em{\hrulefill}\thinspace
2483     \kern\z@
2484 }

\PrintNameList This just prints the names without any additional information.
2485 \newcommand{\PrintNameList}{\PrintNames{}}

\PrintEditorsC
2486 \newcommand{\PrintEditorsC}[1]{%
2487     \PrintNames{Edited by }{#1}%
2488 }

\PrintEditorsA When we consider editor names we have to think about some further compli-
cations. 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.
2489 \newcommand{\PrintEditorsA}[1]{%
2490     \ifx\previous@primary\current@primary
2491         \sameauthors{(ed\Plural{s}.)}%
2492     \else
2493         \def\current@bibfield{\bib'editor}%
2494         \PrintNames{ (ed\Plural{s}.)}{#1}%
2495     \fi
2496     \erase@field\bib'editor
2497 }

\PrintEditorsB
2498 \newcommand{\PrintEditorsB}{%
2499     \PrintNames*{(\SwapBreak{,}~ed\Plural{s}.)}%
2500 }

\PrintContributions
2501 \newcommand{\PrintContributions}[1]{%
2502     \PrintSeries
2503     {\fld@elt}
2504     {\print@contribution}
2505     {}{ and \print@contribution}
2506     {,}{ \print@contribution}
2507     {,}{ and \print@contribution}{#1}%
2508 }

\print@contribution
2509 \newcommand{\print@contribution}[1]{%
2510     \in@={#1}%
2511     \ifin@
2512         \ifnum\series@index=\@ne with \fi
2513         \RestrictedSetKeys{bib}{%
2514             \bib@print@inner\setbib@contribution{\the\rsk@toks}%
2515             }{#1}%
2516     \else
2517         #1%
2518     \fi
2519 }

```

\resolve@inner

```

2520 \def\resolve@inner#1#2{%
2521   \in@={#2}%
2522   \ifin@
2523     \RestrictedSetKeys[]{\bib}{#1{\the\rsk@toks}}{#2}%
2524   \else
2525     \@ifundefined{bi@#2}{%
2526       \XRefWarning{#2}%
2527     }{%
2528       #1{\csname bi@#2\endcsname}%
2529     }%
2530   \fi
2531 }
```

\PrintConference

```

2532 \def\PrintConference{%
2533   \resolve@inner{\bib@print@inner\setbib@conference}
2534 }
```

\PrintConferenceDetails

```

2535 \def\PrintConferenceDetails#1{%
2536   \ifx\@empty\bib'address
2537     \ifx\@empty\bib'date
2538     \else
2539       \PrintConferenceDetails@
2540     \fi
2541   \else
2542     \PrintConferenceDetails@
2543   \fi
2544 }
```

\PrintConferenceDetails@

```

2545 \def\PrintConferenceDetails@{%
2546   \ifnum\lastkern=\@ne\else\space\fi(\kern 1sp
2547   \ifx\@empty\bib'address
2548   \else
2549     \bib'address
2550   \fi
2551   \ifx\@empty\bib'date
2552   \else
2553     \SwapBreak{,}\space
2554     \print@date
2555   \fi
2556   )%\spacefactor\sffcode`\,%
2557 }
```

\PrintBook

```

2558 \def\PrintBook{%
2559   \resolve@inner{\bib@print@inner\setbib@innerbook}
2560 }
```

\PrintReprint

```

2561 \def\PrintReprint{%
2562   \resolve@inner{\bib@reprint}
2563 }
```

\bib@reprint

```

2564 \def\bib@reprint#1{%
```

```

2565 \begingroup
2566 #1\relax % execute definitions locally
2567 \bib@resolve@xrefs
2568 \bib@field@patches
2569 \bib'setup
2570 \IfEmptyBibField{copula}{reprinted in}{\bib'copula} \nopunct
2571 \let\bib'language@empty
2572 \setbib@book
2573 \endgroup
2574 }

```

\PrintTranslation

```

2575 \def\PrintTranslation{%
2576 \resolve@inner{\bib@translation}
2577 }

```

\bib@translation

```

2578 \def\bib@translation#1{%
2579 \begingroup
2580 #1\relax % execute definitions locally
2581 \bib@resolve@xrefs
2582 \bib@field@patches
2583 \bib'setup
2584 \let\PrintPrimary@empty
2585 \bib@append{;}{ % keep this space!
2586 \IfEmptyBibField{language}{English}{\bib'language} transl.%
2587 \IfEmptyBibField{pages}{ in \kern\@one sp}{, }%
2588 } \bib'transition
2589 \let\bib'language@empty
2590 \setbib@@
2591 \endgroup
2592 }

```

\PrintTranslatorsC

```

2593 \newcommand{\PrintTranslatorsC}[1]{%
2594 \PrintNames{translated by }{ }{#1}%
2595 }

```

\PrintTranslatorsA

```

2596 \newcommand{\PrintTranslatorsA}[1]{%
2597 \ifx\previous@primary\current@primary
2598 \sameauthors{(trans.)}%
2599 \else
2600 \def\current@bibfield{\bib'translator}%
2601 \PrintNames{}{(trans.)}{#1}%
2602 \fi
2603 \erase@field\bib'translator
2604 }

```

\PrintTranslatorsB

```

2605 \newcommand{\PrintTranslatorsB}[1]{
2606 \PrintNames*{ }{\SwapBreak{,}~tran\Plural{s}.)}%
2607 }

```

Some special handling for “et alii” or “and others”.

```

2608 \DefineName{alii}{\etaltext}
2609 \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’.)

```
2610 \newcommand{\etaltext}{et al.}
2611 \newcommand{\SubEtal}[1]{\etaltext}
```

6.23 The partial field

`\print@partial`

```
2612 \newcommand{\print@partial}{%
2613     \resolve@inner{\bib@print@inner\setbib@partial}
2614 }
```

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.

```
2615 \newcommand{\parenthesize}[1]{%
2616     \leavevmode\push@bracket\upn{(#1\pop@bracket
2617 }
```

`\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.)

```
2618 \providecommand{\upn}[1]{#1}
```

`\push@bracket`

```
\pop@bracket
2619 \let\bracket@stack\@empty
2620
2621 \def\push@bracket#1{%
2622     \xdef\bracket@stack{#1\bracket@stack}%
2623 }
2624
2625 \def\pop@bracket{%
2626     \iffalse\fi
2627     \@xp\pop@bracket@a\bracket@stack \@empty}%
2628 }
2629
2630 \def\pop@bracket@a#1{%
2631     \leavevmode\popn{#1}%
2632     \xdef\bracket@stack{\iffalse}\fi
2633 }
```

`\bibquotes`

```
2634 \newcommand{\bibquotes}[1]{%
2635     \textquotedblleft#1%
2636     \gdef\deferredquotes{%
2637         \global\let\deferredquotes\@empty
2638         \textquotedblright
2639     }%
2640 }
```

`\mdash` Cf. `textcmds`, where there’s also a penalty added.

```
\ndash
2641 \providecommand{\mdash}{\textemdash}
2642 \providecommand{\ndash}{\textendash}
```

`\strip@MRprefix`

```
2643 \def\strip@MRprefix#1#2#3#4\@nil{%
2644     \def\@tempa{#1#2#3#4}%
2645 }
```

\MR

`\MRhref` For older versions of some AMS document classes, this patch is needed.

```
\PrintPartials
```

```

2676 \newcommand{\DashPages}[1]{%
2677     p\pp@scan@a#1@ndash p@ndash{\pp@scan#1-p@-}\@nil}\@nil.~#1%
2678 }
2679
2680 \def\pp@scan#1-#2@-#3#4\@nil{#3}

```



```

2681
2682 \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.

```

2683 \newcommand{\eprintpages}[1]{%
2684   #1\IfEmptyBibField{eprint}{}\IfEmptyBibField{journal}{ pp.}{}}%
2685 }

```

`\PrintThesisType`

```

2686 \def\PrintThesisType#1{%
2687   \thesis@type#1?\@nil{#1}%
2688 }
2689
2690 \def\thesis@type#1#2\@nil#3{%
2691   \ifx p#1%
2692     Ph.D. Thesis%
2693   \else
2694     \ifx m#1%
2695       Master's Thesis%
2696     \else
2697       #3%
2698     \fi
2699   \fi
2700 }

```

`\PrintDOI` Perhaps need to add allowbreak penalties at the parentheses in a DOI. Also what about prohibiting a break after the leading S?

```

2701 \newcommand{\PrintDOI}[1]{%
2702   DOI #1%
2703 }

```

`\PrintDatePV` Print date in different forms depending on DOI and volume information.

```

2704 \newcommand{\PrintDatePV}[1]{%
2705   \IfEmptyBibField{doi}{%
2706     \let\@tempa\PrintDate
2707   }{%
2708     \IfEmptyBibField{volume}{%
2709       \let\@tempa\PrintDatePosted
2710     }{%
2711       \let\@tempa\PrintDate
2712     }%
2713   }%
2714   \@tempa{#1}%
2715 }

```

`\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.

```

2716 \newcommand{\PrintDate}[1]{(\print@date)}

```

`\PrintDateB` The same, but without the parentheses.

```

2717 \newcommand{\PrintDateB}[1]{\print@date}

```

`\PrintDateField`

```

2718 \newcommand{\PrintDateField}[1]{%

```

```

2719     \@ifempty{#1}{-}{%
2720         \begingroup
2721             \bib@parsedate{#1}%
2722             \print@full@date
2723         \endgroup
2724     }%
2725 }

\print@full@date
2726 \def\print@full@date{%
2727     \ifx\bib@month\@empty
2728     \else
2729         \print@month@day
2730     \fi
2731     \bib@year
2732 }

\print@date
2733 \let\print@date\print@full@date

\print@month@day
2734 \def\print@month@day{%
2735     \bib@monthname
2736     \ifx\@empty\bib@day \else \nobreakspace\number 0\bib@day,\fi
2737     \space
2738 }

\bib@monthname With the Babel package, month names for a given language are typically avail-
able 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.
2739 \newcommand{\bib@monthname}{%
2740     \ifcase 0\bib@month
2741     \or January\or February\or March\or April\or May\or June\or
2742     July\or August\or September\or October\or November\or December\or
2743     Winter\or Spring\or Summer\or Fall\else Unknown Month%
2744     \fi
2745 }

\PrintYear You can use \PrintYear if you want to suppress month/day even when supplied
in the data.
2746 \newcommand{\PrintYear}[1]{\bib@year}

\PrintDatePosted This one is special for AMS use.
2747 \newcommand{\PrintDatePosted}[1]{\unskip, posted on \print@date}

\PrintEdition
2748 \newcommand{\PrintEdition}[1]{%
2749     \afterassignment\print@edition
2750     \count@ 0#1\relax\@nil
2751 }

\print@edition If the number assignment swept up all the contents, produce a cardinal number
from \count@.
2752 \def\print@edition#1#2\@nil{%

```

```

2753 \ifx\relax#1\relax
2754 \ifnum\count@>\z@
2755 \CardinalNumeric\count@
2756 \else
2757 ??th%
2758 \fi
2759 \ \editiontext
2760 \else
2761 \ifnum \count@>\z@ \number\count@ \fi
2762 #1#2\relax
2763 \fi
2764 }

```

\editiontext

```
2765 \newcommand{\editiontext}{ed.}
```

\CardinalNumber

```

2766 \newcommand{\CardinalNumeric}[1]{%
2767 \number#1\relax
2768 \if
2769 \ifnum#1<14
2770 \ifnum#1>\thr@@ T\else F\fi
2771 \else
2772 F%
2773 \fi
2774 T%
2775 th%
2776 \else
2777 \@xp\keep@last@digit\@xp#1\number#1\relax
2778 \ifcase#1th\or st\or nd\or rd\else th\fi
2779 \fi
2780 }

```

\keep@last@digit

```

2781 \def\keep@last@digit#1#2{%
2782 \ifx\relax#2%
2783 \@xp\@gobbletwo
2784 \else
2785 #1=#2\relax
2786 \fi
2787 \keep@last@digit#1%
2788 }

```

\SentenceSpace Note how careful we are here to preserve \frenchspacing.

```
2789 \newcommand{\SentenceSpace}{\relax\ifhmode\spacefactor\`. \fi}
```

\eprint For now, this does nothing. Could do a url/hyperlink or something.

```
2790 \newcommand{\eprint}[1]{\url{#1}}
```

The 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_T_EX support

`\bibliographystyle` Disable `\bibliographystyle` since we're going to handle that behind the scenes.

```
2791 \let\bibliographystyle\@gobble
```

`\bibtex@style`

```
2792 \def\bibtex@style{amsrn}
```

```
2793 \AtBeginDocument{
```

```
2794   \if@filesw
```

```
2795     \immediate\write\@auxout{\string\bibstyle{\bibtex@style}}%
```

```
2796   \fi
```

```
2797 }
```

6.26 Implementing package options

6.26.1 The alphabetic option

```
2798 \IfOption{alphabetic}{%
```

```
2799   \def\bibtex@style{amsra}%
```

```
2800   \let\process@citelist\process@citelist@unsorted
```

```
2801   \amsrefs@option@alphabetic
```

```
2802 }{}
```

6.26.2 The shortalphabetic option

```
2803 \IfOption{shortalphabetic}{%
```

```
2804   \def\bibtex@style{amsrs}%
```

```
2805   \let\process@citelist\process@citelist@unsorted
```

```
2806   \amsrefs@option@shortalphabetic
```

```
2807 }{}
```

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`.

```
2808 \AtBeginDocument{%
```

```
2809   \@ifpackageloaded{backref}{%
```

```
2810     \let\PrintBackRefs\print@backrefs
```

```
2811     \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.)

```
2812   \begingroup
```

```
2813     \makeatletter
```

```
2814     \@input{\jobname.brf}%
```

```
2815     \if@filesw
```

```
2816       \newwrite\tf@brf
```

```
2817       \immediate\openout\tf@brf \jobname.brf\relax
```

```
2818     \fi
```

```
2819   \endgroup
```

```
2820 }{}%
```

```
2821 }
```

6.26.4 The citation-order option

```
2822 \IfOption{citation-order}{%
```

```
2823   \IfOption{alphabetic}{%
```

```
2824     \amsrefs@warning@nl{%
```

```

2825             The citation-order and alphabetic options are
2826             incompatible%
2827         }%
2828     }{
2829         \def\bibtex@style{amsru}%
2830     }
2831 }{}

```

6.26.5 The initials option

```

2832 \IfOption{initials}{% TRUE:
2833     \BibSpec{nameLE}{
2834         +{}{}{initials}
2835         +{}{\IfEmptyBibField{initials}{}{ }}{surname}
2836         +{}{ }{jr}
2837     }
2838
2839     \BibSpec{nameBE}{
2840         +{}{}{surname}
2841         +{}{ }{initials}
2842         %   +{}{ }{jr}
2843     }
2844
2845     \BibSpec{nameinverted}{
2846         +{} {} {surname}
2847         +{,}{ } {initials}
2848         +{,}{ } {jr}
2849     }
2850 }{% initials? FALSE:
2851 %     \let\extract@initials@gobble
2852 } % end conditional code for initials option

```

6.26.6 The logical-quotes option

\deferredquotes

```

2853 \let\deferredquotes\@empty

```

\deferredquoteslogical

```

2854 \IfOption{logical-quotes}{%
2855     \def\deferredquoteslogical{\deferredquotes}%
2856 }{%
2857     \let\deferredquoteslogical\relax
2858 }

```

6.26.7 The non-compressed-cites option

```

2859 \IfOption{non-compressed-cites}{%
2860     \let\cite@compress\cite@print
2861 }{}

```

6.26.8 The non-sorted-cites option

```

2862 \IfOption{non-sorted-cites}{%
2863     \let\process@citelist\process@citelist@unsorted
2864 }{}

```

6.26.9 The short-journals option

```

2865 \IfOption{short-journals}{%
2866     \renewcommand{\DefineJournal}[4]{%
2867         \bib*{#1}{periodical}{
2868             issn={#2},
2869             journal={#3},
2870         }%
2871     }

```

2872 }{}

6.26.10 The short-publishers option

```
2873 \IfOption{short-publishers}{%
2874   \renewcommand{\DefinePublisher}[4]{%
2875     \bib*{#1}{publisher}{%
2876       publisher={#2},%
```

Maybe `short-publishers` should suppress the `address`? Or is that a separate option? I sense a combinatorial explosion coming on....

```
2877       address={#4},
2878     }%
2879   }%
2880 }{}
```

6.26.11 The short-months option

```
2881 \IfOption{short-months}{%
2882   \renewcommand{\bib@monthname}{%
2883     \ifcase 0\bib@month
2884     \or Jan.\or Feb.\or Mar.\or Apr.\or May\or June\or
2885     July\or Aug.\or Sep.\or Oct.\or Nov.\or Dec.\or
2886     Winter\or Spring\or Summer\or Fall\else Unknown Month%
2887   \fi
2888   }%
2889 }{}
```

6.26.12 The y2k option

```
2890 \IfOption{y2k}{%
2891   \IfOption{alphabetic}{%
2892     \def\year@short#1\@nil{#1}%
2893     \def\bibtex@style{amsry}%
2894   }{%
2895     \amsrefs@warning@nl{%
2896       The y2k option can only be used with the~J%
2897       alphabetic option%
2898     }%
2899   }
2900 }{}
```

6.26.13 The bibtex-style option

```
2901 \IfOption{bibtex-style}{%
2902   \RequirePackage{amsbst}
2903 }{}
```

6.26.14 The msc-links option

```
2904 \IfOption{msc-links}{%
```

Unless you're using pdf \TeX , 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.

```
2905   \@ifundefined{href}{%
2906     \def\parse@MR#1 (#2)#3\@nil{%
2907       \def\MR@url{http://www.ams.org/mathscinet-getitem?mr=#1}%
2908       \def\@tempd{#1}%
2909       \def\@tempe{#2}%
2910     }%
2911     \def\MRhref#1#2{%
2912       \begingroup
2913         \parse@MR#1 ()\@empty\@nil%
2914         \href{\MR@url}{\@tempd\vphantom{()}}%
2915         \ifx\@tempe\@empty
2916           \else
```

```

2917          \ \href{\MR@url}{(\@tempe)}%
2918          \fi
2919      \endgroup
2920  }%
2921  }{}
2922 }{}

```

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.

```

2923 \IfOption{author-year}{\PopCatcodes \endinput}

```

`\@biblistsetup`

```

2924 \renewcommand{\@biblistsetup}[1]{%
2925   \RestrictedSetKeys{\biblist}{\the\rsk@toks}{#1}%
2926   \rkvIfEmpty{biblist}{prefix}{}{%
2927     \amsrefs@warning{%
2928       The 'prefix' biblist option cannot be used\MessageBreak
2929       with the author-year option.%
2930     }%
2931   }%
2932   \rkvIfEmpty{biblist}{labels}{}{%
2933     \amsrefs@warning{%
2934       The 'labels' biblist option cannot be used\MessageBreak
2935       with the author-year option.%
2936     }%
2937   }%
2938 }

```

`\generate@label`

```

2939 \def\generate@label{%

```

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

```

2940   \ifx\bib'label\@empty
2941     \begingroup

```

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

```

2942       \global\let\previous@stem\current@stem
2943       \global\let\current@stem\@empty
2944       \global\let\previous@year\current@year
2945       \global\let\current@year\bib@year

```

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

$$\text{\name{Last}_1, \text{\name{First}_1} } \text{\name{Last}_2, \text{\name{First}_2} } \dots \text{\name{Last}_n, \text{\name{First}_n} }$$

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.

```

2946       \@apply\auto@protect\amsrefs@textsymbols
2947       \@apply\auto@protect\amsrefs@textaccents
2948       \def\name##1{\@nx\name{\lncan@a##1,\@nil}}%
2949       \auto@protect\etaltext
2950       \normalize@edef\current@stem{\current@primary}%
2951       \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.

```

2952       \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.)

```

2953         \edef\@tempa{%
2954             \def\@nx\cite@label{\current@stem}%
2955             \def\@nx\bib@label@year{%
2956                 \current@year
2957                 \alpha@label@suffix
2958             }%
2959         }
2960     \xp@endgroup
2961     \@tempa
2962 \fi
2963 }
```

`\lnscan@a`

```
2964 \def\lnscan@a#1,#2\@nil{#1}
```

`\citesel@author`

```
2965 \def\citesel@author#1#2#3#4#5{\PrintCiteNames{#3}}
```

`\citesel@authoryear`

```
2966 \def\citesel@authoryear#1#2#3#4#5{\PrintCNY{#3}{#4}}
```

`\citesel@object`

```
2967 \def\citesel@object#1#2#3#4#5{\PrintCiteNames{#3} \citeleft#4}
```

`\citesel`

```
2968 \let\citesel\citesel@authoryear
```

`\citeleft`

```
2969 \def\citeleft{()%
```

`\citeright`

```
2970 \def\citeright{)%}
```

`\@citeleft`

```
2971 \def\@citeleft{\ifx\citesel\citesel@object\else\citeleft\fi}%
```

`\citepunct`

```
2972 \def\citepunct{; }
```

`\BibLabel`

```

2973 \def\BibLabel{%
2974     \Hy@raisedlink{\hyper@anchorstart{cite.\CurrentBib}\relax\hyper@anchorend}%
2975 }
```

`\process@citelist`

```
2976 \let\process@citelist\process@citelist@unsorted
```

`\ycite`

```

2977 \DeclareRobustCommand{\ycite}[1]{%
2978     \star@\{ \cite@a\citesel@year{#1} \}%
2979 }
```


`\ycites`

```
2980 \DeclareRobustCommand{\ycites}[1]{%
2981   \begingroup
2982     \def\citepunct{, }%
2983     \let\citesel\citesel@year
2984     \cites{#1}%
2985   \endgroup
2986 }
```

`\citeyear`

```
2987 \DeclareRobustCommand{\citeyear}[1]{%
2988   \begingroup
2989     \let\citeleft\@empty
2990     \let\citeright\@empty
2991     \star@{\cite@a\citesel@year{#1}}{}%
2992   \endgroup
2993 }
```

`\ocite`

```
2994 \DeclareRobustCommand{\ocite}[1]{%
2995   \star@{\cite@a\citesel@object{#1}}{}%
2996 }
```

`\ocites`

```
2997 \DeclareRobustCommand{\ocites}[1]{%
2998   \begingroup
2999     \let\@citelist\@ocitelist
3000     \cites{#1}%
3001   \endgroup
3002 }
```

`\ocitelist`

```
3003 \def\@ocitelist#1{%
3004   \PrintSeries{\InnerCite}%
3005   {\ocite}%
3006   {}{ 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.

```
3007   {,}{ \ocite}%
3008   {,}{ and \ocite}%
3009   {}%
3010   {#1}%
3011   {}%
3012 }
```

`\citeauthor`

```
3013 \DeclareRobustCommand{\citeauthor}[1]{%
3014   \star@{\cite@a\citesel@author{#1}}{}%
3015 }
```

`\citeauthorhy`

```
3016 \DeclareRobustCommand{\citeauthorhy}[1]{%
3017   \citeauthor{#1} \ycite{#1}%
3018 }
```

`\fullcite`

```

3019 \DeclareRobustCommand{\fullcite}[1]{%
3020   \begingroup
3021     \let\print@citenames\CiteNamesFull
3022     \star@{\cite@a\citesel@authoryear{#1}}{ }%
3023   \endgroup
3024 }
```

`\fullocite`

```

3025 \DeclareRobustCommand{\fullocite}[1]{%
3026   \begingroup
3027     \let\print@citenames\CiteNamesFull
3028     \star@{\cite@a\citesel@object{#1}}{ }%
3029   \endgroup
3030 }
```

Invert the first author's name.

```

3031 \def\set@firstname#1{%
3032   \set@name{#1}\setbib@nameinverted
3033 }
```

`\PrintCNY`

```

3034 \def\PrintCNY#1#2{%
3035   \PrintCiteNames{#1}%
3036   \@ifnotempty{#2}{\@addpunct{,} #2}%
3037 }
```

`\PrintCiteNames`

```

3038 \def\PrintCiteNames#1{%
3039   \leavevmode
3040   \def\@tempa{#1}%
3041   \ifx\@tempa\prev@names
3042   \else
3043     \gdef\prev@names{#1}%
3044     \@xp\ifx\@car#1.\@nil\CitePrintUndefined
3045     #1\relax
3046   \else
3047     \print@citenames{#1}%
3048   \fi
3049   \fi
3050 }
```

`\CiteNames`

```

3051 \newcommand{\CiteNames}[1]{%
3052   \PrintSeries{name}%
3053   {}%
3054   {}{ 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.

```

3055   {}{\@gobble}%
3056   {}{ \etaltext\@gobble}%
3057   {}%
3058   {#1}%
3059   {}%
3060 }
```

\print@citenames

3061 \let\print@citenames\CiteNames

\CiteNamesFull

3062 \newcommand{\CiteNamesFull}[1]{%

3063 \PrintSeries{name}%

3064 }%

3065 }{ 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.

3066 {,}{ }%

3067 {,}{ and }%

3068 }%

3069 {#1}%

3070 }%

3071 }

\PrintDate No parentheses around the year.

3072 \renewcommand{\PrintDate}[1]{\bib@label@year}

\print@date Only print the year, not the month or day.

3073 \def\print@date{%

3074 \IfEmptyBibField{date}{%

3075 \IfEmptyBibField{year}{\BibField{status}}{\bib@year}%

3076 }%

3077 \bib@year

3078 }%

3079 }

3080 \BibSpec{article}{%

3081 +{ } { \PrintAuthors} {author}

3082 +{.} { \PrintDate} {date}

3083 +{.} { \textit} {title}

3084 +{.} { } {part}

3085 +{:} { \textit} {subtitle}

3086 +{,} { \PrintContributions} {contribution}

3087 +{.} { \PrintPartials} {partial}

3088 +{,} { } {journal}

3089 +{ } { \textbf} {volume}

3090 +{,} { \issuetext} {number}

3091 +{,} { \eprintpages} {pages}

3092 +{,} { } {status}

3093 +{,} { \PrintDOI} {doi}

3094 +{,} { available at \eprint} {eprint}

3095 +{ } { \parenthesize} {language}

3096 +{ } { \PrintTranslation} {translation}

3097 +{;} { \PrintReprint} {reprint}

3098 +{.} { } {note}

3099 +{.} { } {transition}

3100 +{ } { \SentenceSpace \PrintReviews} {review}

3101 }

3102

3103 \BibSpec{book}{%

3104 +{ } { \PrintPrimary} {transition}

3105 +{.} { \PrintDate} {date}

3106 +{.} { \textit} {title}

3107 +{.} { } {part}

```

3108      +{:} { \textit}                {subtitle}
3109      +{,} { \PrintEdition}            {edition}
3110      +{} { \PrintEditorsB}            {editor}
3111      +{,} { \PrintTranslatorsC}       {translator}
3112      +{,} { \PrintContributions}      {contribution}
3113      +{,} { }                          {series}
3114      +{,} { \voltext}                 {volume}
3115      +{,} { }                          {publisher}
3116      +{,} { }                          {organization}
3117      +{,} { }                          {address}
3118      +{,} { }                          {status}
3119      +{} { \parenthesize}              {language}
3120      +{} { \PrintTranslation}          {translation}
3121      +{;} { \PrintReprint}             {reprint}
3122      +{.} { }                          {note}
3123      +{.} {}                           {transition}
3124      +{} { \SentenceSpace \PrintReviews} {review}
3125 }
3126
3127 \BibSpec{collection.article}{%
3128      +{} { \PrintAuthors}              {author}
3129      +{.} { \PrintDate}                 {date}
3130      +{.} { \textit}                    {title}
3131      +{.} { }                           {part}
3132      +{:} { \textit}                    {subtitle}
3133      +{,} { \PrintContributions}        {contribution}
3134      +{,} { \PrintConference}           {conference}
3135      +{} { \PrintBook}                  {book}
3136      +{,} { }                           {booktitle}
3137      +{,} { pp.~}                       {pages}
3138      +{,} { }                           {status}
3139      +{,} { \PrintDOI}                  {doi}
3140      +{,} { available at \eprint}        {eprint}
3141      +{} { \parenthesize}              {language}
3142      +{} { \PrintTranslation}          {translation}
3143      +{;} { \PrintReprint}             {reprint}
3144      +{.} { }                          {note}
3145      +{.} {}                           {transition}
3146      +{} { \SentenceSpace \PrintReviews} {review}
3147 }
3148
3149 \BibSpec{report}{%
3150      +{} { \PrintPrimary}               {transition}
3151      +{.} { \PrintDate}                 {date}
3152      +{.} { \textit}                    {title}
3153      +{.} { }                           {part}
3154      +{:} { \textit}                    {subtitle}
3155      +{,} { \PrintEdition}              {edition}
3156      +{,} { \PrintContributions}        {contribution}
3157      +{,} { Technical Report }          {number}
3158      +{,} { }                           {series}
3159      +{,} { }                           {organization}
3160      +{,} { }                           {address}
3161      +{,} { \eprint}                    {eprint}
3162      +{,} { }                           {status}
3163      +{} { \parenthesize}              {language}
3164      +{} { \PrintTranslation}          {translation}
3165      +{;} { \PrintReprint}             {reprint}
3166      +{.} { }                          {note}

```

```

3167   +{.} {} {transition}
3168   +{} {\SentenceSpace \PrintReviews} {review}
3169 }
3170
3171 \BibSpec{thesis}{%
3172   +{} {\PrintAuthors} {author}
3173   +{.} { \PrintDate} {date}
3174   +{.} { \textit} {title}
3175   +{:} { \textit} {subtitle}
3176   +{,} { \PrintThesisType} {type}
3177   +{,} { } {organization}
3178   +{,} { } {address}
3179   +{,} { \eprint} {eprint}
3180   +{,} { } {status}
3181   +{} { \parenthesize} {language}
3182   +{} { \PrintTranslation} {translation}
3183   +{;} { \PrintReprint} {reprint}
3184   +{.} { } {note}
3185   +{.} {} {transition}
3186   +{} {\SentenceSpace \PrintReviews} {review}
3187 }
3188
3189 \BibSpec{webpage}{%
3190   +{} {\PrintAuthors} {author}
3191   +{.} { \PrintDate} {date}
3192   +{.} { \emph} {title}
3193   +{:} { \emph} {subtitle}
3194   +{,} { \url} {url}
3195   +{.} { Accessed \PrintDateField} {accessdate}
3196   +{.} { } {note}
3197   +{.} {} {transition}
3198 }
3199 \PopCatcodes
3200 \</pkg>

```

6.27 The amsbst package

```

3201 (*bst)
3202 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3203 \ProvidesPackage{amsbst}[2013/03/07 v2.14]
3204 %\RequirePackage{amsrefs}[2004/03/29]
3205 \BibSpec{article}{%
3206   +{} {\PrintAuthors} {author}
3207   +{.} { } {title}
3208   +{.} { } {part}
3209   +{:} { } {subtitle}
3210   +{.} { \PrintContributions} {contribution}
3211   +{.} { \PrintPartials} {partial}
3212   +{.} { \emph} {journal}
3213   +{} { } {volume}
3214   +{} { \parenthesize} {number}
3215   +{:} {} {pages}
3216   +{,} { \PrintDateB} {date}
3217   +{,} { } {status}
3218   +{.} { \PrintTranslation} {translation}
3219   +{.} { Reprinted in \PrintReprint} {reprint}
3220   +{.} { } {note}
3221   +{.} {} {transition}
3222 }

```

```

3223
3224 \BibSpec{partial}{%
3225     +{} {} {part}
3226     +{:} {} {subtitle}
3227     +{.} {} {\PrintContributions} {contribution}
3228     +{.} {} {\emph} {journal}
3229     +{} {} {volume}
3230     +{} {} {\parenthesize} {number}
3231     +{:} {} {pages}
3232     +{,} {} {\PrintDateB} {date}
3233 }
3234
3235 \BibSpec{book}{%
3236     +{} {} {\PrintPrimary} {transition}
3237     +{.} {} {\emph} {title}
3238     +{:} {} {} {part}
3239     +{:} {} {\emph} {subtitle}
3240     +{.} {} {} {series}
3241     +{,} {} {\voltext} {volume}
3242     +{.} {} {Edited by \PrintNameList} {editor}
3243     +{.} {} {Translated by \PrintNameList} {translator}
3244     +{.} {} {\PrintContributions} {contribution}
3245     +{.} {} {} {publisher}
3246     +{.} {} {} {organization}
3247     +{,} {} {} {address}
3248     +{,} {} {\PrintEdition} {edition}
3249     +{,} {} {\PrintDateB} {date}
3250     +{.} {} {} {note}
3251     +{.} {} {} {transition}
3252     +{.} {} {\PrintTranslation} {translation}
3253     +{.} {} {Reprinted in \PrintReprint} {reprint}
3254     +{.} {} {} {transition}
3255 }
3256
3257 \BibSpec{collection.article}{%
3258     +{} {} {\PrintAuthors} {author}
3259     +{.} {} {} {title}
3260     +{:} {} {} {part}
3261     +{:} {} {} {subtitle}
3262     +{.} {} {\PrintContributions} {contribution}
3263     +{.} {} {\PrintConference} {conference}
3264     +{.} {} {\PrintBook} {book}
3265     +{.} {} {In } {booktitle}
3266     +{,} {} {pages~} {pages}
3267     +{.} {} {\PrintDateB} {date}
3268     +{.} {} {\PrintTranslation} {translation}
3269     +{.} {} {Reprinted in \PrintReprint} {reprint}
3270     +{.} {} {} {note}
3271     +{.} {} {} {transition}
3272 }
3273
3274 \BibSpec{conference}{%
3275     +{} {} {} {title}
3276     +{} {} {\PrintConferenceDetails} {transition}
3277 }
3278
3279 \BibSpec{innerbook}{%
3280     +{.} {} {\emph} {title}
3281     +{.} {} {} {part}

```

```

3282 +{:} { \emph} {subtitle}
3283 +{.} { } {series}
3284 +{,} { \voltext} {volume}
3285 +{.} { Edited by \PrintNameList} {editor}
3286 +{.} { Translated by \PrintNameList} {translator}
3287 +{.} { \PrintContributions} {contribution}
3288 +{.} { } {publisher}
3289 +{.} { } {organization}
3290 +{,} { } {address}
3291 +{,} { \PrintEdition} {edition}
3292 +{,} { \PrintDateB} {date}
3293 +{.} { } {note}
3294 +{.} {} {transition}
3295 }
3296
3297 \BibSpec{report}{%
3298 +{} {\PrintPrimary} {transition}
3299 +{.} { \emph} {title}
3300 +{.} { } {part}
3301 +{:} { \emph} {subtitle}
3302 +{.} { \PrintContributions} {contribution}
3303 +{.} { Technical Report } {number}
3304 +{,} { } {series}
3305 +{.} { } {organization}
3306 +{,} { } {address}
3307 +{,} { \PrintDateB} {date}
3308 +{.} { \PrintTranslation} {translation}
3309 +{.} { Reprinted in \PrintReprint} {reprint}
3310 +{.} { } {note}
3311 +{.} {} {transition}
3312 }
3313
3314 \BibSpec{thesis}{%
3315 +{} {\PrintAuthors} {author}
3316 +{,} { \emph} {title}
3317 +{:} { \emph} {subtitle}
3318 +{.} { \PrintThesisType} {type}
3319 +{.} { } {organization}
3320 +{,} { } {address}
3321 +{,} { \PrintDateB} {date}
3322 +{.} { \PrintTranslation} {translation}
3323 +{.} { Reprinted in \PrintReprint} {reprint}
3324 +{.} { } {note}
3325 +{.} {} {transition}
3326 }
3327
3328 \BibSpec{webpage}{%
3329 +{} {\PrintAuthors} {author}
3330 +{.} { } {title}
3331 +{:} { } {subtitle}
3332 +{.} { \PrintDateB} {date}
3333 +{.} { \url} {url}
3334 +{.} { Accessed \PrintDateField} {accessdate}
3335 +{.} { } {note}
3336 +{.} {} {transition}
3337 }

```

\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.

```
3338 \renewcommand{\PrintEditorsA}[1]{%
3339     \def\current@bibfield{\bib'editor}%
3340     \PrintNames{},{, editor\Plural{s}}{#1}%
3341     \erase@field\bib'editor
3342 }
```

\PrintTranslatorsA

```
3343 \renewcommand{\PrintTranslatorsA}[1]{%
3344     \def\current@bibfield{\bib'translator}%
3345     \PrintNames{},{, translator\Plural{s}}{#1}%
3346     \erase@field\bib'translator
3347 }
```

```
3348 \endinput
```

The usual `\endinput` to ensure that random garbage at the end of the file doesn't get copied by `docstrip`.

```
3349 \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.

Symbols		
\ 59, 1787	\@addpunct .. 6, 14, 14, 108, 1448, 3036
\& 25	\@alph 1085, 2272
\' 40, 59, 1788	\@apply 69
\+ 24	\@arabic ... 1093, 2173
\, 24, 120, 2556	\@bibdef 27, 28, 633, 647, 651, 663, 1261, 1264, 1280
\- 24	\@bibitem 24, 541
\.	.. 24, 115, 1789, 2789	\@biblist 502, 507, 537
.aux	8, 8, 8, 8, 9, 9, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10, 10, 45, 54, 56, 56, 56, 56, 57	\@biblistsetup 557, 559, 2924
.bbl	. 8, 8, 9, 27, 28, 43	\@bibtitlestyle ... 1223, 1226
.bib 8	\@car .. 2209, 2398, 3044
.brf 92	\@carcube 2210
.dtx 8	\@chomp ... 86, 1982, 1994, 2047, 2048
.ltx 27, 28, 42	\@chomper 14, 14, 89, 103
\:	... 24, 118, 685, 701	\@citeleft 1512, 1520, 2971
\; 24, 119	\@citelist 48, 1435, 1436, 2999
\= 24, 1790	\@citestyle 1439, 1456
? option 16	\@citex 49
\@undefined 1485	
\@MM	.. 2358, 2385, 2407	
		\@close@bbl@file .. 1277, 1349, 1358, 1368
		\@concat 60, 2174
		\@count@chars 2156, 2161
		\@currentlabel 1067, 1137
		\@current 136
		\@currname 138
		\@empty 13, 18, 18, 20, 44, 44
		\@endbiblist 504, 509, 572
		\@extract@initial . 72, 2122, 2124, 2136
		\@file@und . 1294, 1297, 1300, 1314
		\@firstofone .. 15, 37, 1043, 1412, 1434, 1477, 1765, 1771, 2282
		\@firstoftwo 48, 1057, 2341
		\@firstone . 2209, 2211
		\@firstthree 2210, 2233

- \@for 34
 - \@gobblefour 753
 - \@gobblethree 1530
 - \@ifclassloaded ...
 - . 1201, 1202, 1203
 - \@ifempty 45,
 - 269, 1153, 1381,
 - 1406, 1492,
 - 1665, 1671,
 - 1687, 1739, 2719
 - \@ifnextchar 18
 - \@ifnotempty 45, 240,
 - 718, 742, 1393,
 - 1514, 2214, 3036
 - \@ifpackageloaded .
 - 1764,
 - 1767, 1773, 2809
 - \@ifpackagewith ...
 - 5, 8, 1774
 - \@ifsame@patterns .
 - 1046, 1052
 - \@ifsamepat 1053, 1055
 - \@input@ 1281
 - \@lappend
 - 62, 1785, 1814,
 - 1815, 1817, 1818
 - \@latex@warning ...
 - 574, 1500
 - \@lbibitem 24, 542
 - \@let@token ... 66, 72
 - \@listctr 533, 547
 - \@m 111, 552, 606
 - \@marknames 2213, 2225
 - \@maxlabelwidth 545,
 - 578, 581, 582, 590
 - \@mkboth 41, 1205
 - \@multiauthorlabel
 - . 2203, 2207, 2243
 - \@nametoken . 2211,
 - 2219, 2231, 2233
 - \@nameuse 62
 - \@nil 13, 37,
 - 63, 64, 69, 69, 70
 - \@nilgobble
 - 65, 83, 1924, 2224
 - \@nmbrlistfalse ...
 - .. 519, 1087, 1095
 - \@nmbrlisttrue 546, 1079
 - \@noitemerr 574
 - \@nopunctsfcodes ...
 - ... 123, 124, 2400
 - \@ocitelist 2999, 3003
 - \@oneauthorlabel ..
 - 2201, 2228
 - \@open@bbl@file 676,
 - 1274, 1348, 1364
 - \@pass@options 135
 - \@prev@cite 1642
 - \@secondoftwo
 - ... 48, 1059, 2343
 - \@sptoken ... 1936,
 - 2011, 2061, 2109
 - \@starttoc .. 92, 92, 92
 - \@suffix@format ...
 - .. 1085, 1093,
 - 2272, 2294, 2305
 - \@tabacckludge . 61,
 - 1844, 1856, 1875
 - \@tempa 65
 - \@tempcnta 13, 55
 - \@tempcntb 55
 - \@tempd 2908, 2914
 - \@tempdima 580, 581, 582
 - \@tempe 2909, 2915, 2917
 - \@tempswafalse
 - .. 636, 1128, 2274
 - \@tempswatruer . 632,
 - 1146, 1149, 2278
 - \@temptokena 65
 - \@temptokenb
 - 842, 848, 881
 - \@testdef 41
 - \@threeauthors
 - 2239, 2253
 - \@uclclist 60, 60
 - \@urlbreak 25, 27
 - \@urlfix 24, 32
 - \@urlsetup 20, 22
 - \@xifempty 45, 47
 - \^ 107, 1791
 - \~ 1792
 - \~ 29, 1793
 - _ 1433, 2759, 2917
- ## A
- \A 604
 - abbrv BIB_T_EX style . 15
 - \add@toks@ 61
 - address field
 - 3, 3, 3, 42, 42, 94
 - \addto@defer@list .
 - 764, 768
 - \addto@hook ... 61, 881
 - \AE 1823
 - \ae 60, 60, 1823
 - \after@deleting@token
 - 41, 1943,
 - 1953, 2021,
 - 2059, 2075, 2118
 - \afterassignment 12,
 - 42, 83, 775, 2749
 - \aftergroup ... 20, 292
 - alpha BIB_T_EX style . 15
 - \alpha@label
 - .. 1068, 1077,
 - 1082, 1090, 1099
 - \alpha@label@
 - . 1069, 1082, 1090
 - \alpha@label@suffi
 - 2192,
 - 2268, 2294,
 - 2301, 2305, 2957
 - \alpha@suffi
 - .. 2270, 2271,
 - 2293, 2294,
 - 2295, 2302, 2305
 - alphabetic option ..
 - ... 15, 74, 77, 92
 - amsart class 92
 - amsbook class 47
 - amsbst package ... 101
 - amsgen package 14
 - amsproc class 47
 - amsrdoc.tex 2
 - AMSREFS package 5, 6, 7
 - amsrefs package ...
 - .. 1, 2, 2, 2, 2,
 - 3, 3, 4, 4, 4, 5,
 - 5, 5, 6, 7, 9, 9,
 - 16, 24, 42, 56,
 - 80, 92, 95, 104, 104
 - \amsrefs@lbibitem 515
 - \amsrefs@bibcite ..
 - . 1710, 1715, 1719
 - \amsrefs@bibitem ..
 - 515, 541
 - \amsrefs@error
 - 106, 259, 309, 656
 - \amsrefs@label@prefix
 - 562,
 - 2172, 2173, 2179
 - \amsrefs@lbibitem .
 - 515, 542
 - \amsrefs@option@alphabetic
 - 1081, 2801
 - \amsrefs@option@numeric
 - 1076
 - \amsrefs@option@shortalphabetic
 - 1089, 2806
 - \amsrefs@textaccents
 - 60, 911, 1127,
 - 1783, 1785,
 - 1843, 1857,
 - 1874, 1975,
 - 2181, 2284, 2947
 - \amsrefs@textsymbols
 - 62,
 - 910, 1126, 1812,
 - 1814, 1815,
 - 1817, 1818,
 - 1842, 1858,
 - 1872, 1974,
 - 2180, 2283, 2946
 - \amsrefs@warning ..
 - 105, 566,

- 954, 955, 1160,
 1174, 1179,
 1285, 1311,
 1389, 1394,
 1399, 1582,
 1968, 2040,
 2470, 2927, 2933
 \amsrefs@warning@nl
 .. 4, 6, 9, 158,
 168, 2824, 2895
 amsxport BibTeX style
 73, 74
 \append@label@year
 . 1086, 1094, 2187
 \append@label@year@
 1086, 2258
 \append@to@stem ...
 .. 2174, 2235,
 2241, 2251, 2261
 \ar@hyperlink
 . 1513, 1759, 1768
 \ar@SK@cite . 1511,
 1762, 1775, 1778
 article entry type 4, 7
 \AtBeginDocument 92,
 651, 652, 1711,
 1763, 2793, 2808
 \AtEndDocument ... 1197
 author field 5, 5
 author-year option .
 48,
 77, 82, 83, 95, 95
 \AuthorList 7
 \auto@protect 12, 12,
 12, 57, 62, 910,
 911, 1108, 1125,
 1126, 1127,
 1842, 1857,
 1865, 1872,
 1971, 1972,
 1974, 1975,
 2180, 2181,
 2182, 2183,
 2946, 2947, 2949
 \auto@stringify ...
 ... 58, 2283, 2284

B
 \b 1794
 \b@ 39, 49, 50,
 50, 53, 56, 57, 57
 \b@* 1572
 \b@aaa 53
 \b@bbb 53
 \b@moo 8, 9, 45
 \b@whatever .. 9, 10, 50
 \b@xyz 10
 \b@zzz 53
 babel package 37
 \back@cite . 1747, 2811
 \BackCite ... 1465,
 1544, 1746, 2811
 backref package ...
 92, 92, 92
 backrefs option
 9, 92, 92, 92
 \bbl@copy 681, 735
 \bbl@copy@a .. 739, 741
 \bbl@out 1346,
 1356, 1357,
 1359, 1367, 1370
 \bbl@write
 678, 682, 745,
 1347, 1359, 1369
 \bblname 1268, 1290, 1357
 \begin 9, 9, 9
 \bgroup
 90, 1893, 1961,
 2024, 2078, 2121
 \bi@ 31
 \bib 3, 3, 3, 5, 7, 7, 7,
 10, 10, 10, 10,
 24, 27, 37, 42,
 42, 44, 44, 44,
 44, 45, 49, 62, 629
 \bib' 34
 \bib'bar 19
 \bib'date 36
 \bib'given 64, 64
 \bib'initials 64
 \bib'jr 64
 \bib'language 36
 \bib'pages 19
 \bib'surname 64
 \bib* 27, 28
 \bib@append
 ... 19, 20, 20,
 20, 247, 260, 2585
 \bib@author 38
 \bib@bar 19
 \bib@cite 38, 787, 1097
 \bib@cite@a 1102, 1105
 \bib@cite@check ...
 1117, 1120
 \bib@citeyear 2260, 2264
 \bib@day 1014,
 1021, 1022, 2736
 \bib@dbfile
 .. 1291, 1294,
 1297, 1300,
 1319, 1322, 1325
 \bib@div@mark
 . 1200, 1215, 1228
 \bib@end .. 32, 804, 821
 \bib@exec .. 11, 27,
 31, 652, 660, 665
 \bib@field@patches
 783, 810,
 965, 2568, 2582
 \bib@label 74, 74, 96, 96
 \bib@label@year ...
 .. 1111, 1144,
 1158, 1165,
 1172, 2955, 3072
 \bib@language .. 36,
 1035, 1037,
 1040, 1042,
 1043, 1046,
 1047, 1048, 1062
 \bib@language@fixup
 1010, 1032
 \bib@mklab ... 548, 579
 \bib@month .. 1013,
 1020, 1023,
 2727, 2740, 2883
 \bib@monthname
 . 2735, 2739, 2882
 \bib@parsedate 992,
 995, 1015, 2721
 \bib@parsedate@a ..
 1016, 1018
 \bib@print
 .. 27, 31, 652, 778
 \bib@print@inner ..
 ... 806, 2514,
 2533, 2559, 2613
 \bib@reprint 2562, 2564
 \bib@reset 915
 \bib@resolve@xrefs
 782, 809,
 829, 2567, 2581
 \bib@selectlanguage
 784, 1045
 \bib@start
 .. 74, 96, 779, 817
 \bib@store 27, 665, 774
 \bib@translation ..
 2576, 2578
 \bib@year 38,
 990, 999, 1012,
 1019, 1024,
 2259, 2262,
 2731, 2746,
 2945, 3075, 3077
 \BibAbbrevWarning .
 863, 898, 954
 bibchapter environ-
 ment 1210
 \bibchapter 1235
 \bibcite 9,
 9, 9, 9, 9, 9,
 10, 10, 10, 10,
 41, 56, 56, 56,
 523, 533, 1195,
 1197, 1715, 1719

- \bibcite@a . [1710](#), [1721](#)
- \bibcite@b . [1726](#), [1730](#)
- \bibcite@write [1103](#), [1185](#)
- \bibdata [8](#), [9](#)
- \bibdefer@list [767](#),
[771](#), [1278](#), [1279](#)
- bibdiv environment [1232](#)
- \bibdiv [1238](#)
- \BibField [7](#), [7](#), [185](#), [3075](#)
- \BibItem . [634](#), [637](#), [640](#)
- \bibitem [9](#), [9](#), [9](#), [24](#)
- \BibLabel
[543](#), [614](#), [620](#), [2973](#)
- \biblanguagedefault
. [1035](#), [1062](#)
- \biblanguageEnglish
. [1062](#)
- \bibliography [8](#), [8](#), [9](#)
- \bibliographystyle
. [8](#), [92](#), [2791](#)
- biblist environment
. [1](#), [24](#), [24](#), [500](#)
- \biblist [92](#), [562](#),
[565](#), [566](#), [568](#), [1239](#)
- biblist* environment [506](#)
- \biblistfont . [511](#), [539](#)
- \bibname [41](#), [1198](#), [1210](#)
- \bibquotes [3](#), [6](#), [6](#), [6](#), [2634](#)
- bibsection environ-
ment [1219](#)
- \bibsection [1233](#)
- \BibSelect
. [1262](#), [1265](#), [1268](#)
- \bibselect
[27](#), [30](#), [30](#), [51](#),
[1259](#), [1286](#), [1287](#)
- \bibselect* [28](#), [51](#)
- \bibselect@msg [1271](#)
- \BibSpec [3](#), [4](#), [4](#),
[5](#), [5](#), [5](#), [6](#), [7](#), [7](#),
[7](#), [8](#), [18](#), [18](#), [18](#), [238](#)
- \bbspec@scan
[18](#), [18](#), [18](#), [241](#), [246](#)
- \bbspec@scan@error
. [255](#), [259](#)
- \BibSpecAlias [20](#),
[306](#), [474](#), [475](#),
[476](#), [477](#), [478](#),
[479](#), [480](#), [481](#), [482](#)
- \bibstyle [8](#), [9](#), [2795](#)
- bibtex-style option [94](#)
- \bibtex@style [2792](#),
[2795](#), [2799](#),
[2804](#), [2829](#), [2893](#)
- book class [42](#)
- \bracket@stack [2619](#),
[2622](#), [2627](#), [2632](#)
- bst [3](#)
- \bud [1837](#)
- \bysame [7](#), [7](#), [7](#), [2480](#), [2481](#)
- C**
- \c [1795](#)
- \c@bib [2173](#)
- \c@bib@env [590](#), [595](#), [598](#)
- \calc@alpha@suffix
. [2188](#), [2273](#), [2952](#)
- \calc@author@part .
. [1084](#), [1092](#), [2186](#)
- \calc@author@part@
. [1084](#), [2199](#)
- \calc@author@part@short
. [1092](#), [2206](#)
- \CardinalNumber [2766](#)
- \CardinalNumeric . .
. [8](#), [8](#), [2755](#), [2766](#)
- \catcode [26](#)
- \cdprime [1836](#)
- \ch@ck [1361](#)
- \chapter* [41](#)
- \check@mathfonts [23](#)
- \chomp@ [96](#), [101](#)
- \citation [8](#), [8](#), [9](#), [9](#),
[9](#), [10](#), [10](#), [10](#), [1523](#)
- citation-order op-
tion [74](#), [92](#)
- cite package [47](#)
- \cite [8](#), [8](#), [8](#), [9](#), [9](#), [9](#),
[10](#), [10](#), [10](#), [39](#),
[45](#), [46](#), [46](#), [47](#),
[50](#), [50](#), [50](#), [50](#), [1379](#)
- \cite@a [48](#),
[1463](#), [1464](#),
[1570](#), [2978](#),
[2991](#), [2995](#),
[3014](#), [3022](#), [3028](#)
- \cite@b [48](#), [1474](#), [1481](#)
- \cite@bc [49](#), [1482](#), [1484](#)
- \cite@begingroup . .
. [1441](#), [1457](#),
[1466](#), [1543](#), [1550](#)
- \cite@cb [1510](#),
[1521](#), [1559](#), [1646](#)
- \cite@cj [49](#), [53](#), [55](#), [1507](#)
- \cite@cjs [1594](#), [1655](#)
- \cite@compress
. [1595](#), [1663](#),
[1683](#), [1707](#), [2860](#)
- \cite@compress@a . .
. [1667](#), [1670](#), [1700](#)
- \cite@compress@b . .
. [1676](#), [1686](#)
- \cite@dash [55](#), [1604](#),
[1658](#), [1661](#),
[1691](#), [1695](#), [1703](#)
- \cite@dest [1731](#), [1732](#)
- \cite@dest@b [1736](#), [1738](#)
- \cite@endgroup [1446](#),
[1458](#), [1518](#),
[1545](#), [1551](#), [1565](#)
- \cite@label
. [1067](#), [1111](#),
[1137](#), [1138](#),
[1140](#), [1155](#),
[1158](#), [1165](#), [2954](#)
- \cite@nobib@test . .
. [1487](#), [1491](#)
- \cite@print [1645](#),
[1652](#), [1664](#),
[1675](#), [1678](#),
[1682](#), [1693](#),
[1698](#), [1706](#), [2860](#)
- \cite@single [1380](#), [1404](#)
- \cite@sort [1598](#), [1608](#)
- \cite@sort@a [1612](#), [1615](#)
- \cite@sort@b [1621](#), [1631](#)
- \cite@sorted@e
. [54](#), [1575](#), [1601](#)
- \cite@sorted@s
. [53](#), [1575](#), [1591](#)
- \CiteAltPunct [47](#), [47](#)
- \citeAltPunct [1433](#), [1444](#)
- \citeauthor
. [1539](#), [3013](#), [3017](#)
- \citeauthority [1540](#), [3016](#)
- \citedest [9](#),
[10](#), [10](#), [56](#), [57](#), [1731](#)
- \citeform [1434](#), [1758](#)
- \citeleft [1429](#),
[1440](#), [1448](#),
[1470](#), [1520](#),
[1553](#), [2967](#),
[2969](#), [2971](#), [2989](#)
- \citelist
. [50](#), [1394](#), [1415](#),
[1435](#), [1517](#), [1546](#)
- \citemid
. [47](#), [47](#), [1429](#), [1514](#)
- \citen [1532](#)
- \CiteNames [3051](#), [3061](#)
- \CiteNamesFull
. [3021](#), [3027](#), [3062](#)
- \CitePrintUndefined
. [1129](#), [1496](#),
[1503](#), [1506](#),
[1593](#), [1654](#), [3044](#)
- \citepunct [47](#), [1429](#),
[1440](#), [1444](#),
[1448](#), [2972](#), [2982](#)
- \citeright [1429](#),
[1449](#), [1453](#),
[1471](#), [1515](#),
[1554](#), [2970](#), [2990](#)
- \cites [1390](#), [1531](#),
[1534](#), [1536](#),
[1547](#), [2984](#), [3000](#)

- `\cites@a` 1385,
 1406, 1407, 1531
`\cites@b` ... 1409, 1418
`\cites@c` ... 1422, 1424
`\cites@init` . 1447,
 1459, 1467, 1552
`\cites@o` ... 1382, 1405
`\cites@oo` .. 1405, 1406
`\CITSEL` 48
`\citesel` 39,
 39, 45, 49, 50,
 57, 57, 1377, 2968
`\citesel@author` ...
 . 1469, 2965, 3014
`\citesel@authoryear`
 . 2966, 2968, 3022
`\citesel@number` ...
 . 1375, 1377, 1574
`\citesel@object` ...
 2967,
 2971, 2995, 3028
`\citesel@other`
 1570, 1571
`\citesel@update` ...
 1372, 1526
`\citesel@write`
 1188, 1193
`\citesel@year` 1376,
 2978, 2983, 2991
`\citeyear` 2987
`\CloseBBLFile` 1358, 1366
`\closein` 1319
`\clubpenalty` . 553, 554
`cmr font` 48
`cmsy font` 6
`collection entry type` 7
`\compare@stems`
 2275, 2310
`\copy@bibdef`
 .. 27, 51, 668,
 758, 1261, 1562
`\copy@bibdef@a`
 672, 675, 721
`\count` 13, 1353
`\count@` 90
`\count@chars` 2144, 2153
`\cprime` 1835
`\CPU@normal` 1506, 1654
`\CPU@sort` .. 1590, 1593
`\cs` 61, 62, 62
`\CurLine` 44,
 44, 44, 1325,
 1326, 1332, 1343
`\current@bibfield` .
 20, 268,
 296, 2476, 2493,
 2600, 3339, 3344
`\current@citekey` ..
 815, 819, 826, 2303
`\current@primary` 74,
 83, 95, 956,
 962, 968, 971,
 975, 2184, 2463,
 2466, 2473,
 2490, 2597, 2950
`\current@stem`
 . 74, 95, 1180,
 2169, 2174,
 2178, 2179,
 2191, 2275,
 2285, 2286,
 2942, 2943,
 2950, 2951, 2954
`\current@year`
 77, 2171, 2277,
 2944, 2945, 2956
`\CurrentBib` .. 517,
 530, 613, 616,
 823, 1071, 1098,
 1176, 1195, 2974
`\CurrentBibType` 28, 654
`\cydot` 1838
- D**
- `\d` 1796
`\DashPages` .. 6, 6, 2676
`\dateLANGUAGE` 37
`\Dbar` 1832
`\dbar` 1832
`\DeclareBooleanOption`
 128, 129,
 130, 131, 132,
 133, 143, 144,
 145, 146, 147, 148
`\DeclareExclusiveOptions`
 126,
 127, 140, 141, 142
`\DeclareNameAccent`
 1784,
 1787, 1788,
 1789, 1790,
 1791, 1792,
 1793, 1794,
 1795, 1796,
 1797, 1798,
 1799, 1800,
 1801, 1802,
 1803, 1804,
 1805, 1806,
 1807, 1808,
 1809, 1810, 1811
`\DeclareNameSymbol`
 1813,
 1821, 1822,
 1823, 1824,
 1825, 1826,
 1827, 1828,
 1829, 1830,
 1831, 1832,
 1833, 1834,
 1835, 1836,
 1837, 1838, 1839
`\DeclareOption` ... 134
`\DeclareRobustCommand`
 18, 1435,
 1503, 1531,
 2977, 2980,
 2987, 2994,
 2997, 3013,
 3016, 3019, 3025
`\def` 20, 50, 50
`\defer@bibdef`
 ... 670, 760, 1316
`\deferredquotes` ...
 .. 2406, 2636,
 2637, 2853, 2855
`\deferredquoteslogical`
 2395, 2854
`\DefineAdditiveKey`
 189,
 190, 191, 192,
 193, 194, 195, 196
`\DefineDummyKey` .. 235
`\DefineJournal` 3, 15,
 33, 42, 1247, 2866
`\DefineName`
 . 3, 32, 42, 44,
 1244, 2608, 2609
`\DefinePublisher` ..
 15, 33,
 42, 42, 1253, 2874
`\DefineSimpleKey` ..
 197, 198,
 199, 200, 201,
 202, 203, 204,
 205, 206, 207,
 208, 209, 210,
 211, 212, 213,
 214, 215, 216,
 217, 218, 219,
 220, 221, 222,
 223, 224, 225,
 226, 227, 228,
 229, 230, 231,
 232, 233, 234,
 236, 237, 498, 499
`\DH` 1826
`\dh` 1826
`\DJ` 1827
`\dj` 1827
`\do` 12, 13, 17, 59
`docstrip` 104
`\dospecials` 13
`\DSK@append` 913
`\DSK@def` 912
`\dudot` 1805

- \DuplicateBibKeyWarning . . . [1122](#), [1173](#), [1178](#)
- \DuplicateBibLabelWarning [1178](#), [2288](#)
- E**
- \edef [61](#), [61](#)
- \editiontext [6](#), [6](#), [7](#), [2759](#), [2765](#)
- editor field [19](#)
- \emph [465](#), [466](#), [3192](#), [3193](#), [3212](#), [3228](#), [3237](#), [3239](#), [3280](#), [3282](#), [3299](#), [3301](#), [3316](#), [3317](#)
- \empty@cite [1486](#), [1490](#)
- \EmptyNameWarning [1918](#), [1968](#)
- \EmptyPrimaryWarning [2464](#), [2469](#)
- \endbibchapter . . . [1235](#)
- \endbibdiv [1242](#)
- \endbiblist [1241](#)
- \endbibsection . . . [1233](#)
- \endgroup [20](#)
- \endinput [44](#), [104](#)
- \endlist [576](#)
- environments:
 - bibchapter . . . [1210](#)
 - bibdiv [1232](#)
 - biblist [1](#), [24](#), [24](#), [500](#)
 - biblist* [506](#)
 - bibsection . . . [1219](#)
 - tabbing [62](#)
 - thebibliography [1237](#)
- \eprint [330](#), [392](#), [435](#), [453](#), [2790](#), [3094](#), [3140](#), [3161](#), [3179](#)
- \eprintpages . [327](#), [347](#), [2683](#), [3091](#)
- \erase@field [19](#), [19](#), [20](#), [20](#), [291](#), [2496](#), [2603](#), [3341](#), [3346](#)
- \errmessage [264](#)
- \etalchar [48](#), [60](#), [61](#), [1810](#), [2241](#), [2246](#), [2256](#)
- \etaltext [60](#), [1108](#), [1125](#), [1811](#), [2183](#), [2245](#), [2352](#), [2608](#), [2609](#), [2610](#), [2949](#), [3056](#)
- \ExecuteOptions . . [149](#)
- export-bibtex pack-
age [28](#)
- \extr@cite . . . [1160](#), [1496](#), [1501](#), [1523](#), [1530](#), [1749](#), [1760](#), [1762](#)
- \extract@initial [2096](#), [2101](#), [2118](#), [2133](#)
- \extract@initial@a [69](#), [2103](#), [2105](#)
- \extract@initial@b [2110](#), [2117](#)
- \extract@initial@c [2112](#), [2120](#)
- \extract@initial@d [2127](#), [2129](#)
- \extract@initials [1922](#), [1969](#), [2851](#)
- \extract@surnames [2223](#), [2230](#), [2249](#)
- F**
- \fld@elt [187](#), [192](#), [193](#), [194](#), [195](#), [196](#), [2503](#), [2661](#), [2665](#), [2672](#)
- \font [23](#)
- \fontseries [1504](#)
- \fontshape [1504](#)
- \foo [61](#)
- \format@cite [1758](#), [1760](#), [1768](#)
- \frenchspacing [81](#), [91](#), [114](#), [555](#)
- \fsa@n [69](#)
- \fullcite . . . [1537](#), [3019](#)
- \fullocite . . . [1538](#), [3025](#)
- \futurelet [64](#)
- G**
- \G@refundefinedtrue [1493](#)
- \g@undef . . . [59](#), [725](#), [1278](#)
- \generate@alphalabel [1083](#), [1091](#), [2175](#)
- \generate@label [785](#), [1066](#), [1078](#), [1083](#), [1091](#), [2939](#)
- \get@current@properties [277](#), [294](#), [2452](#)
- \get@namepart [65](#), [65](#), [65](#), [69](#), [1909](#), [1912](#), [1915](#), [1926](#), [2224](#)
- \get@namepart@a [66](#), [69](#), [1929](#), [1931](#), [1943](#)
- \get@namepart@b [66](#), [69](#), [1933](#), [1935](#)
- \get@namepart@c [69](#), [1937](#), [1942](#)
- \get@namepart@d [64](#), [1939](#), [1945](#)
- \get@namepart@e [1947](#), [1949](#)
- \get@namepart@f [66](#), [66](#), [70](#), [1957](#), [1960](#)
- \get@nth@property [296](#)
- \get@numberof . . [73](#), [2185](#), [2231](#), [2338](#)
- \get@property [879](#)
- H**
- \H [1797](#)
- har.bst [8](#)
- \hbox [2482](#)
- \hfil [584](#)
- \hfill [615](#)
- hij.bib [8](#)
- \href [2914](#), [2917](#)
- \hrulefill [2482](#)
- \Hy@backout [1751](#)
- \Hy@raisedlink [616](#), [1771](#), [2974](#)
- \hyper@@link [1760](#)
- \hyper@anchorend [616](#), [1770](#), [2974](#)
- \hyper@anchorstart [616](#), [1769](#), [2974](#)
- hyperref package [56](#), [56](#), [58](#), [58](#)
- \hyph@to@space [2212](#), [2225](#)
- \hyphenpenalty [30](#)
- I**
- \i [61](#), [62](#), [1821](#), [1876](#)
- \if@tempswa . . . [669](#), [679](#), [1152](#), [2292](#)
- \ifBR@verbose [1745](#), [1748](#)
- \IfEmptyBibField [7](#), [7](#), [18](#), [186](#), [2416](#), [2570](#), [2586](#), [2587](#), [2674](#), [2684](#), [2705](#), [2708](#), [2835](#), [3074](#), [3075](#)
- \IfFileExists [11](#), [155](#), [165](#), [1293](#), [1296](#), [1299](#)
- \ifin@ [731](#), [935](#), [1130](#), [1443](#), [2141](#), [2511](#), [2522](#)
- ifoption package . . [14](#)
- \ifsame@stems [2276](#), [2309](#)
- \ifx [12](#)
- \in@ [730](#), [934](#), [1129](#), [1442](#), [2140](#), [2510](#), [2521](#)
- \ini@hyphen [1971](#), [1988](#)

- 1994, 2025,
 2027, 2137, 2138
 initials option . 64, 93
 \InnerCite .. 1420,
 1425, 1450,
 1463, 1555, 3004
 inputenc package .. 61
 \interlinepenalty 552
 inverted attribute .. 65
 isbn field 7
 \issuetext 6, 6, 326,
 346, 2675, 3090
 \itemindent 624
- J**
- \j .. 61, 62, 1822, 1877
 journal field 33
- K**
- \k 1798
 \keep@last@digit ..
 2777, 2781
 \kern 788,
 2483, 2546, 2587
 klm.bib 8
- L**
- \L 1828
 \l 1828
 label field 74, 95
 \labelsep 540, 600, 626
 \labelwidth .. 598,
 599, 620, 621, 622
 \language name 1046, 1062
 \lasp 1833
 \lastkern 270, 2388,
 2396, 2408, 2546
 \lastpenalty 2389, 2391
 \lc@do 1858, 1863
 \lc@edef 1854, 2314, 2315
 \lcode 29
 \lcsymbol 60
 \leftmargin 599, 600,
 621, 623, 624, 626
 \let 20, 20, 50, 50
 \lfhook 1806
 \linebreak 5
 \list 543
 lite option 16
 \lncan@a .. 2948, 2964
 logical-quotes op-
 tion 6, 93
 \lowercase
 . 29, 711, 857,
 894, 1042, 1859
 ltoutenc.dtx 62
- M**
- \m@ne 267, 1056
 \m@th 1456
- \macrotext 52, 55
 \makeatletter 92
 \makelabel 548
 \MakeUppercase ... 1205
 \mathchar 30
 \mathcode 28, 33
 mathscinet package .
 15, 59, 60, 61, 61
 \mathsurround 48
 \mdash 8, 8, 2641
 \mddefault 1504
 \mdseries 1456
 \meaning 20, 52
 \MessageBreak
 ... 107, 1391,
 1400, 2928, 2934
 \MessageBreakNS ...
 ... 107, 1176,
 1180, 1181, 1182
 \modify@xref@fields
 687, 705
 \ModifyBibLabel ...
 2297, 2324
 \month@ 90
 \moo 19, 19, 19
 \MR 2651
 \MR@url 2907, 2914, 2917
 \MRhref 2655, 2659, 2911
 msc-links option ... 94
 \MultipleBibSelectWarning
 1282, 1284
 \MultipleCiteKeyWarning
 1380, 1388
- N**
- name entry type 3
 \name 74,
 74, 74, 74, 95,
 95, 95, 95, 187,
 189, 190, 191,
 844, 860, 864,
 870, 887, 2182,
 2185, 2229,
 2239, 2240,
 2244, 2247,
 2282, 2433,
 2948, 3052, 3063
 \name@split ... 64,
 65, 65, 1908, 2447
 \name@split@finish
 1915, 1917
 \name@split@given .
 ... 65, 1909, 1911
 \name@split@jr
 1912, 1914
 \NameCheck 1920
 \ndash 8, 8, 88,
 2641, 2677, 2682
 \NeedsTeXFormat 2, 3202
- \newcount .. 2270, 2327
 \newcounter .. 497, 586
 \newdimen 578
 \newif 2309
 \newlabel 590
 \newread 1291
 \NG 1829
 \ng 1829
 \NoBibDBFile 1303, 1310
 \nobreak 68
 \nobreakspace 1972, 2736
 nobysame option ... 129
 \nocite 51, 1541
 non-compressed-cites
 option 93
 non-sorted-cites op-
 tion 15, 93
 \NonNumericCiteWarning
 1577, 1581
 \nopunct . 6, 6, 123,
 1440, 1660, 2570
 \normal@bibdef
 27, 27,
 651, 653, 666, 1280
 \NormalCatcodes ... 37
 \normalfont 512
 \normalize@edef ...
 1840,
 1976, 2184, 2950
 \number
 30, 2407, 2736,
 2761, 2767, 2777
- O**
- \O 1825
 \o 1825
 \ObsoleteCiteOptionWarning
 1384, 1398
 \ocite 1532,
 1535, 1538,
 2994, 3005,
 3006, 3007, 3008
 \ocitelist 3003
 \ocites . 48, 1536, 2997
 \OE 60, 1824
 \oe 60, 1824
 \on@line 14
 \OpenBBLFile 1274, 1350
 \openin 1294, 1297, 1300
 \or 2361, 2363,
 2741, 2742,
 2743, 2778,
 2884, 2885, 2886
 organization field ... 3
 \OtherCite . 1555, 1570
 \othercitelist
 1546, 1549
 \othercites 1541, 1542

- `\output@inner@xref@` 694, 729
`\output@xref@` 693, 713, 717, 732
`\output@xref@a` 688, 689, 690, 691, 692, 709
- P**
- `\PackageError` 106
`\PackageInfo` 1749
`\PackageWarning` .. 105
`\PackageWarningNoLine` 4
`\parenthesize` ... 5, 5, 5, 6, 6, 87, 331, 371, 393, 437, 455, 2615, 3095, 3119, 3141, 3163, 3181, 3214, 3230
`\parse@MR` .. 2906, 2913
`partial field` 87
`pcatcode package` .. 16
`place field` 3, 3, 3
`\Plural` ... 7, 7, 79, 81, 2412, 2491, 2494, 2499, 2606, 3340, 3345
`\polhk` 1808
`\pop@bracket` 2616, 2619
`\pop@bracket@a` 2627, 2630
`\PopCatcodes` 2923, 3199
`\pp@scan` ... 2677, 2680
`\pp@scan@a` . 2677, 2682
`\prev@cite` 54, 54, 54, 54, 54, 1605, 1640, 1648, 1675, 1693, 1696, 1697, 1704, 1705
`\prev@cite@cb` 54, 1641, 1692
`\prev@citekey` 816, 826, 2297
`\prev@names` 1460, 3041, 3043
`\previous@primary` . 83, 575, 957, 962, 2473, 2490, 2597
`\previous@stem` 2168, 2178, 2275, 2942
`\previous@year` . 77, 2170, 2277, 2944
`\prime` 6
`\print@backrefs` ... 1753, 2810
`\print@citenames` .. 3021, 3027, 3047, 3061
`\print@contribution` .. 2504, 2505, 2506, 2507, 2509
`\print@date` . 2554, 2716, 2717, 2733, 2747, 3073
`\print@edition` 2749, 2752
`\print@full@date` .. 2722, 2726, 2733
`\print@month@day` .. 2729, 2734
`\print@one@dash` ... 1659, 1691
`\print@partial` 2612, 2666, 2667, 2668, 2669
`\print@primary` 969, 972, 976, 2466
`\PrintAuthors` 5, 5, 6, 6, 6, 317, 380, 446, 464, 490, 976, 2472, 3081, 3128, 3172, 3190, 3206, 3258, 3315, 3329
`\PrintBackRefs` 823, 1756, 2810
`\PrintBook` ... 386, 2558, 3135, 3264
`\PrintCiteNames` ... 2965, 2967, 3035, 3038
`\PrintCNY` .. 2966, 3034
`\PrintConference` .. 385, 2532, 3134, 3263
`\PrintConferenceDetails` .. 403, 2535, 3276
`\PrintConferenceDetails@` . 2539, 2542, 2545
`\PrintContributions` 321, 342, 363, 384, 413, 429, 2501, 3086, 3112, 3133, 3156, 3210, 3227, 3244, 3262, 3287, 3302
`\PrintDate` ... 8, 8, 8, 467, 2706, 2711, 2716, 3072, 3082, 3105, 3129, 3151, 3173, 3191
`\PrintDateB` .. 369, 388, 419, 434, 452, 2717, 3216, 3232, 3249, 3267, 3292, 3307, 3321, 3332
`\PrintDateField` 469, 2718, 3195, 3334
`\PrintDatePosted` .. 2709, 2747
`\PrintDatePV` 325, 345, 2704
`\PrintDOI` . 329, 391, 2701, 3093, 3139
`\PrintEdition` 6, 7, 7, 7, 8, 360, 410, 428, 2748, 3109, 3155, 3248, 3291
`\PrintEditorsA` .. 6, 6, 7, 7, 7, 82, 83, 972, 2489, 3338
`\PrintEditorsB` 6, 7, 7, 361, 411, 2498, 3110
`\PrintEditorsC` 7, 7, 7, 2486
`\PrinteSeries@a` .. 2337
`\PrintISBNs` 2671
`\PrintNameList` 352, 2485, 3242, 3243, 3285, 3286
`\PrintNames` ... 82, 2429, 2477, 2485, 2487, 2494, 2499, 2594, 2601, 2606, 3340, 3345
`\PrintNames@a` 2430, 2432
`\PrintPartials` 322, 2663, 3087, 3211
`\PrintPrimary` . 6, 6, 356, 424, 2462, 2584, 3104, 3150, 3236, 3298
`\printref` 1757
`\PrintReprint` . 333, 373, 395, 439, 457, 2561, 3097, 3121, 3143, 3165, 3183, 3219, 3253, 3269, 3309, 3323
`\PrintReviews` . 7, 7, 336, 376, 398, 442, 460, 2660, 3100, 3124, 3146, 3168, 3186
`\PrintSeries` .. 78, 78, 79, 79, 82,

- 2328, 2433,
 2502, 2661,
 2664, 2672,
 3004, 3052, 3063
 \PrintSeries@a
 ... 79, 2335, 2337
 \PrintThesisType ..
 449,
 2686, 3176, 3318
 \PrintTranslation ..
 332,
 372, 394, 438,
 456, 2575, 3096,
 3120, 3142,
 3164, 3182,
 3218, 3252,
 3268, 3308, 3322
 \PrintTranslatorsA
 7, 7,
 83, 969, 2596, 3343
 \PrintTranslatorsB
 7, 7, 2605
 \PrintTranslatorsC
 7, 7, 362,
 412, 2593, 3111
 \PrintYear
 . 8, 8, 8, 90, 2746
 proceedings entry
 type 7
 \process@bare@dot ..
 2059, 2069
 \process@citelist ..
 1451,
 1589, 2800,
 2805, 2863, 2976
 \process@citelist@sorted
 1573, 1589
 \process@citelist@unsorted
 1578,
 1586, 2800,
 2805, 2863, 2976
 \process@dots .. 68,
 68, 71, 1980, 2042
 \process@hyphens 68,
 70, 71, 1979, 1989
 \process@names
 1981, 2093
 \process@one@dot ..
 70, 2045,
 2053, 2075, 2090
 \process@one@dot@a
 2055, 2057
 \process@one@dot@b
 2062, 2074
 \process@one@dot@c
 2064, 2077
 \process@one@dot@d
 . 2072, 2084, 2086
 \process@one@hyphen
 69, 1992,
 1999, 2021, 2036
 \process@one@hyphen@a
 70, 71, 2001, 2003
 \process@one@hyphen@b
 . 2009, 2012, 2020
 \process@one@hyphen@c
 2014, 2023
 \process@one@hyphen@d
 69, 2030, 2032
 \process@xrefs 677, 703
 \ProcessExclusiveOptions
 153
 \ProcessOptions .. 151
 \prop ... 286, 288, 2455
 \prop@reset 299
 \protected@edef ...
 1109,
 1163, 1212, 1221
 \ProvidesFile 1314
 \ProvidesPackage 3, 3203
 publisher field 33
 \purge@accent 1874, 1905
 \purge@edef
 . 1870, 2312, 2313
 \purge@edef@
 .. 1880, 1885,
 1895, 1896, 1900
 \purge@edef@a
 63, 1887, 1889
 \push@bracket 2616, 2619
 \PushCatcodes 37
- R**
- \r 1799
 \rasp 1834
 \read 1325
 \ReadBibData 1275, 1292
 \ReadBibData@a
 . 1304, 1308, 1313
 \ReadBibLoop
 . 1317, 1321, 1335
 \ReadBibLoop@a
 1326, 1330
 \ReadBibLoop@b
 1337, 1341
 \ReadBibLoop@e ... 1330
 \refname 41, 1199, 1219
 \RequirePackage ...
 4, 12,
 36, 125, 156,
 166, 179, 182,
 183, 2902, 3204
 \reset@nth@property 882
 \resetbiblist
 ... 596, 619, 1239
 \ResetCapSFCodes ..
 556, 603
 \resolve@inner 2520,
 2533, 2559,
 2562, 2576, 2613
 \restore@labelwidth
 544, 594
 \RestrictedSetKeys
 11,
 20, 298, 560,
 659, 681, 705,
 2513, 2523, 2925
 review field 5, 7
 rkeyval package ...
 5, 10, 20, 69
 \rkv@DSAK 20
 \rkv@setter .. 930, 950
 \rkvIfAdditive ... 276
 \rkvIfEmpty
 . 16, 186, 561,
 564, 2926, 2932
 \romannumeral 107
 \rsk@resume 747
 \rsk@set 28, 681
 \rsk@set@a 20
 \rsk@set@b 20
 \rsk@toks .. 10, 11,
 28, 31, 31, 299,
 560, 660, 705,
 2514, 2523, 2925
- S**
- \safe@set 82,
 1609, 1616,
 1632, 1666,
 1673, 1689, 2259
 \same@stemsfalse . 2319
 \same@stemstrue .. 2317
 \sameauthors . 6, 7,
 7, 7, 7, 7, 2474,
 2480, 2491, 2598
 \save@labelwidth ..
 573, 587
 \save@primary . 825, 958
 \section* 41
 \selbibdef@a . 750, 752
 \selbibdef@b . 754, 757
 \select@auxlanguage
 ... 278, 285, 2453
 \selective@bibdef .
 27,
 51, 749, 1264, 1562
 \selectlanguage ...
 . 36, 37, 288, 1048
 \SentenceSpace 336,
 376, 398, 442,
 460, 1754, 2789,
 3100, 3124,
 3146, 3168, 3186
 \series@add 2346, 2350
 \series@add@... .. 79

- `\series@add@a` 2330, 2362
 - `\series@add@b` 2331, 2365
 - `\series@add@c`
 - . 2332, 2367, 2374
 - `\series@add@d` 2333, 2371
 - `\series@add@e` 2334, 2356
 - `\series@index`
 - 20, 267,
 - 296, 843, 855,
 - 882, 2327, 2345,
 - 2351, 2353,
 - 2358, 2360,
 - 2370, 2373, 2512
 - `\series@total`
 - .. 2339, 2340,
 - 2364, 2370, 2373
 - `\set` 688,
 - 689, 690, 691,
 - 692, 693, 694,
 - 695, 696, 697, 698
 - `\set@firstname`
 - . 2430, 2440, 3031
 - `\set@name` ... 2441,
 - 2444, 2446, 3032
 - `\set@name@a`
 - ... 83, 2448, 2450
 - `\set@namea` 2450
 - `\set@othername`
 - .. 2430, 2435,
 - 2436, 2437, 2443
 - `\setbib@` 27, 28, 28,
 - 28, 30, 30, 31, 31
 - `\setbib@@` . 780, 789,
 - 794, 797, 800,
 - 803, 1002, 2590
 - `\setbib@article` ...
 - 789, 1002
 - `\setbib@book` 2572
 - `\setbib@conference` 2533
 - `\setbib@contribution`
 - 2514
 - `\setbib@incollection`
 - 483, 794, 797, 800
 - `\setbib@innerbook` 2559
 - `\setbib@inproceedings`
 - 486
 - `\setbib@misc` 657
 - `\setbib@nameBE`
 - 2419, 2456
 - `\setbib@nameinverted`
 - 2424, 3032
 - `\setbib@nameLE`
 - . 2414, 2441, 2444
 - `\setbib@partial` .. 2613
 - `\setbib@TYPE` 18
 - `\setcounter` 501
 - `\settowidth` .. 580, 620
 - `\sfcode` 81,
 - 115, 116, 117,
 - 118, 119, 120,
 - 606, 2398, 2556
 - `\shade@cite`
 - . 1757, 1758, 1765
 - `shaderef` package .. 58
 - `short-journals` op-
 - tion 15, 93
 - `short-months` option 94
 - `short-publishers` op-
 - tion .. 15, 94, 94
 - `shortalphabetic` op-
 - tion 92
 - `showkeys` package ..
 - 46, 58, 58
 - `\SingularPlural` ...
 - .. 7, 7, 79, 81,
 - 2341, 2343, 2412
 - `\sixt@cn` ... 1355, 1361
 - `\SK@` .. 1101, 1762, 1780
 - `\SK@@label` . 1101, 1779
 - `\SK@@ref` 1762
 - `\soft` 1809
 - `sorted` option 14
 - `\space` 69
 - `\spacefactor` . 111,
 - 123, 2398, 2400,
 - 2556, 2652, 2789
 - `\SS` 1830
 - `\ss` 1830
 - `\star@` ... 739, 1405,
 - 1463, 2978,
 - 2991, 2995,
 - 3014, 3022, 3028
 - `\star@bibdef` .. 27,
 - 633, 664, 721, 1316
 - `\stepcounter` 538
 - `\string` 62
 - `\strip@MRprefix` ...
 - 2643, 2654
 - `\strip@prefix` ... 20, 52
 - `\SubEtal` 2354, 2356, 2610
 - `\surround@names` ...
 - 2214, 2216
 - `\SwapBreak` 80,
 - 80, 272, 2331,
 - 2332, 2333,
 - 2334, 2383,
 - 2499, 2553, 2606
 - `\symbol` 60, 60
- T**
- `\t` 1800
 - `tabbing` environment 62
 - `\textbf` 5, 324, 344, 3089
 - `textcmds` package ..
 - 8, 15, 87
 - `\textemdash` ... 8, 2641
 - `\textendash` 8, 1660, 2642
 - `\textfont` 23
 - `\textit` 80,
 - 318, 320, 341,
 - 357, 359, 381,
 - 383, 425, 427,
 - 447, 448, 3083,
 - 3085, 3106,
 - 3108, 3130,
 - 3132, 3152,
 - 3154, 3174, 3175
 - `\textquotedblleft` 2635
 - `\textquotedblright` 2638
 - `\textsc` 78
 - `\textsubscript` 6
 - `\textsuperscript` ... 6
 - `\texttt` 20
 - `\tf@brf` 2816, 2817
 - `\TH` 1831
 - `\th` 1831
 - `\the` 28, 31, 31
 - `\thebib` 518,
 - 617, 620, 1071,
 - 1073, 1099, 2173
 - `thebibliography` envi-
 - ronment ... 1237
 - `\thebibliography` .. 92
 - `\thesis@type` 2687, 2690
 - `\thinspace` 2482
 - `\thr@@` 2364, 2770
 - `\toks@` 18, 65
 - `\tprime` 6, 6, 6, 6
 - `\TrailingHyphenWarning`
 - 2008, 2039
 - `transition` field ...
 - ... 5, 5, 5, 18, 32
 - `\tsub` 6, 6
 - `\tsup` 6, 6
 - `\ttfamily` 1504
 - `\tw@` 2295, 2353
 - `\typeout` 173, 1270, 1272
- U**
- `\u` 1801
 - `\uarc` 1804
 - `\udot` 1807
 - `\undefined` 1223
 - `\UndefinedCiteWarning`
 - 1494, 1499
 - `\unkern` 2388
 - `\unpenalty` 2392
 - `\unskip` ... 68, 1988,
 - 2387, 2652, 2747
 - `\uparrow` 1754
 - `\upn` 87, 87,
 - 2616, 2618, 2631
 - `upref` package 87
 - `\upshape` 1456
 - `\url` 18, 468,
 - 2790, 3194, 3333

- `\use@accent`
 . 1844, 1850, 1856
`\utilde` 1803
- V**
- `\v` 1802
`\value` 533
`\vdef` 53, 641, 1410, 1475
`\voltext` ... 6, 6, 6,
 365, 415, 2674,
 3114, 3241, 3284
`\vphantom` 2914
- W**
- `\widowpenalty` 554
`\wrap@accent`
 ... 63, 1843, 1851
 wrapper macros, defi-
 nition 12
- X**
- `xref field` 34, 34
`\xref@add@toks` 912, 927
`\xref@append` . 913, 948
`\xref@check@a`
 831, 832, 833, 837
`\xref@check@aa` 844, 854
`\xref@check@ab` 866, 873
`\xref@check@b`
 834, 835, 890
`\xref@check@c`
 . 33, 830, 901, 906
`\xref@ID`
 916, 917, 918, 920
`\XRefWarning`
 ... 918, 955, 2526
`\XrefWarning` 955
`xyz.aux` 8
- Y**
- `xyz.bbl` 9, 9
`xyz.tex` 8
`y2k option` .. 73, 94, 128
`\ycite` 50, 1533,
 1535, 1539,
 1540, 2977, 3017
`\ycites` 1534, 2980
`\year@short`
 . 2257, 2264, 2892
- Z**
- `\Z` 608
`\zap@space`
 .. 643, 1412, 1477