

The **amsrefs** package

Michael Downes and David M. Jones
American Mathematical Society

Version 2.02, 2006/10/25

Contents

1	Introduction	3
2	Package options	3
3	More about the \bib command	3
3.1	Field names for the \bib command	3
3.2	Bibliography entry types	4
4	Customizing the bibliography style	5
5	Miscellaneous commands provided by the amsrefs package . . .	6
6	Implementation	9
6.1	Overview	9
6.1.1	Normal L ^A T _E X processing of cites	9
6.2	How cites are processed by amsrefs	10
6.3	Data structures	11
6.4	Preliminaries	12
6.5	Utilities	13
6.6	Declaring package options	16
6.6.1	The ? option	18
6.7	Loading auxiliary packages	18
6.7.1	The lite option	18
6.8	Key-value setup	18
6.8.1	Standard field names (the bib group)	18
6.8.2	Auxiliary properties (the prop group)	20
6.9	Bibliography type specifications	20
6.10	The standard bibliography types	23
6.11	The \biblist environment	26
6.12	Processing bibliography entries	29
6.12.1	\@bibdef Implementations	30
6.12.2	\bib@exec Implementations	34
6.12.3	Resolving cross-references	35
6.12.4	Bib field preprocessing	39
6.12.5	Date setup	40
6.12.6	Language setup	41
6.12.7	Citation label setup	42

6.12.8	Printing the bibliography	45
6.13	Name, journal and publisher abbreviations	46
6.14	Processing <code>.ltb</code> files	46
6.15	Citation processing	49
6.15.1	The <code>\citesel</code> structure	49
6.15.2	The basic <code>\cite</code> command	50
6.15.3	Fancier <code>\cite</code> commands	55
6.15.4	The <code>\nocite</code> command	56
6.15.5	Citation sorting	57
6.15.6	Range compression	59
6.15.7	Munging the <code>.aux</code> file	62
6.15.8	Back references	64
6.15.9	<code>hyperref</code> , <code>showkeys</code> and <code>shaderef</code> support . . .	64
6.16	Lexical structure of names	65
6.16.1	Text accents	65
6.16.2	Text symbols	67
6.16.3	<code>\edef</code> -like macros for names	68
6.17	Name parsing	71
6.18	Extracting initials	75
6.18.1	The algorithm	76
6.18.2	The implementation	76
6.19	Generating alphabetic labels	82
6.19.1	The algorithm	82
6.19.2	The implementation	83
6.20	Generating short alphabetic labels	87
6.21	Formatting series	88
6.22	Formatting names and series of names	92
6.23	The <code>partial</code> field	98
6.24	Special formatting for other fields	98
6.25	BIB _{TEX} support	104
6.26	Implementing package options	104
6.26.1	The <code>alphabetic</code> option	104
6.26.2	The <code>shortalphabetic</code> option	104
6.26.3	The <code>backrefs</code> option	105
6.26.4	The <code>citation-order</code> option	105
6.26.5	The <code>initials</code> option	105
6.26.6	The <code>logical-quotes</code> option	106
6.26.7	The <code>non-compressed-cites</code> option	106
6.26.8	The <code>non-sorted-cites</code> option	106
6.26.9	The <code>short-journals</code> option	106
6.26.10	The <code>short-publishers</code> option	106
6.26.11	The <code>short-months</code> option	107
6.26.12	The <code>y2k</code> option	107
6.26.13	The <code>bibtex-style</code> option	107
6.26.14	The <code>author-year</code> option	107
6.27	The <code>amsbst</code> package	114

1 Introduction

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

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

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

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

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

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

2 Package options

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

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

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

3 More about the `\bib` command

3.1 Field names for the `\bib` command

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

fulljournal Used internally by `\DefineJournal`.

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

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

The following fields are included for backwards compatibility:

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

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

The following fields are reserved for future use:

doi Digital Object Identifier

setup This is a special field that can be used to give arbitrary commands to be executed at the beginning of the current `\bib` entry, after all the fields have been read. The idea is that one can alter the formatting of an individual entry through this field, to handle special cases.

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

url Universal Resource Locator.

3.2 Bibliography entry types

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

collection.article

proceedings.article

partial

conference

innerbook

name

nameLE

nameBE

nameinverted

publisher

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

miscellaneous

periodical

4 Customizing the bibliography style

If you use the `amsrefs` package as is, the bibliography style you get is the kind of style customarily seen in AMS publications. The recommended way to get a different bibliography style is to write a \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 L^AT_EX, 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.

`\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`.

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.

\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 BibTeX .

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

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

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

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

Second \LaTeX pass Now the `.bbl` file exists and contains some `\bibitem` commands. At `\begin{document}`, \LaTeX reads the `.aux` file, hoping to find some `\bibtex` 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 L^AT_EX pass Now the `.aux` file contains some `\bibcite` commands. Once again, L^AT_EX reads the `.aux` file when it reaches `\begin{document}`.

1. A `\bibcite{moo}{Moody}` causes L^AT_EX to define `\b@moo` with ‘Moody’ as the replacement text.
2. If two `\bibcite` commands have the same citation key, L^AT_EX 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, L^AT_EX will give an ‘Undefined citation’ warning. This often happens if the `.aux` file is incomplete due to a T_EX error on the preceding pass.

6.2 How cites are processed by `amsrefs`

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

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

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

We therefore have

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

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

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

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

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

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

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

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

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

6.3 Data structures

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

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

The scanned result is to assign

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

```

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

```

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

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

6.4 Preliminaries

```
1 <*pkg>
```

Standard declaration of package name and date.

```
2 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3 \ProvidesPackage{amsrefs}[2006/10/25 v2.02]
```

```
\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.

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

```

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

```

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

```

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

```

`\macrotext`

```

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

```

`\vdef` “Verbatim” def.

```

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

```

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

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

```

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

```

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.

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

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

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

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

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

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

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

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

```

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

```

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

```

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

```

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

```

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

```

\amsrefs@warning

```

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

```

\amsrefs@error

```

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

```

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

```

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

```

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

```

107 \@ifundefined{\@addpunct}{%

```

```

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

```

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

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

`\@nopunctsfcode`

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

6.6 Declaring package options

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

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

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

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

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

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

`y2k`

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

`nobysame`

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

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

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

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

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

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

```
132 \DeclareBooleanOption{initials}
```

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

```
133 \DeclareOption{abbrev}{%
```

```
134     \@pass@options
```

```
135     \@currentx
```

```
136     {initials,short-months,short-journals,short-publishers}%
```

```
137     \@currname
```

```
138 }
```

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

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

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

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

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

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

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

Option for giving information about the available options:

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

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

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

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

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

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

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

```
148
```

```
149 \ProcessOptions\relax
```

```
150
```

```
151 \ProcessExclusiveOptions
```

```
152 \IfOption{backrefs}{%
```

```
153     \IfFileExists{hyperref.sty}{%
```

```
154         \RequirePackage{hyperref}[1999/07/08]
```

```
155     }{}}%
```

```

156 \IfFileExists{backref.sty}{%
157     \RequirePackage{backref}[1999/05/30]
158 }{}%
159 }{}

```

6.6.1 The ? option

Note that in the following auxiliary package list, `getwidth` is not (yet) included.

```

160 \IfOption{?}{%
161     \typeout{^^J%
162         Documentation for the amsrefs package is found in amsrdoc.dvi^^J%
163         (or .pdf or .tex).
164     ^^J%
165 }%
166 }{}%

```

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.

```

167 \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.

```

168 \IfOption{lite}{% True? Then don't load the next two packages.
169 }{% False? OK, let's load them:
170     \RequirePackage{textcmds}[2001/12/14]
171     \RequirePackage{mathscinet}[2002/01/01]
172 }

```

6.8 Key-value setup

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

```

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

```

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

```

174 \newcommand{\IfEmptyBibField}{\rkvIfEmpty{bib}}

```

6.8.1 Standard field names (the bib group)

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

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

```

175 \let\fld@elt=?
176 \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.

```

177 \DefineAdditiveKey{bib}{author}{\name}
178 \DefineAdditiveKey{bib}{editor}{\name}
179 \DefineAdditiveKey{bib}{translator}{\name}
180 \DefineAdditiveKey{bib}{contribution}{\fld@elt}
181 \DefineAdditiveKey{bib}{isbn}{\fld@elt}
182 \DefineAdditiveKey{bib}{issn}{\fld@elt}
183 \DefineAdditiveKey{bib}{review}{\fld@elt}
184 \DefineAdditiveKey{bib}{partial}{\fld@elt}

185 \DefineSimpleKey{bib}{address}
186 \DefineSimpleKey{bib}{book}
187 \DefineSimpleKey{bib}{booktitle}
188 \DefineSimpleKey{bib}{conference}
189 %\DefineSimpleKey{bib}{contributor}
190 \DefineSimpleKey{bib}{copula}
191 \DefineSimpleKey{bib}{date}
192 \DefineSimpleKey{bib}{doi}
193 \DefineSimpleKey{bib}{edition}
194 \DefineSimpleKey{bib}{eprint}
195 \DefineSimpleKey{bib}{fulljournal}
196 \DefineSimpleKey{bib}{hyphenation}
197 \DefineSimpleKey{bib}{institution}
198 \DefineSimpleKey{bib}{journal}
199 \DefineSimpleKey{bib}{label}
200 \DefineSimpleKey{bib}{language}
201 \DefineSimpleKey{bib}{name}
202 \DefineSimpleKey{bib}{note}
203 \DefineSimpleKey{bib}{number}
204 \DefineSimpleKey{bib}{organization}
205 \DefineSimpleKey{bib}{pages}
206 \DefineSimpleKey{bib}{part}
207 \DefineSimpleKey{bib}{place}
208 \DefineSimpleKey{bib}{publisher}
209 \DefineSimpleKey{bib}{reprint}
210 \DefineSimpleKey{bib}{school}
211 \DefineSimpleKey{bib}{series}
212 \DefineSimpleKey{bib}{setup}
213 \DefineSimpleKey{bib}{status}
214 \DefineSimpleKey{bib}{subtitle}
215 \DefineSimpleKey{bib}{title}
216 \DefineSimpleKey{bib}{translation}
217 \DefineSimpleKey{bib}{type}
218 \DefineSimpleKey{bib}{url}
219 \DefineSimpleKey{bib}{volume}
220 \DefineSimpleKey{bib}{xref}
221 \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.

```
222 \DefineDummyKey{bib}{transition}
```

6.8.2 Auxiliary properties (the prop group)

```
223 \DefineSimpleKey{prop}{inverted}
```

```
224 \DefineSimpleKey{prop}{language}
```

6.9 Bibliography type specifications

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

```
225 \newcommand{\BibSpec}[2]{%
```

```
226   \toks@{\@emptytoks}
```

```
227   \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.

```
228       \@ifnextchar{+}{\@xp\bibspeg@scan\@gobble}{\bibspeg@scan}%
```

```
229       #2\@empty\@empty\@empty\@empty
```

```
230   }%
```

```
231   \@xp\edef\csname setbib@#1\endcsname{\the\toks@}%
```

```
232 }
```

`\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.

```
233 \def\bibspeg@scan#1#2#3#4{%
```

```
234   \add@toks@{\bib@append{#1}{#2}}%
```

```
235   \edef\@tempa{%
```

```
236     \toks@{\the\toks@ \@xp\@nx\csname bib'#3\endcsname}%
```

```
237   }%
```

```
238   \@tempa
```

```
239   \ifx\@empty#4%
```

```
240     \@xp\@gobble % end the recursion
```

```
241   \else
```

```
242     \ifx +#4\else\bibspeg@scan@error\fi
```

```

243     \fi
244     \bibspec@scan
245 }

```

```
\bibspec@scan@error
```

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

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

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

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

```

247 \def\bib@append#1#2#3{%
248     \ifx\@empty#3%
249     \else
        Known bug: Need better error message here.
250     \ifx\relax#3%
251         \errmessage{#3=\relax}%
252     \else
253         \begingroup
254             \series@index\m@ne
255             \def\current@bibfield{#3}%
256             \@ifempty{#1}{%
257                 \@temptokena{\ifnum\lastkern=\@ne\ignorespaces\fi #2}%
258             }{%
259                 \@temptokena{\SwapBreak{#1}#2}%
260             }%
261             \toks@\@xp{#3}%
262             \edef\@tempa{\the\@temptokena{\the\toks@}}%
263             \rkvalIfAdditive#3}{%
264                 \get@current@properties
265                 \select@auxlanguage
266             }%
267             \@tempa
268         \endgroup
269     \fi
270 \fi
271 }

```

```
\select@auxlanguage
```

```
272 \def\select@auxlanguage{%
```

```

273 \ifx\prop'language\@empty
274 \else
275 \xp\selectlanguage\xp{\prop'language}%
276 \fi
277 }

```

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

The obvious definition of `\erase@field` is

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

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

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

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

```

278 \def\erase@field#1{%
279 \aftergroup\let\aftergroup#1\aftergroup\@empty
280 }

```

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

```

281 \def\get@current@properties{%
282 \begingroup
283 \xp\get@nth@property\xp\@tempa\current@bibfield\series@index
284 \edef\@tempa{%
285 \nx\RestrictedSetKeys\{prop\}%
286 \def\@nx\@tempa{\nx\prop@reset \nx\the\@nx\rsk@toks}%
287 }\@tempa}%
288 }%
289 \@tempa
290 \xp\endgroup
291 \@tempa
292 }

```

\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.

```

293 \newcommand{\BibSpecAlias}[2]{%
294   \@xp\def\@xp\@tempa\@xp{\csname setbib@#1\@xp\endcsname}%
295   \@xp\ifx\csname setbib@#2\endcsname\@tempa
296     \amsrefs@error{%
297       Mirror alias #1->#2 not allowed (infinite loop)}\@ehc
298   \else
299     \@xp\def\csname setbib@#1\@xp\endcsname
300     \@xp{\csname setbib@#2\endcsname}%
301   \fi
302 }
```

6.10 The standard bibliography types

```

303 \BibSpec{article}{%
304   +{ } { \PrintAuthors}           {author}
305   +{,} { \textit}                 {title}
306   +{.} { }                        {part}
307   +{:} { \textit}                 {subtitle}
308   +{,} { \PrintContributions}     {contribution}
309   +{.} { \PrintPartials}          {partial}
310   +{,} { }                        {journal}
311   +{ } { \textbf}                 {volume}
```

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

```

312   +{ } { \PrintDatePV}            {date}
313   +{,} { \issuetext}              {number}
314   +{,} { \eprintpages}            {pages}
315   +{,} { }                        {status}
316   +{,} { \PrintDOI}               {doi}
317   +{,} { available at \eprint}    {eprint}
318   +{ } { \parenthesize}            {language}
319   +{ } { \PrintTranslation}        {translation}
320   +{;} { \PrintReprint}            {reprint}
321   +{.} { }                        {note}
322   +{.} { }                        {transition}
323   +{ } { \SentenceSpace \PrintReviews} {review}
324 }
325
326 \BibSpec{partial}{%
327   +{ } { }                        {part}
328   +{:} { \textit}                 {subtitle}
329   +{,} { \PrintContributions}     {contribution}
330   +{,} { }                        {journal}
331   +{ } { \textbf}                 {volume}
332   +{ } { \PrintDatePV}            {date}
333   +{,} { \issuetext}              {number}
334   +{,} { \eprintpages}            {pages}
335 }
336
```

```

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

```

```

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

```

```

437   +{,} { } {organization}
438   +{,} { } {address}
439   +{,} { \PrintDateB} {date}
440   +{,} { \eprint} {eprint}
441   +{,} { } {status}
442   +{ } { \parenthesize} {language}
443   +{ } { \PrintTranslation} {translation}
444   +{;} { \PrintReprint} {reprint}
445   +{.} { } {note}
446   +{.} {} {transition}
447   +{ } {\SentenceSpace \PrintReviews} {review}
448 }

449 \BibSpecAlias{periodical}{book}
450 \BibSpecAlias{collection}{book}
451 \BibSpecAlias{proceedings}{book}
452 \BibSpecAlias{manual}{book}
453 \BibSpecAlias{miscellaneous}{book}
454 \BibSpecAlias{misc}{miscellaneous}
455 \BibSpecAlias{unpublished}{book}
456 \BibSpecAlias{proceedings.article}{collection.article}
457 \BibSpecAlias{techreport}{report}

```

`\setbib@incollection`

```

458 \edef\setbib@incollection{%
459   \exp\@nx\csname setbib@collection.article\endcsname
460 }

```

`\setbib@inproceedings`

```

461 \edef\setbib@inproceedings{%
462   \exp\@nx\csname setbib@collection.article\endcsname
463 }

```

Some more entry types for implementing abbreviations.

```

464 \BibSpec{name}{%
465   +{ } {\PrintAuthors} {name}
466 }
467
468 \BibSpec{publisher}{%
469   +{,} { } {publisher}
470   +{,} { } {address}
471 }

```

6.11 The biblist environment

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

Use a standard L^AT_EX counter for numbering bibliography items.

```

472 \newcounter{bib}

```

`biblist`

```

473 \newenvironment{biblist}{%

```

```

474 \setcounter{bib}\z@
475 \@biblist
476 }{%
477 \@endbiblist
478 }

```

biblist*

```

479 \newenvironment{biblist*}{%
480 \@biblist
481 }{%
482 \@endbiblist
483 }

```

\@biblist

```

484 \newcommand\@biblist[1][1]{%
485 \stepcounter{bib@env}
486 \normalfont
487 \footnotesize
488 \labelsep .5em\relax
489 \list{\BibLabel}{%
490 \restore@labelwidth
491 \@maxlabelwidth\z@
492 \@nmbrlisttrue
493 \def\@listctr{bib}%
494 \let\makelabel\@bib@mklab
495 #1\relax
496 }%
497 \sloppy

```

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

```

498 \interlinepenalty\@m
499 \clubpenalty\@M
500 \widowpenalty\clubpenalty
501 \frenchspacing
502 \ResetCapSFCodes
503 }

```

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

```

504 \def\@endbiblist{%
505 \save@labelwidth
506 \def\@noitemerr{\@latex@warning{Empty bibliography list}}%
507 \endlist
508 }

```

\@maxlabelwidth

```

509 \newdimen\@maxlabelwidth

```

\bib@mklab

```
510 \def\bib@mklab#1{%
511     \settowidth\@tempdima{#1}%
512     \ifdim \@tempdima > \@maxlabelwidth
513         \global\@maxlabelwidth\@tempdima
514     \fi
515     #1\hfil
516 }
```

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

\save@labelwidth

```
518 \def\save@labelwidth{%
519     \if@files
520         \immediate\write\@auxout{%
521             \string\newlabel{[bibenv:\the\c@bib@env]}\the\@maxlabelwidth}%
522         }%
523     \fi
524 }
```

\restore@labelwidth

```
525 \def\restore@labelwidth{%
526     \@xp\ifx \c@name r@[bibenv:\the\c@bib@env]\endcsname \relax
527     \resetbiblist{00}%
528     \else
529         \@xp\labelwidth\c@name r@[bibenv:\the\c@bib@env]\endcsname
530         \leftmargin\labelwidth
531         \advance\leftmargin\labelsep
532     \fi
533 }
```

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

```
534 \providecommand{\ResetCapSFCodes}{%
535     \count@='A
536     \def\@tempa{%
537         \sfcode\count@=\@m
538         \advance\count@\@ne
539         \ifnum\count@>'Z\relax \expandafter\@gobble \fi
540     \@tempa
541     }%
542     \@tempa
543 }
```

\CurrentBib In case this is undefined sometimes.

```
544 \def\CurrentBib{??}
```

\BibLabel

```
545 \newcommand{\BibLabel}{%}
```

```

546     [\hyper@anchorstart{cite.\CurrentBib}\relax\thebib\hyper@anchorend]%
547 }

```

`\resetbiblist`

```

548 \newcommand{\resetbiblist}[1]{%
549     \settowidth\labelwidth{\def\thebib{#1}\BibLabel}%
550     \leftmargin\labelwidth
551     \ifdim\labelwidth=\z@
552         \leftmargin=1em
553         \itemindent=-\leftmargin
554     \else
555         \advance\leftmargin\labelsep
556     \fi
557 }

```

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.

```

558 \newcommand{\bib}{%
559     \begingroup
560     \@ifstar{%
561         \@tempswatrue
562         \let\@bibdef\star@bibdef
563         \BibItem
564     }{%
565         \@tempswafalse
566         \BibItem
567     }%
568 }

```

`\BibItem` *Arguments:*

```

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

```

```

569 \newcommand{\BibItem}[2]{%
570     \def\@tempa{#1}%
571     \edef\@tempb{%
572         \@nx\@bibdef\@xp\@nx\curname setbib@#2\endcsname{#2}%
573         {\macrotext\@tempa}%
574     }%
575     \@tempb
576 }

```

`\@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.
```

```
577 \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.
```

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

6.12.1 `\@bibdef` Implementations

`\normal@bibdef` *Arguments:*

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

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

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

```
580   \def\CurrentBibType{#2}%
581   \ifx\relax#1%
582     \amsrefs@error{Undefined entry type: #2}\@ehc
583     \let#1\setbib@misc
584   \fi
585   \RestrictedSetKeys{}{bib}%
586   {\bib@exec{#3}{\the\rsk@toks}{#1}\endgroup}%
587 }
588
589 \let\@bibdef\normal@bibdef
```

```

\star@bibdef  Arguments:
               #1 <- \setbib@bibtype.
               #2 <- bibtype.
               #3 <- citekey.

590 \def\star@bibdef{%
591     \let\bib@exec\bib@store
592     \normal@bibdef
593 }

```

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

```

594 \def\copy@bibdef{%
595     \if@tempwa
596         \@xp\defer@bibdef
597     \else
598         \@xp\copy@bibdef@a
599     \fi
600 }

```

\copy@bibdef@a

```

601 \def\copy@bibdef@a#1#2#3#4{%
602     \@open@bbl@file
603     \process@xrefs{#4}%
604     \bbl@write{%
605         \string\bib\if@tempwa*\fi{#3}{#2}\string{\iffalse}\fi
606     }%

```

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.

```

607     \RestrictedSetKeys{\global\let\rsk@set\bbl@copy}\@empty
608     {\bbl@write{\iffalse{\fi\string}^^J}}%
609     \endgroup}{#4}%
610 }

```

```

611 \catcode'\:=11
612

```

```

613 \def\modify@xref@fields{%
614     \let\set:bib'author\output@xref@a
615     \let\set:bib'editor\output@xref@a
616     \let\set:bib'translator\output@xref@a
617     \let\set:bib'journal\output@xref@a
618     \let\set:bib'publisher\output@xref@a
619     \def\set:bib'xref##1##2{\output@xref@{##1}\@empty}%
620     \def\set:bib'book##1##2{\output@inner@xref@{##1}\@empty}%
621     \let\set:bib'conference\set:bib'book
622     \let\set:bib'partial\set:bib'book
623     \let\set:bib'reprint\set:bib'book
624     \let\set:bib'translation\set:bib'book
625 }

```

```

626
627 \catcode'\:=12
628
629 \def\process@xrefs#1{%
630     \begingroup
631         \RestrictedSetKeys{\modify@xref@fields}{bib}{\the\rsk@toks}{#1}%
632     \endgroup
633 }
634
635 \def\output@xref@a#1#2{%
636     \def\@tempa{#1}%
637     \lowercase{\def\@tempb{#1}}%
638     \ifx\@tempa\@tempb
639         \output@xref@{#1}%
640     \fi
641 }
642
643 \def\output@xref@#1{%
644     \@ifnotempty{#1}{%
645         \@ifundefined{bi@#1}{%
646             \begingroup
647                 \let\star@bibdef\copy@bibdef@a
648                 \csname bi@#1\endcsname
649             \endgroup
650         }%
651         \@xp\g@undef\csname bi@#1\endcsname
652     }%
653 }
654
655 \def\output@inner@xref@#1{%
656     \in@={#1}%
657     \ifin@\else
658         \output@xref@{#1}%
659     \fi
660 }

```

\bbl@copy

```

661 \def\bbl@copy#1\endcsname#2{%
662     \begingroup
663         \def\@tempa{#1}%
664         \toks@{{#2}}%
665         \star@{\bbl@copy@a}{}%
666 }

```

\bbl@copy@a

```

667 \def\bbl@copy@a#1{%
668     \@ifnotempty{#1}{%
669         \add@toks@{*{#1}}%
670     }%
671     \bbl@write{ \space\@tempa=\the\toks@,}%

```

```

672   \endgroup
673   \rsk@resume
674 }

```

`\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.

```

```

675 \def\selective@bibdef#1#2#3{%
676   \xp\selbibdef@a\csname b@#3\endcsname{#1}{#2}{#3}%
677 }

```

`\selbibdef@a`

```

678 \def\selbibdef@a#1{%
679   \def\@tempa{\endgroup\@gobblefour}%
680   \ifx\relax#1\else \xp\selbibdef@b#1\@nil \fi
681   \@tempa
682 }

```

`\selbibdef@b`

```

683 \def\selbibdef@b#1#2#3\@nil{%
684   \ifx 1#2\let\@tempa\copy@bibdef\fi
685 }

```

`\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.

```

```

686 \def\defer@bibdef#1#2#3#4{%
687   \xp\gdef\csname bi@#3\endcsname{%
688     \bib*{#3}{#2}{#4}%
689   }%
690   \xp\addto@defer@list \csname bi@#3\endcsname
691   \endgroup
692 }

```

`\bibdefer@list`

```

693 \let\bibdefer@list\@empty

```

`\addto@defer@list`

```

694 \def\addto@defer@list#1{%
695   \begingroup

```

```

696      \def\do{\@nx\do\@nx}%
697      \xdef\bibdefer@list{\bibdefer@list\do#1}%
698  \endgroup
699 }

```

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.

```

700 \def\bib@store#1{%
701   \afterassignment\@gobble
702   \@xp\xdef\csname bi@#1\endcsname
703 }

```

\numeric@refs

```

704 \def\numeric@refs{00}

```

\bib@print *Arguments:*

```

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

```

```

705 \def\bib@print#1#2#3{%
706   \bib@start{#1}%
707   \let\setbib@@#3%
708   #2\relax      % execute definitions locally
709   \bib@resolve@xrefs
710   \bib@field@patches
711   \bib@selectlanguage
712   \generate@label
713   \bib'setup
714   \bib@cite{#1}%
715   \kern\@ne sp
716   \ifx\setbib@@\setbib@article
717     \ifx\bib'booktitle\@empty
718       \ifx\bib'book\@empty
719         \ifx\bib'conference\@empty
720           \else
721             \let\setbib@@\setbib@incollection
722           \fi
723         \else
724           \let\setbib@@\setbib@incollection
725         \fi
726       \else
727         \let\setbib@@\setbib@incollection
728       \fi
729     \fi
730     \setbib@@
731   \bib@end
732 }

```

`\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.`

```

733 \def\bib@print@inner#1#2{%
734   \begingroup
735     #2\relax      % execute definitions locally
736     \bib@resolve@xrefs
737     \bib@field@patches
738     \bib'setup
739     #1%
740   \endgroup
741 }
```

`\current@citekey`

```

742 \let\current@citekey\@empty
```

`\prev@citekey`

```

743 \let\prev@citekey\@empty
```

`\bib@start` There used to be more to it.

```

744 \def\bib@start#1{%
745   \begingroup
746     \def\current@citekey{#1}%
747 }
```

`\bib@end` Instead of being handled by `\bib@end`, ending punctuation is normally handled via the transition field (q.v.)

```

748 \def\bib@end{%
749   \relax
750   \@xp\PrintBackRefs\@xp{\CurrentBib}%
751   \par
752   \save@primary
753   \global\let\prev@citekey\current@citekey
754   \endgroup
755 }
```

6.12.3 Resolving cross-references

`\bib@resolve@xrefs`

```

756 \def\bib@resolve@xrefs{%
757   \xref@check@c\bib'xref
758   \xref@check@a\bib'author
759   \xref@check@a\bib'editor
760   \xref@check@a\bib'translator
761   \xref@check@b\bib'journal
762   \xref@check@b\bib'publisher
763 }
```

`\xref@check@a` Resolve a contributor (typically a `\DefineName`) alias. Requires rebuilding the list.

```

764 \def\xref@check@a#1{%
765     \ifx\@empty#1\relax
766     \else
767         \begingroup
768             \toks@\@emptytoks
769             \@temptokenb\@emptytoks
770             \series@index\z@
771             \def\name{\xref@check@aa#1}%
772             #1\relax
773             \edef\@tempa{%
774                 \def\@nx#1{\the\toks@}%
775                 \the\@temptokenb
776             }%
777             \@xp\endgroup
778             \@tempa
779     \fi
780 }
```

`\xref@check@aa`

```

781 \def\xref@check@aa#1#2{%
782     \advance\series@index\@ne
783     \def\@tempa{#2}%
784     \lowercase{\def\@tempb{#2}}%
785     \ifx\@tempa\@tempb
786         \ifx\@tempa\@empty
787             \add@toks@\{\name{}\}%
788         \else
789             \@ifundefined{bi@#2}{%
790                 \BibAbbrevWarning{#2}%
791                 \add@toks@\{\name{#2}}%
792             }{%
793                 \xref@check@ab#1{#2}%
794             }%
795         \fi
796     \else
797         \add@toks@\{\name{#2}}%
798     \fi
799 }
```

`\xref@check@ab`

```

800 \def\xref@check@ab#1#2{%
801     \csname bi@#2\endcsname
802     \ifx\@empty\bib'name
803         \@temptokena{#2}%
804     \else
805         \@temptokena\@xp{\bib'name}%
806         \get@property\@tempa\bib'name

```

```

807     \edef\@tempa{%
808         \@nx\addto@hook\@temptokenb{%
809             \@nx\reset@nth@property\@nx#1\the\series@index{\@tempa}%
810         }%
811     }%
812     \@tempa
813 \fi
814 \edef\@tempa{\@nx\add@toks@{\@nx\name{\the\@temptokena}}}%
815 \@tempa
816 }

```

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

```

817 \def\xref@check@b#1{%
818     \ifx\@empty#1%
819     \else
820         \toks@{\xp{#1}%
821             \edef\@tempb{\lowercase{\def\@nx\@tempa{\the\toks@}}}%
822             \@tempb
823             \ifx\@tempa#1\relax % all lowercase
824                 \@ifundefined{bi@#1}{%
825                     \BibAbbrevWarning{#1}%
826                 }{%

```

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.

```

827         \let#1\@empty
828         \xref@check@c\@tempa
829     }%
830 \fi
831 \fi
832 }

```

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

```

833 \def\xref@check@c#1{%
834     \ifx#1\@empty
835     \else
836         \begingroup
837             \@apply\auto@protect\amsrefs@textsymbols
838             \@apply\auto@protect\amsrefs@textaccents
839             \let\DSK@def\xref@add@toks
840             \let\DSK@append\xref@append
841             \toks@\@emptytoks
842             \let\bib@reset\@empty

```

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

```

843         \@for\xref@ID:=#1\do{%
844             \@ifundefined{bi\xref@ID}{%
845                 \XRefWarning{\xref@ID}%

```

```

846         }{%
847             \csname bi@\xref@ID\endcsname
848         }%
849     }%
850     \edef\@tempa{\endgroup\the\toks@}%
851     \@tempa
852     \fi
853 }

```

`\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.

```

```

854 \def\xref@add@toks#1#2#3{%
855     \ifx#1\@empty
856         \edef\@tempa{%
857             \@nx\add@toks@{\@xp\@nx\csname\rkv@setter#1\endcsname{#2}{#3}}%
858         }%
859         \@tempa
860     \else
861         \in@\bib'title{#1}%
862         \ifin@
863             \ifx\bib'booktitle\@empty
864                 \edef\@tempa{%
865                     \@nx\add@toks@{%
866                         \@xp\@nx\csname set:bib'booktitle\endcsname
867                     }%
868                 }%
869                 \@tempa
870                 \add@toks@{{#2}{#3}}%
871             \fi
872         \fi
873     \fi
874 }

875 \def\xref@append#1#2#3#4{%
876     \edef\@tempa{%
877         \@nx\add@toks@{\@xp\@nx\csname\rkv@setter#2\endcsname{#3}{#4}}%
878     }%
879     \@tempa
880 }

```

`\BibAbbrevWarning`

```

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

```

`\XrefWarning`

```

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

```

6.12.4 Bib field preprocessing

```

\current@primary
883 \let\current@primary\@empty

\previous@primary
884 \let\previous@primary\@empty

\save@primary
885 \IfOption{nobysame}{%
886   \let\save@primary\@empty
887 }{%
888   \def\save@primary{%
889     \global\let\previous@primary\current@primary
890   }%
891 }

\bib@field@patches
892 \def\bib@field@patches{%
893   \ifx\bib'author\@empty
894     \ifx\bib'editor\@empty
895       \let\current@primary\bib'translator
896       \let\print@primary\PrintTranslatorsA
897     \else
898       \let\current@primary\bib'editor
899       \let\print@primary\PrintEditorsA
900     \fi
901   \else
902     \let\current@primary\bib'author
903     \let\print@primary\PrintAuthors
904   \fi
905   \ifx\bib'address\@empty
906     \let\bib'address\bib'place
907   \fi
908   \ifx\bib'organization\@empty
909     \ifx\bib'institution\@empty
910       \let\bib'organization\bib'school
911     \else
912       \let\bib'organization\bib'institution
913     \fi
914   \fi
915   \ifx\bib'date\@empty
916     \ifx\bib'year\@empty
917       \let\bib@year\bib'status
918     \else
919       \bib@parsedate\bib'year
920     \fi
921   \else
922     \bib@parsedate\bib'date
923   \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.

```

924 \def\@tempa{year}%
925 \ifx\bib'volume\@tempa
926 \let\bib'volume\bib@year
927 \let\bib'date\@empty
928 \fi

```

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

```

929 \bib@language@fixup
930 }

```

6.12.5 Date setup

`\bib@year`

```

931 \let\bib@year\@empty

```

`\bib@month`

```

932 \let\bib@month\@empty

```

`\bib@day`

```

933 \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.

```

934 \def\bib@parsedate#1{%
935 \exp\bib@parsedate@a#1---\@nil
936 }

```

`\bib@parsedate@a`

```

937 \def\bib@parsedate@a#1-#2-#3-#4\@nil{%
938 \def\bib@year{#1}%
939 \def\bib@month{#2}%
940 \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.

```

941 \ifx\@empty\bib@day
942 \ifx\@empty\bib@month
943 \let\bib'date\bib@year
944 \else
945 \def\bib'date{#1-#2}%
946 \fi
947 \else
948 \def\bib'date{#1-#2-#3}%
949 \fi
950 }

```

6.12.6 Language setup

`\bib@language@fixup`

```

951 \def\bib@language@fixup{%
952   \ifx\bib'hyphenation\@empty
953     \ifx\bib'language\@empty
954       \let\bib@language\biblanguagedefault
955     \else
956       \let\bib@language\bib'language
957     \fi
958   \else
959     \let\bib@language\bib'hyphenation
960   \fi
961   \def\@tempa##1 ##2\@nil{\lowercase{\def\bib@language{##1}}}%
  The mysterious \@firstofone here is to preserve the space before the \@nil.
962   \@firstofone{\@xp\@tempa\bib@language} \@nil
963 }
```

`\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.

```

964 \def\bib@selectlanguage{%
965   \@ifsame@patterns{\language\name}{\bib@language}{\}%
966   \@ifundefined{date\bib@language}{\}%
967   \@xp\selectlanguage\@xp{\bib@language}%
968   }%
969 }%
970 }
```

`\@ifsame@patterns`

```

971 \def\@ifsame@patterns#1#2{%
972   \@xp\@ifsamepat\csname l@#1\@xp\endcsname\csname l@#2\endcsname
973 }
```

`\@ifsamepat`

```

974 \def\@ifsamepat#1#2{%
975   \ifnum \ifx\relax#1\m@ne\else#1\fi = \ifx\relax#2\m@ne\else#2\fi
976   \@xp\@firstoftwo
977   \else
978     \@xp\@secondoftwo
979   \fi
980 }
```

`\language\name`
`\biblanguageEnglish`
`\biblanguagedefault`
`\bib@language`

```

981 \providecommand{\language}{english}
982 \def\biblanguageEnglish{english}
983 \let\biblanguagedefault\biblanguageEnglish
984 \let\bib@language\@empty

```

6.12.7 Citation label setup

`\generate@label`

```
985 \let\generate@label\relax
```

`\cite@label`

```
986 \def\cite@label{\@currentlabel}
```

`\alpha@label`

```
987 \let\alpha@label\relax
```

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

Arguments:

`#1` <- *citekey*.

```

988 \def\bib@cite#1{%
989     \def\CurrentBib{#1}%
990     \alpha@label           % modify \thebib if necessary
991     \item\leavevmode
992     \SK@\SK@@label{#1}%
993     \@xp\bib@cite@a\csname b@#1\endcsname
994     \bibcite@write{#1}%
995 }

996 \def\bib@cite@a#1{%
997     \ifx\relax#1%
998         \begingroup
999         \auto@protect\etaltext
1000         \protected@edef\@tempa{%
1001             \gdef\@nx#1{%
1002                 \@nx\citesel 01{\cite@label}{\bib@label@year}{}}%
1003             }%
1004         }%
1005     \@xp\endgroup
1006     \@tempa
1007 \else
1008     \@xp\bib@cite@check\@xp#1#1\@empty\@empty\@empty\@empty\@empty
1009 \fi
1010 }

```

`\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.
```

```
1011 \def\bib@cite@check#1#2#3#4#5#6#7{%
1012     \ifx 1#4\relax
1013         \DuplicateBibKeyWarning
1014     \else
```

This has gotten *way* out of hand.

```
1015     \begingroup
1016         \auto@protect\etaltext
1017         \@apply\auto@protect\amsrefs@textsymbols
1018         \@apply\auto@protect\amsrefs@textaccents
1019         \@tempswafalse
1020         \in@CitePrintUndefined{#5}%
1021         \ifin@
1022             \let\@tempa\@empty
1023         \else
1024             \def\@tempa{#5}%
1025         \fi
1026         \ifx\@tempa\@empty
1027         \else
1028             \@xp\ifx\@xp\@currentlabel\cite@label
1029                 \edef\@tempb{\cite@label}%
1030             \else
1031                 \let\@tempb\cite@label
1032             \fi
1033             \ifx\@tempa\@tempb
1034                 \def\@tempa{#6}%
1035                 \ifx\@tempa\bib@label@year
1036                     \else
1037                         \@tempswatrue
1038                     \fi
1039                 \else
1040                     \@tempswatrue
1041                 \fi
1042             \fi
1043             \if@tempswa
1044                 \@ifempty{#6}{%
1045                     \def\@tempa{#5}%
```

```

1046          \let\@tempb\cite@label
1047      }{%
1048          \def\@tempa{#5, #6}%
1049          \def\@tempb{\cite@label, \bib@label@year}%
1050      }%
1051          \amsrefs@warning{Citation label for \extr@cite#1 is
1052              changing from '\@tempa ' to '\@tempb '}%
1053      \fi
1054      \protected@edef\@tempa{%
1055          \gdef\@nx#1{%
1056              \@nx\citesel #31{\cite@label}{\bib@label@year}{#7}%
1057          }%
1058      }%
1059      \@xp@endgroup
1060      \@tempa
1061  \fi
1062 }

```

\bib@label@year

```

1063 \let\bib@label@year\@empty

```

\DuplicateBibKeyWarning

```

1064 \def\DuplicateBibKeyWarning{%
1065     \amsrefs@warning{%
1066         Duplicate \protect\bib\space key
1067         '\CurrentBib ' detected\MessageBreakNS}%
1068 }

```

\bibcite@write

```

1069 \def\bibcite@write#1{%
1070     \if@filesw
1071         \let\citesel\citesel@write
1072         \csname b@#1\endcsname
1073     \fi
1074 }

```

\citesel@write

```

1075 \def\citesel@write#1#2#3#4#5{%
1076     \begingroup
1077         \toks@{#3}{#4}%
1078         \immediate\write\@auxout{\string\bibcite{\CurrentBib}{\the\toks@}}%
1079     \endgroup
1080 }

```

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

```

1081 \AtEndDocument{\let\bibcite\@gobbletwo}

```

6.12.8 Printing the bibliography

```
\bibname
1082 \providecommand{\bibname}{Bibliography}
```

```
\refname
1083 \providecommand{\refname}{References}
```

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

```
1084 \newenvironment{bibchapter}[1][\bibname]{%
1085     \begingroup
1086         \protected@edef\@{\endgroup\protect\chapter*{#1}}%
1087         \@
1088 }{\par}
```

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

```
1089 \newenvironment{bibsection}[1][\refname]{%
1090     \begingroup
1091         \protected@edef\@{\endgroup\protect\section*{#1}}%
1092         \@
1093 }{\par}
```

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

```
1094 \@ifundefined{chapter}{%
1095     \newenvironment{bibdiv}{\bibsection}{\endbibsection}
1096 }{%
1097     \newenvironment{bibdiv}{\bibchapter}{\endbibchapter}
1098 }
```

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

```
1099 \renewenvironment{thebibliography}[1]{%
1100     \bibdiv
1101     \biblist[\resetbiblist{#1}]%
1102 }{%
1103     \endbiblist
1104     \endbibdiv
1105 }
```

6.13 Name, journal and publisher abbreviations

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

`\DefineName`

```
1106 \newcommand{\DefineName}[2]{%
1107     \bib*{#1}{name}{name={#2}}%
1108 }
```

`\DefineJournal`

```
1109 \newcommand{\DefineJournal}[4]{%
1110     \bib*{#1}{periodical}{
1111         issn={#2},
1112         journal={#4}
1113     }%
1114 }
```

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

```
1115 \newcommand{\DefinePublisher}[4]{%
1116     \bib*{#1}{publisher}{%
1117         publisher={#3},
1118         address={#4}
1119     }%
1120 }
```

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`

```
1121 \newcommand{\bibselect}{%
1122     \@ifstar{%
1123         \let\@bibdef\copy@bibdef
1124         \BibSelect
1125     }{%
1126         \let\@bibdef\selective@bibdef
1127         \BibSelect
1128     }%
1129 }
```

`\BibSelect`

```
1130 \newcommand{\BibSelect}[2][\bblname]{%
1131     \if@filesw
```

```

1132      \typeout{Trying to create bbl file '#1.bbl' ...}%
1133      \def\bibselect@msg{%
1134          \typeout{ ... rats. Unable to create bbl file.}%
1135      }%
1136      \let\@open@bbl@file\OpenBBLFile
1137      \@for\@tempa:=#2\do{\ReadBibData{\@tempa}}%
1138  \fi
1139  \@close@bbl@file
1140  \@apply\g@undef\bibdefer@list
1141  \global\let\bibdefer@list\@empty

```

Now read the .bbl file we just created.

```

1142  \let\@bibdef\normal@bibdef
1143  \@input@{#1.bbl}%
1144  \let\BibSelect\MultipleBibSelectWarning
1145 }

```

\MultipleBibSelectWarning

```

1146 \newcommand\MultipleBibSelectWarning[2][{}]{%
1147     \amsrefs@warning{%
1148         Multiple \string\bibselect 's found (only one
1149         \string\bibselect\space per biblist environment is allowed)%
1150     }%
1151 }

```

\bblname

```

1152 \def\bblname{\jobname}

```

\bib@dbfile

```

1153 \newread\bib@dbfile

```

\ReadBibData

```

1154 \newcommand{\ReadBibData}[1]{%
1155     \IfFileExists{#1.ltb}{%
1156         \openin\bib@dbfile=\@filef@und \relax
1157     }{%
1158         \IfFileExists{#1.ltx}{%
1159             \openin\bib@dbfile=\@filef@und \relax
1160         }{%
1161             \IfFileExists{#1.tex}{%
1162                 \openin\bib@dbfile=\@filef@und \relax
1163             }{%
1164                 \begingroup
1165                 \NoBibDBFile{#1}%
1166                 \let\ReadBibData@a\endgroup
1167             }%
1168         }%
1169     }%
1170     \ReadBibData@a
1171 }

```

\NoBibDBFile

```
1172 \def\NoBibDBFile#1{%
1173     \amsrefs@warning{No data file #1.ltb (.ltx, .tex) found}%
1174 }
```

\ReadBibData@a

```
1175 \def\ReadBibData@a{%
1176     \ProvidesFile{\@filef@und}\relax
1177     \begingroup
1178         \let\star@bibdef\defer@bibdef
1179         \ReadBibLoop
1180     \endgroup
1181     \closein\bib@dbfile
1182 }
```

\ReadBibLoop

```
1183 \def\ReadBibLoop{%
1184     \ifeof\bib@dbfile
1185         \@xp\@gobble
1186     \else
1187         \read\bib@dbfile to\CurLine
1188         The \@empty is in case \CurLine is empty.
1189         \@xp\ReadBibLoop@a\CurLine\@empty\@nil
1190     \fi
1191 }
```

\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.

```
1192 \long\def\ReadBibLoop@a#1#2\@nil{%
1193     \ifx\bib#1%
1194         \CurLine % just exec it
1195     \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.

```
1196     \ifx\endinput#1%
1197         \let\ReadBibLoop\@empty
1198     \else
```

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

```
1199         \@xp\ReadBibLoop@b#1#2\@empty\@nil
1200     \fi
1201 \fi
1202 }
```

```

\ReadBibLoop@b
1203 \long\def\ReadBibLoop@b#1#2\@nil{%
1204     \ifx\bib#1%
1205         \CurLine % just exec it
1206     \fi
1207 }

1208 \let\bbl@out=\relax
1209 \let\bbl@write\@gobble
1210 \let\@open@bbl@file\relax
1211 \let\@close@bbl@file\relax

\OpenBBLFile
1212 \def\OpenBBLFile{%
1213     \if@files
1214         % Just use the next unused output stream
1215         \count@\count17
1216         \advance\count@\@ne
1217         \ifnum\count@<\sist@\@n
1218             \global\chardef\bbl@out=\count@
1219             \immediate\openout\bbl@out=\bblname.bbl\relax
1220             \global\let\@close@bbl@file\CloseBBLFile
1221             \gdef\bbl@write{\immediate\write\bbl@out}%
1222         \else
1223             \ch@ck\count@\sist@\@n\write
1224         \fi
1225     \fi
1226     \global\let\@open@bbl@file\relax
1227 }

\CloseBBLFile
1228 \def\CloseBBLFile{%
1229     \immediate\closeout\bbl@out\relax
1230     \global\let\@close@bbl@file\relax
1231     \global\let\bbl@write\@gobble
1232     \global\let\bbl@out\relax
1233 }

```

6.15 Citation processing

6.15.1 The \citesel structure

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

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

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

The second status flag is 1 if this key has already been used by a `define-cite` command (such as `\bib`); 0 otherwise. This makes it possible to issue a warning

message as soon as the conflict is seen, on the first L^AT_EX 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
1234 \def\citesel@update#1#2#3#4#5#6{%
1235   \gdef#6{\citesel 1#2{#3}{#4}{#5}}%
1236 }

\citesel@number
1237 \def\citesel@number#1#2#3#4#5{#3}

\citesel@year
1238 \def\citesel@year#1#2#3#4#5{#4}

\citesel
1239 \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.

```
1240 \expandafter\let\csname cite \endcsname\relax

\cite Need to handle the standard [...] option for compatibility's sake.
1241 \renewcommand{\cite}[2] [] {%
1242   \if\cite@single#2,\@gobble \else\MultipleCiteKeyWarning{#2}{#1}\fi
1243   \@ifempty{#1}{%
1244     \cites@o{#2}%
1245   }{%
1246     \ObsoleteCiteOptionWarning
1247     \cites@a{*{#1}}{#2}%
1248   }%
1249 }
```

`\MultipleCiteKeyWarning`

```

1250 \def\MultipleCiteKeyWarning#1#2{%
1251     \amsrefs@warning{%
1252         Use of \string\cites\space is recommended instead of %
1253         \string\cite\space\MessageBreak
1254         for multiple cites '#1'%
1255     \@ifnotempty{#2}{%
1256         \amsrefs@warning{Star option requires \string\citelist\space here}%
1257     }%
1258     \global\let\MultipleCiteKeyWarning\@gobbletwo
1259 }

```

`\ObsoleteCiteOptionWarning`

```

1260 \def\ObsoleteCiteOptionWarning{%
1261     \amsrefs@warning{%
1262         The form \string\cite{...}*{...} is recommended\MessageBreak
1263         instead of \string\cite[...]{...}%
1264     \global\let\ObsoleteCiteOptionWarning\@empty
1265 }

```

`\cite@single`

```

1266 \edef\cite@single#1,#2{\iffalse{\fi\iffalse{\fi\string}#2.\string}}

```

`\cites@o`

```

1267 \def\cites@o#1{\star@\cites@oo{#1}}{}

```

`\cites@oo`

```

1268 \def\cites@oo#1#2{\@ifempty{#2}{\cites@a{#1}}{\cites@a{*{#2}}{#1}}}

```

`\cites@a`

```

1269 \def\cites@a#1#2{%
1270     \begingroup
1271     \toks@{\endgroup \cites@b{#1}}%
1272     \vdef\@tempa{#2}%
1273     \edef\@tempa{%
1274         \the\toks@ \@firstofone{\@xp\zap@space\@tempa} \@empty
1275     }%
1276     \@tempa,\@empty
1277     \edef\@tempa{\endgroup\@nx\citelist{\the\toks@}}%
1278     \@tempa
1279 }

```

`\cites@b`

```

1280 \def\cites@b#1#2,#3{%
1281     \begingroup
1282     \toks@{\InnerCite{#2}#1}%
1283     \ifx\@empty#3\@xp\@gobble\fi
1284     \cites@c#3%
1285 }

```

`\cites@c`

```
1286 \def\cites@c#1,#2{%
1287     \add@toks@{\InnerCite{#1}}%
1288     \ifx\@empty#2\@xp\@gobble\fi
1289     \cites@c#2%
1290 }
```

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

```
1291 \def\citeleft{[]
1292 \def\citeright{[]
1293 \def\citemid{,\penalty9999 \space}
1294 \def\citepunct{,\penalty9999 \hskip.13em plus.1em minus.05em\relax}
```

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

```
1295 \def\citeAltPunct{;\ }
```

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

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

```
1296 \providecommand{\citeform}{\@firstofone}
```

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

```
1297 \DeclareRobustCommand{\citelist}{\@citelist}
```

`\@citelist`

```
1298 \def\@citelist#1{%
1299     \leavevmode
1300     \begingroup
1301         \@citestyle
1302         \citeleft\nopunct    % suppress first \citepunct
1303         \cite@begingroup
1304         \in@*{#1}%
1305         \ifin@
1306             \let\citepunct\CiteAltPunct
1307         \fi
1308         \let\cite@endgroup\@empty
1309         \cites@init
1310         \def\citeleft{\@addpunct{\citepunct}}%
1311         \let\citeright\ignorespaces
```

```

1312         \def\cite{\InnerCite}%
1313         \process@citelist{#1}%
1314     \endgroup
1315     \citeright
1316 \endgroup
1317 }

```

`\@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).

```

1318 \providecommand{\@citestyle}{\m@th\upshape\mdseries}

```

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

```

1319 \def\cite@begingroup{\begingroup\let\cite@begingroup\relax}

```

`\cite@endgroup`

```

1320 \let\cite@endgroup\endgroup

```

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

```

1321 \def\cites@init{%
1322     \gdef\prev@names{???}%
1323     \let\cites@init\@empty
1324 }

```

`\InnerCite`

```

1325 \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`.

```

1326 \def\cite@a#1#2{%
1327     \BackCite{#2}%
1328     \cite@begingroup
1329         \cites@init
1330         \let\citesel#1\relax
1331         \ifx\citesel\citesel@author
1332             \let\citeleft\@empty
1333             \let\citeright\@empty
1334         \fi
1335         \begingroup
1336             \toks@{\endgroup \cite@b}%
1337             \vdef\@tempa{#2}%
1338             \edef\@tempa{%
1339                 \the\toks@{\@firstofone{\@xp\zap@space\@tempa} \@empty}%
1340             }%
1341         \@tempa
1342 }

```

`\cite@b` *Arguments:*

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

```
1343 \def\cite@b#1#2{%
1344   \@xp\cite@bc\csname b@#1\@xp\endcsname {#1}{#2}%
1345 }
```

`\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`.

```
1346 \def\cite@bc#1#2{%
1347   \ifx#1\@undefined \global\let#1\relax \fi
1348   \ifx#1\relax      \global\let#1\empty@cite \fi
1349   \@xp\cite@nobib@test#1{}{}{}{}\@nil#1%
1350   \cite@cj#1%
1351 }
```

`\empty@cite`

```
1352 \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.`

```
1353 \def\cite@nobib@test#1#2#3#4#5\@nil#6{%
1354   \@ifempty{#4}{%
1355     \G@refundefinedtrue
1356     \UndefinedCiteWarning#6%
1357     \xdef#6{\@nx\citesel #2#3{%
1358       \@nx\CitePrintUndefined{\extr@cite#6}}{}{}}%
1359   }{}%
1360 }
```

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

```
1361 \def\UndefinedCiteWarning#1{%
1362   \@latex@warning{%
1363     Citation '\extr@cite#1' on page \thepage\space undefined}%
1364 }
```

`\CitePrintUndefined`

```
1365 \DeclareRobustCommand{\CitePrintUndefined}[1]{%
1366   \begingroup\fontshape{n}\fontseries\mddefault \ttfamily ?#1\endgroup
1367 }
```

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

```
1368 \let\CPU@normal\CitePrintUndefined
```

`\cite@cj` *Arguments:*

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

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

```
1369 \def\cite@cj#1#2{%
1370     \leavevmode
1371     \begingroup
1372         \cite@cb#1% write info to aux file
1373         \ar@SK@cite#1%
1374         \@citeleft
1375         \ar@hyperlink{#1}%
1376         \@ifnotempty{#2}{\citimid{#2}}%
1377         \citeright
1378     \endgroup
1379     \ignorespaces % ignore spaces inside \citelist
1380 \cite@endgroup
1381 }
```

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

```
1382 \def\@citeleft{\citeleft}
```

`\cite@cb`

```
1383 \def\cite@cb#1{%
1384     \if@filesw
1385         \immediate\write\@auxout{\string\citation{\extr@cite#1}}%
1386     \fi
    Define \citesel to make \b@whatever update itself.
1387     \begingroup
1388         \let\citesel\citesel@update
1389         #1#1%
1390     \endgroup
1391 }
```

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

```
1392 \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.)

```
1393 \DeclareRobustCommand{\cites}{\cites@a{}}
```

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

```
1394 \providecommand{\citen}{\ocite}
```

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

```

1395 \def\ycite{\cite}

\ycites
1396 \let\ycites\cites

\ocite
1397 \let\ocite\ycite

\ocites
1398 \let\ocites\cites

\fullcite
1399 \let\fullcite\cite

\fullocite
1400 \let\fullocite\ocite

\citeauthor
1401 \let\citeauthor\ycite

\citeauthority
1402 \let\citeauthority\ycite

6.15.4 The \nocite command

\nocite
1403 \renewcommand{\nocite}[1]{\othercites{#1}}

\othercites
1404 \newcommand{\othercites}[1]{%
1405     \cite@begingroup
1406     \let\cite@endgroup\@empty
1407     \def\citelist{\othercitelist}%
1408     \cites{#1}%
1409 }

\othercitelist
1410 \newcommand{\othercitelist}[1]{%
1411     \cite@begingroup
1412     \let\cite@endgroup\@empty
1413     \cites@init
1414     \let\citeleft\relax
1415     \let\citeright\ignorespaces
1416     \def\InnerCite{\OtherCite}%
1417     \def\cite@cj ##1##2{%
1418         \begingroup
1419         \@xp\citesel##1%
1420         \cite@cb ##1%
1421         \endgroup

```

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

```

1422         \@xp\ifx\csname b@*\endcsname ##1%
1423         \global\let\selective@bibdef\copy@bibdef
1424         \fi
1425         \ignorespaces
1426         \cite@endgroup
1427     }%
1428     #1\relax
1429     \endgroup
1430 }
```

`\OtherCite`

```

1431 \def\OtherCite#1{\cite@a\citesel@other{#1}{}}}
```

`\citesel@other`

```

1432 \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.

```

1433 \@namedef{b@*}{\citesel 11{*}{*}{*}}
```

6.15.5 Citation sorting

`\process@citelist@sorted`

```

1434 \def\process@citelist@sorted#1{%
1435     \ifx\citesel\citesel@number
1436         \cite@sorted@s #1\cite@sorted@e
1437     \else
1438         \NonNumericCiteWarning
1439         \process@citelist@unsorted{#1}%
1440     \fi
1441 }
```

`\NonNumericCiteWarning`

```

1442 \def\NonNumericCiteWarning{%
1443     \amsrefs@warning{%
1444         Unable to confirm that cite keys are numeric: not sorting%
1445     }%
1446 }
```

`\process@citelist@unsorted`

```

1447 \def\process@citelist@unsorted#1{%
1448     \ignorespaces#1\relax
1449 }
```

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

```

1450 \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.

```
1451 \def\CPU@sort#1{2147483647}
```

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

```
1452 \def\cite@sorted@s{%
1453   \begingroup
1454     \let\CitePrintUndefined\CPU@sort
1455     \let\cite@cjs\cite@cj
1456     \let\cite@cj\cite@compress
1457     \begingroup
1458       \toks@{\emptytoks}
1459       \let\cite@cj\cite@sort
1460       \ignorespaces
1461 }
```

`\cite@sorted@e`

```
1462 \def\cite@sorted@e{%
1463   \@xp\endgroup
1464   \the\toks@
1465   \cite@dash
1466   \prev@cite
1467   \endgroup
1468 }
```

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

Arguments:

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

`#2` <- *optional arg*.

```
1469 \def\cite@sort#1#2{%
1470   \safe@set\@tempcnta#1% highest number so far
1471   \toks@{\cite@cj#1{#2}}%
1472   \@temptokena\toks@
1473   \let\cite@cj\cite@sort@a
1474   \ignorespaces
1475 }
```

`\cite@sort@a`

```
1476 \def\cite@sort@a#1#2{%
1477   \safe@set\@tempcntb#1%
1478   \ifnum\@tempcntb > \@tempcnta
1479     \add@toks@{\cite@cj#1{#2}}%
1480     \@tempcnta\@tempcntb
1481   \else
1482     \let\cite@cj\cite@sort@b
1483     \toks@{\emptytoks}
1484     \def\@tempb{\add@toks@{\cite@cj#1{#2}}}%
1485     \the\@temptokena
```

```

1486      \@tempb
1487      \let\cite@cj\cite@sort@a
1488      \fi
1489      \@temptokena\toks@
1490      \ignorespaces
1491  }

\cite@sort@b

1492 \def\cite@sort@b#1#2{%
1493   \safe@set\count@#1%
1494   \ifnum\@tempcntb < \count@
1495     \@tempb
1496     \let\@tempb\@empty
1497   \fi
1498   \add@toks@\{\cite@cj#1{#2}}%
1499   \ignorespaces
1500 }
```

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`

```
1501 \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).

```
1502 \def\prev@cite@cb{%
1503     \ifx\@prev@cite\@empty
1504     \else
1505         \begingroup
1506             \def\cite@print##1##2{%
1507                 \cite@cb##1%
1508             }%
1509             \prev@cite
1510         \endgroup
1511     \fi
1512 }
```

`\cite@print`

```
1513 \def\cite@print#1#2{%
1514     \begingroup
1515         \let\CitePrintUndefined\CPU@normal
1516         \cite@cjs#1{#2}%
1517     \endgroup
1518 }
```

`\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.

```
1519 \let\cite@dash\@empty
```

```
\print@one@dash
```

```
1520 \def\print@one@dash{%
1521   \textendash \nopunct
1522   \let\cite@dash\@empty
1523 }
```

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:
```

```
1524 \def\cite@compress#1#2{%
1525   \cite@print#1{#2}%
1526   \@ifempty{#2}{%
1527     \safe@set\@tempcnta#1%
1528     \let\cite@cj\cite@compress@a
1529   }{}%
1530 }
```

```
\cite@compress@a State 1:
```

```
1531 \def\cite@compress@a#1#2{%
1532   \@ifempty{#2}{%
1533     \advance\@tempcnta\@ne
1534     \safe@set\@tempcntb#1%
1535     \ifnum\@tempcnta=\@tempcntb
1536       \def\prev@cite{\cite@print#1{}}%
1537       \let\cite@cj\cite@compress@b
1538     \else
1539       \cite@print#1{}%
1540       \@tempcnta\@tempcntb
1541     \fi
1542   }{}%
1543   \cite@print#1{#2}%
1544   \let\cite@cj\cite@compress
1545   }%
1546 }
```

```
\cite@compress@b State 2:
```

```
1547 \def\cite@compress@b#1#2{%
1548   \@ifempty{#2}{%
1549     \advance\@tempcnta\@ne
1550     \safe@set\@tempcntb#1%
1551     \ifnum\@tempcnta=\@tempcntb
```

```

1552         \let\cite@dash\print@one@dash
1553         \prev@cite@cb
1554         \def\prev@cite{\cite@print#1{}}%
1555     \else
1556         \cite@dash
1557         \prev@cite
1558         \let\prev@cite\@empty
1559         \cite@print#1{}%
1560         \@tempcnta\@tempcntb
1561         \let\cite@cj\cite@compress@a
1562     \fi
1563 }{%
1564     \cite@dash
1565     \prev@cite
1566     \let\prev@cite\@empty
1567     \cite@print#1{#2}%
1568     \let\cite@cj\cite@compress
1569 }%
1570 }

```

6.15.7 Munging the .aux file

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

```

1571 \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.

```

1572 \AtBeginDocument{%
1573     \if@filesw
1574         \immediate\write\@auxout{%
1575             \string\@ifundefined{amsrefs@bibcite}{}{}%
1576             \string\let\string\bibcite\string\amsrefs@bibcite
1577         }%
1578     }%
1579 \fi

```

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

```

1580     \let\bibcite\amsrefs@bibcite
1581 }

```

`\bibcite@a` *Arguments:*

```

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

```

```

1582 \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.

```

1583     \ifx\relax#1%
1584         \gdef#1{\citesel 00#2{}}%
1585     \else
1586         \begingroup
1587             \xp\bibcite@b\@xp#1#1{#2}%
1588         \endgroup
1589     \fi
1590 }

\bibcite@b Arguments:
    #1 <- \b@citekey.
    #2 <- \citesel.
    #3 <- cited?.
    #4 <- used?.
    #5 <- label.
    #6 <- year.
    #7 <- backrefs.
    #8 <- {newlabel}{newyear}.
1591 \def\bibcite@b#1#2#3#4#5#6#7#8{\gdef#1{\citesel#3#4#8{#7}}}}

\citedest The \citedest command goes into the .aux file to provide back-reference sup-
port.
1592 \newcommand{\citedest}[1]{\@xp\cite@dest\csname b@#1\endcsname}

\cite@dest
1593 \def\cite@dest#1{%
1594     \ifx\relax#1%
1595         \gdef#1{\citesel 00{}{}{}}%
1596     \fi
1597     \xp\cite@dest@b\@xp#1#1%
1598 }

\cite@dest@b Arguments:
    #1 <- \b@citekey.
    #2 <- \citesel.
    #3 <- cited?.
    #4 <- used?.
    #5 <- label.
    #6 <- year.
    #7 <- backrefs.
    #8 <- {more backrefs}.
1599 \def\cite@dest@b#1#2#3#4#5#6#7#8{%
1600     \@ifempty{#7}{%
1601         \def#1{\citesel #3#4{#5}{#6}{#8}}%
1602     }{%
1603         \gdef#1{\citesel #3#4{#5}{#6}{#7,#8}}%
1604     }%
1605 }

```

6.15.8 Back references`\ifBR@verbose`

```
1606 \@ifundefined{ifBR@verbose}{\let\ifBR@verbose\iffalse \let\fi\fi}{}%
```

`\BackCite`

```
1607 \let\BackCite\@gobble
```

`\back@cite`

```
1608 \def\back@cite#1{%
1609   \ifBR@verbose
1610     \PackageInfo{backref}{back cite \string '\extr@cite#1'}%
1611     \fi
1612     \Hy@backout{#1}%
1613 }
```

`\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.

```
1614 \def\print@backrefs#1{%
1615   \space\SentenceSpace$\uparrow$\csname br@#1\endcsname
1616 }
```

`\PrintBackRefs`

```
1617 \let\PrintBackRefs\@gobble
```

6.15.9 hyperref, showkeys and shaderef support`\shade@cite`

```
1618 \newcommand{\shade@cite}{\overshadebox{shaderef-gray}}
```

`\ar@hyperlink`

```
1619 \def\ar@hyperlink#1{%
1620   \hyper@link [cite]{}{cite.\extr@cite#1}{\shade@cite{#1}}%
1621 }
```

`\ar@SK@cite`

```
1622 \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.

```
1623 \AtBeginDocument{%
1624   \@ifpackageloaded{shaderef}{}{%
1625     \let\shade@cite\empty
1626   }%
1627   \@ifpackageloaded{hyperref}{}{%
1628     \let\ar@hyperlink\shade@cite
1629     \let\hyper@anchorstart\@gobble
1630     \let\hyper@anchorend\relax
```

```

1631 }%
1632 \@ifpackageloaded{showlabels}{\{%
1633     \let\ar@SK@cite@gobble
1634     \let\SK@@label@gobble
1635     \let\SK@@gobbletwo
1636 }%
1637 }

```

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 `~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 83).³

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

`\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 \'
```

```
1638 \let\amsrefs@textaccents\@empty
```

`\DeclareNameAccent` *Arguments:*

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

```
1639 \def\DeclareNameAccent{%
1640     \@lappend\amsrefs@textaccents
1641 }
```

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

```
1642 \DeclareNameAccent\"
1643 \DeclareNameAccent\'
1644 \DeclareNameAccent\
1645 \DeclareNameAccent=
1646 \DeclareNameAccent^
1647 \DeclareNameAccent`
1648 \DeclareNameAccent~%
1649 \DeclareNameAccent\b
1650 \DeclareNameAccent\c
1651 \DeclareNameAccent\d
1652 \DeclareNameAccent\H
1653 \DeclareNameAccent\k
1654 \DeclareNameAccent\r
1655 \DeclareNameAccent\t
1656 \DeclareNameAccent\u
1657 \DeclareNameAccent\v
```

From `mathscinet`:

```
1658 \DeclareNameAccent\utilde
1659 \DeclareNameAccent\uarc
1660 \DeclareNameAccent\dudot
1661 \DeclareNameAccent\lfhook
1662 \DeclareNameAccent\udot
1663 \DeclareNameAccent\polhk
1664 \DeclareNameAccent\soft
```

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

```
1665 \DeclareNameAccent\etalchar
1666 %\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`.

```
1667 \let\amsrefs@textsymbols\@empty
```

`\DeclareNameSymbol` *Arguments:*

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

```
1668 \def\DeclareNameSymbol#1#2{%
1669   \@lappend\amsrefs@textsymbols#1%
1670   \@lappend\amsrefs@textsymbols#2%
1671   \ifx#1#2\else
1672     \@lappend\amsrefs@textsymbols#2%
1673     \@lappend\amsrefs@textsymbols#2%
1674   \fi
1675 }
```

Here are the standard L^AT_EX and `mathscinet` text symbols.

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

```
1676 \DeclareNameSymbol\i\i
1677 \DeclareNameSymbol\j\j
1678 \DeclareNameSymbol\AE\ae
1679 \DeclareNameSymbol\OE\oe
1680 \DeclareNameSymbol\O\o
1681 \DeclareNameSymbol\DH\dh
```

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

```

1682 \DeclareNameSymbol\DJ\dj
1683 \DeclareNameSymbol\L\l
1684 \DeclareNameSymbol\NG\ng
1685 \DeclareNameSymbol\SS\ss
1686 \DeclareNameSymbol\TH\th

```

From mathscinet:

```

1687 \DeclareNameSymbol\Dbar\dbar
1688 \DeclareNameSymbol\lasp\lasp
1689 \DeclareNameSymbol\rasp\rasp
1690 \DeclareNameSymbol\cprime\cprime
1691 \DeclareNameSymbol\cdprime\cdprime
1692 \DeclareNameSymbol\bud\bud
1693 \DeclareNameSymbol\cydot\cydot

```

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

```
1694 \DeclareNameSymbol~~%
```

6.16.3 \edef-like macros for names

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

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

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

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

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

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

```

1695 \def\normalize@edef#1#2{%
1696     \begingroup
1697         \@apply\auto@protect\amsrefs@textsymbols
1698         \@apply\wrap@accent\amsrefs@textaccents

```

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

```

1699     \let\@tabacckludge\use@accent
1700     \let~\space
1701     \edef\@tempa{\def\@nx#1{#2}}%
1702     \@xp\endgroup
1703     \@tempa
1704 }

```

`\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

```
1705 \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}`.

```
1706 \def\wrap@accent#1{%
1707     \def##1{\@nx#1##1}%
1708 }
```

`\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.

```
1709 \def\lc@edef#1#2{%
1710     \begingroup
1711     \let\@tabacckludge\use@accent %%??
1712     \@apply\auto@protect\amsrefs@textaccents
1713     \@apply\lc@do\amsrefs@textsymbols
1714     \edef\@tempa{\lowercase{\def\@nx#1{#2}}}%
1715     \@xp@endgroup
1716     \@tempa
1717 }
```

`\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.

```
1718 \def\lc@do#1\do#2{%
1719     \ifx#1#2%
1720         \auto@protect#1%
1721     \else
1722         \def#1{#2}%
1723     \fi
1724 }
```

`\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`.

```
1725 \def\purge@edef#1#2{%
```

```

1726 \begingroup
1727 \apply\auto@protect\amsrefs@textsymbols
1728 \let~\space
1729 \apply\purge@accent\amsrefs@textaccents
1730 \let\@tabacckludge\@gobble

```

As mentioned above (page 67), \i and \j are semantically like text accents; hence, they require special treatment here.

```

1731 \def\i{i}%
1732 \def\j{j}%
1733 \edef\@tempa{#2}%
1734 \toks@{\emptytoks}
1735 \exp\purge@edef@ \@tempa \@nil
1736 \edef\@tempa{\def\@nx#1{\the\toks@}}%
1737 \exp\endgroup
1738 \@tempa
1739 }

```

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

```

1740 \def\purge@edef@{%
1741 \futurelet\@let@token
1742 \purge@edef@a
1743 }

```

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

```

1744 \def\purge@edef@a#1{%
    If we’ve run into the \@nil terminator, we’re done.
1745 \ifx\@let@token\@nil
1746 \let\@tempa\@empty
1747 \else

```

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

```

1748 \ifx\@let@token\bgroup
1749 \def\@tempa{%
1750 \purge@edef@ #1\@nil
1751 \purge@edef@
1752 }%
1753 \else

```

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

```

1754 \add@toks@{#1}%
1755 \let\@tempa\purge@edef@
1756 \fi
1757 \fi
1758 \@tempa
1759 }

```

`\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).

```
1760 \def\purge@accent#1{%
1761     \def#1##1{##1}%
1762 }
```

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 65), 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.)

```
1763 \def\name@split{%
1764   \get@namepart\bib'surname\name@split@given
1765 }
```

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

```
1766 \def\name@split@given{%
1767   \get@namepart\bib'given\name@split@jr
1768 }
```

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

```
1769 \def\name@split@jr{%
1770   \get@namepart\bib'jr\name@split@finish
1771 }
```

`\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.

```
1772 \def\name@split@finish{%
1773     \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.

```
1774 %%         \ifx\@empty\bib'given
1775 %%             \NameCheck \bib'surname ??\@nil
1776 %%         \else
1777             \extract@initials\bib'given
1778 %%         \fi
1779     \@nilgobble
1780 }
```

`\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.

```
1781 \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.

```
1782     \toks@{#1}%
1783     \@temptokena{#2}%
1784     \get@namepart@a
1785 }
```

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

```
1786 \def\get@namepart@a{%
1787     \futurelet\@let@token
1788     \get@namepart@b
1789 }
```

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

```
1790 \def\get@namepart@b{%
```

```

1791 \ifx\@let@token\@sptoken
1792 \xp\get@namepart@c
1793 \else
1794 \xp\get@namepart@d
1795 \fi
1796 }

```

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

```

1797 \def\get@namepart@c{%
1798 \after@deleting@token\get@namepart@a
1799 }

```

`\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.

```

1800 \def\get@namepart@d{%
1801 \futurelet\@let@token
1802 \get@namepart@e
1803 }

```

`\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.

```

1804 \def\get@namepart@e{%
1805 \ifx\@let@token,%
1806 \xp\let\the\toks@\@empty
1807 \edef\@tempa{%
1808 \nx\after@deleting@token\the\@temptokena
1809 }%
1810 \xp\@tempa
1811 \else
1812 \xp\get@namepart@f
1813 \fi
1814 }

```

`\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.

```

1815 \def\get@namepart@f#1#2,{%
1816 \ifx\@let@token\bgroup

```

⁵More or less. If the second argument is brace-delimited, the braces will be lost. But as mentioned above (page 72), we don't really care.

```

1817      \@xp\def\the\toks@{{#1}#2}%
1818      \else
1819      \@xp\def\the\toks@{{#1}#2}%
1820      \fi
1821      \the\@temptokena
1822 }

```

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

```

1823 \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.

```

1824 \def\extract@initials#1{%
1825     \begingroup
1826         \auto@protect\ini@hyphen
1827         \auto@protect\nobreakspace
1828         \let~\relax
1829         \@apply\auto@protect\amsrefs@textsymbols
1830         \@apply\auto@protect\amsrefs@textaccents
1831         \normalize@edef\@tempa{#1}%

```

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

```

1832      \ifx\@tempa\@empty
1833      \else

```

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

```

1834      \process@hyphens\@tempa
1835      \process@dots\@tempa
1836      \process@names\@tempa
1837      \@chomp\@tempa{~}%
1838      \fi
1839      \edef\@tempa{\def\@nx\bib’initials{\@tempa}}%
1840      \@xp\endgroup
1841      \@tempa
1842 }

```

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

```

1843 \def\ini@hyphen{\unskip-\nobreak}

```

`\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.

```

1844 \def\process@hyphens#1{%
1845     \begingroup
1846     \toks@\@emptytoks
1847     \@xp\process@one@hyphen #1-\@nil
1848     \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.)

```

1849     \@chomp\@tempa{ \ini@hyphen\space}%
1850     \edef\@tempa{\def\@nx#1{\@tempa}}%
1851     \@xp\endgroup
1852     \@tempa
1853 }

```

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

```

1854 \def\process@one@hyphen{%
1855     \futurelet\@let@token
1856     \process@one@hyphen@a
1857 }

```

`\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.

```

1858 \def\process@one@hyphen@a{%
1859   \ifx\@let@token\@nil
1860     \let\fsa@n\@gobble
1861   \else
1862     \ifx\@let@token -%
1863       \TrailingHyphenWarning
1864       \let\fsa@n\process@one@hyphen@b
1865     \else
1866       \ifx\@let@token\@sptoken
1867         \let\fsa@n\process@one@hyphen@b
1868       \else
1869         \let\fsa@n\process@one@hyphen@c
1870       \fi
1871     \fi
1872   \fi
1873   \fsa@n
1874 }
```

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

```

1875 \def\process@one@hyphen@b{%
1876   \after@deleting@token\process@one@hyphen
1877 }
```

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

```

1878 \def\process@one@hyphen@c#1#2-{%
1879   \ifx\bgroup\@let@token
1880     \add@toks@{#1}#2 \ini@hyphen\space}%
1881   \else
1882     \add@toks@{#1#2 \ini@hyphen\space}%
1883   \fi
1884   \futurelet\@let@token
1885   \process@one@hyphen@d
1886 }
```

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

```

1887 \def\process@one@hyphen@d{%
1888   \ifx\@let@token\@nil
1889     \@xp\@gobble
```

```

1890     \else
1891         \@xp\process@one@hyphen
1892     \fi
1893 }

```

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

```

1894 \def\TrailingHyphenWarning{%
1895     \amsrefs@warning{Trailing hyphen deleted from name}%
1896 }

```

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

```

1897 \def\process@dots#1{%
1898     \begingroup
1899     \toks@\@emptytoks
1900     \@xp\process@one@dot #1.\@nil
1901     \edef\@tempa{\the\toks@}%
1902     \@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.

```

1903     \@chomp\@tempa{ }%
1904     \edef\@tempa{\def\@nx#1{\@tempa}}%
1905     \@xp\endgroup
1906     \@tempa
1907 }

```

`\process@one@dot`

```

1908 \def\process@one@dot{%
1909     \futurelet\@let@token
1910     \process@one@dot@a
1911 }

```

`\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`.

```

1912 \def\process@one@dot@a{%
1913     \ifx\@let@token .%
1914         \def\fsa@n{\after@deleting@token\process@bare@dot}%
1915     \else
1916         \ifx\@let@token\@sptoken
1917             \let\fsa@n\process@one@dot@b
1918         \else
1919             \let\fsa@n\process@one@dot@c
1920         \fi
1921     \fi
1922     \fsa@n
1923 }

```

`\process@bare@dot`

```

1924 \def\process@bare@dot{%
1925   \add@toks@{. }%
1926   \futurelet\@let@token
1927   \process@one@dot@d
1928 }

```

\process@one@dot@b

```

1929 \def\process@one@dot@b{%
1930   \after@deleting@token\process@one@dot
1931 }

```

\process@one@dot@c

```

1932 \def\process@one@dot@c#1#2.{%
1933   \ifx\bgroup\@let@token
1934     \add@toks@{{#1}#2. }%
1935   \else
1936     \add@toks@{#1#2. }%
1937   \fi
1938   \futurelet\@let@token
1939   \process@one@dot@d
1940 }

```

\process@one@dot@d

```

1941 \def\process@one@dot@d{%
1942   \ifx\@let@token\@nil
1943     \@xp\@gobble
1944   \else
1945     \@xp\process@one@dot
1946   \fi
1947 }

```

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

```

1948 \def\process@names#1{%
1949   \begingroup
1950     \toks@\@emptytoks
1951     \@xp\extract@initial #1 \@nil
1952     \edef\@tempa{\def\@nx#1{the\toks@}}%
1953   \@xp\endgroup
1954   \@tempa
1955 }

```

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

```

1956 \def\extract@initial{%
1957   \futurelet\@let@token
1958   \extract@initial@a
1959 }

```

`\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.

```

1960 \def\extract@initial@a{%
1961     \ifx\@let@token\@nil
1962         \let\fsa@n\@gobble
1963     \else
1964         \ifx\@let@token\@sptoken
1965             \let\fsa@n\extract@initial@b
1966         \else
1967             \let\fsa@n\extract@initial@c
1968         \fi
1969     \fi
1970     \fsa@n
1971 }
```

`\extract@initial@b`

```

1972 \def\extract@initial@b{%
1973     \after@deleting@token\extract@initial
1974 }
```

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

```

1975 \def\extract@initial@c#1#2 {%
1976     \ifx\@let@token\bgroup
```

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

```

1977         \@extract@initial {{#1}}#2\@nil
1978     \else
1979         \@extract@initial #1#2\@nil
1980     \fi
1981     \futurelet\@let@token
1982     \extract@initial@d
1983 }
```

`\extract@initial@d`

```

1984 \def\extract@initial@d{%
1985     \ifx\@let@token\@nil
1986         \@xp\@gobble
1987     \else
1988         \@xp\extract@initial
1989     \fi
1990 }
```

`\@extract@initial` This handles the four cases mentioned on page 76.

```

1991 \def\@extract@initial#1#2\@nil{%
1992     \ifx\ini@hyphen#1%
1993         \add@toks@{\ini@hyphen}%

```

```

1994   \else
1995       \in@{.\@nil}{#1#2\@nil}% Look for a period at the end of the name
1996       \ifin@
1997           \add@toks@{#1#2~}%
1998       \else
1999           \count@chars\@tempcnta{#1#2}%
2000           \ifnum\@tempcnta > \@ne
2001               \add@toks@{#1.~}%
2002           \else
2003               \add@toks@{#1~}%
2004           \fi
2005       \fi
2006   \fi
2007 }

```

\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.

```

2008 \def\count@chars#1#2{%
2009     \begingroup
2010         \@tempcnta\z@
2011         \@count@chars#2\@nil
2012         \edef\@tempb{#1=\the\@tempcnta\relax}%
2013     \exp\endgroup
2014     \@tempb
2015 }

```

\@count@chars

```

2016 \def\@count@chars#1{%
2017     \ifx #1\@nil
2018     \else
2019         \advance\@tempcnta\@ne
2020         \exp\@count@chars
2021     \fi
2022 }

```

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`, 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

```
2023 \let\previous@stem\@empty
```

```
2024 \let\current@stem\@empty
```

```
2025 \let\previous@year\@empty
```

```
2026 \let\current@year\@empty
```

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

```
2027 \def\append@to@stem{\global\@concat\current@stem}
```

```
\generate@alphalabel
```

```
2028 \def\generate@alphalabel{%
```

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

```
2029 \ifx\bib'label\@empty
```

```
2030 \begingroup
```

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

```
2031 \global\let\previous@stem\current@stem
```

```
2032 \global\let\current@stem\@empty
```

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

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

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

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.

```

2033      \@apply\auto@protect\amsrefs@textsymbols
2034      \@apply\auto@protect\amsrefs@textaccents
2035      \auto@protect\name
2036      \auto@protect\etaltext
2037      \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.

```

2038      \get@numberof\@tempcnta\name\@tempa
2039      \calc@author@part

```

Next append the year part.

```

2040      \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.

```

2041      \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.)

```

2042      \edef\@tempa{%
2043          \def\@nx\bib'label{%
2044              \current@stem
2045              \alpha@label@suffix
2046          }%
2047      }%
2048      \@xp\endgroup
2049      \@tempa
2050      \fi
2051  }

```

`\calc@author@part`

```

2052 \def\calc@author@part{%
2053     \ifnum \@tempcnta = 1
2054         \@xp\@oneauthorlabel\@xp{\@tempa}%
2055     \else
2056         \@xp\@multiauthorlabel\@xp{\@tempa}%
2057     \fi
2058 }

```

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

```

2059 \def\@firstone#1{\@car#1\@empty\@nil}

```

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

```

2060 \def\@firstthree#1{\@carcube#1\@empty\@empty\@empty\@nil}

```

\@nametoken

2061 \let\@nametoken\@firstone

\hyph@to@space

2062 \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).

2063 \def\@marknames#1{%

2064 \ifnotempty{#1}{\surround@names#1 ' }%

2065 }

\surround@names

2066 \def\surround@names#1 {%

2067 \ifx '#1%

2068 \else

2069 \@nx\@nametoken{#1}%

2070 \@xp\surround@names

2071 \fi

2072 }

\extract@surnames

2073 \def\extract@surnames#1#2{%

2074 \get@namepart\@tempb\@nilgobble #2,\@nil

2075 \edef\@tempb{\@nx\@marknames{\@xp\hyph@to@space\@tempb\@gobble-}}%

2076 \edef#1{\@tempb}%

2077 }

\@oneauthorlabel This is the easy case.

2078 \newcommand{\@oneauthorlabel}[1]{%

2079 \def\name##1{%

2080 \extract@surnames\@tempa{##1}%

2081 \get@numberof\@tempcnta\@nametoken\@tempa

2082 \ifnum \@tempcnta = 1

2083 \let\@nametoken\@firstthree

2084 \fi

2085 \append@to@stem{\@tempa}%

2086 }%

2087 #1%

2088 }

\@threeauthors

2089 \def\@threeauthors\name#1\name#2\name#3#4\@empty{%

2090 \name{#1}\name{#2}\name{#3}%

2091 \append@to@stem{\etalchar{+}}%

2092 }

\@multiauthorlabel

```

2093 \newcommand{\@multiauthorlabel}[1]{%
2094     \def\name##1{%
2095         \ifx\etaltext ##1%
2096             \def\@tempa{\@nx\etalchar{+}}%
2097             \let\name\@gobble
2098         \else
2099             \extract@surnames\@tempa{##1}%
2100         \fi
2101         \append@to@stem{\@tempa}%
2102     }%
2103     \ifnum \@tempcnta > 4 \exp \@threeauthors \fi
2104     #1\@empty
2105 }

```

\etalchar

```

2106 \newcommand{\etalchar}[1]{${\#1}$}

```

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

```

2107 \def\year@short#1#2#3#4\@nil{#3#4}

```

\append@label@year

```

2108 \def\append@label@year{%
2109     \safe@set\@tempcnta\bib@year
2110     \edef\bib@citeyear{\the\@tempcnta}%
2111     \append@to@stem{%
2112         \ifx\bib@year\@empty
2113         \else
2114             \exp\year@short \bib@citeyear \@nil
2115         \fi
2116     }%
2117 }

2118 \let\alpha@label@suffix\@empty
2119
2120 \newcount\alpha@suffix
2121 \alpha@suffix\@ne
2122 \let\@suffix@format\@alph

```

\calc@alpha@suffix

```

2123 \def\calc@alpha@suffix{%
2124     \@tempswafalse
2125     \compare@stems\previous@stem\current@stem
2126     \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.

```

2127     \ifx\previous@year\current@year
2128         \@tempswatrue
2129     \fi
2130 \fi
2131 \if@tempswa
2132     \global\advance\alpha@suffix\@ne
2133     \edef\alpha@label@suffix{\@suffix@format\alpha@suffix}%
2134     \ifnum\alpha@suffix=\tw@
2135         \immediate\write\@auxout{%
2136             \string\ModifyBibLabel{\prev@citekey}%
2137         }%
2138     \fi
2139 \else
2140     \let\alpha@label@suffix\@empty
2141     \global\alpha@suffix\@ne
2142     \@xp\ifx \csname b@\current@citekey @suffix\endcsname \relax
2143     \else
2144         \edef\alpha@label@suffix{\@suffix@format\alpha@suffix}%
2145     \fi
2146 \fi
2147 }

\ifsame@stems
2148 \newif\ifsame@stems

\compare@stems
2149 \def\compare@stems#1#2{%
2150     \begingroup
2151         \purge@edef\@tempa{#1}%
2152         \purge@edef\@tempb{#2}%
2153         \lc@edef\@tempa{\@tempa}%
2154         \lc@edef\@tempb{\@tempb}%
2155         \ifx\@tempa\@tempb
2156             \def\@tempa{\same@stemstrue}%
2157         \else
2158             \def\@tempa{\same@stemsfalse}%
2159         \fi
2160     \@xp\endgroup
2161     \@tempa
2162 }

\ModifyBibLabel
2163 \def\ModifyBibLabel#1{%
2164     \global\@xp\let\csname b@#1@suffix\endcsname\@empty
2165 }

```

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 104 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

$f_0\{A\}$
 $f_0\{A\} p_1 i_1 f_1\{B\}$
 $f_0\{A\} p_2 i_2 f_2\{B\} p_3 i_3 f_3\{B\}$

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} {f0} {p1} {i1f1} {p2} {i2f2} {p3} {i3f3}
{S} {\m{T1}\dots\m{Tn}} {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.

```
2166 \newcount\sseries@index
```

\PrintSeries

```
2167 \def\PrintSeries#1#2#3#4#5#6#7#8{%
2168   \begingroup
2169     \def\sseries@add@a{#2}%
2170     \def\sseries@add@b{\SwapBreak{#3}#4}%
2171     \def\sseries@add@c{\SwapBreak{#5}#6}%
2172     \def\sseries@add@d{\SwapBreak{#7}#8}%
2173     \def\sseries@add@e{\SwapBreak{#7}}%
2174     \PrintSeries@a{#1}%
2175 }
```

\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**.

```
2176 \def\PrintSeries@a#1#2#3#4{%
2177   \get@numberof\@tempcnta#1{#3}%
2178   \chardef\sseries@total=\@tempcnta
2179   \ifnum\sseries@total=\@ne
2180     \let\SingularPlural\@firstoftwo
2181   \else
2182     \let\SingularPlural\@secondoftwo
2183   \fi
2184   \series@index=\z@
2185   \let#1\sseries@add
2186   #2#3#4\relax
2187   \endgroup
2188 }
```

\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**.

```
2189 \def\sseries@add#1{%
```

```

2190     \advance\series@index\@ne
2191     \ifx\etaltext#1\relax
2192         \ifnum\series@index=\tw@
2193             \def\@tempa{\space\SubEal}%
2194         \else
2195             \def\@tempa{\series@add@e\space\SubEal}%
2196         \fi

```

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

```

2197     \series@index\@MM
2198     \else
2199         \ifcase\series@index
2200         \or

```

Material before name 1:

```

2201         \let\@tempa\series@add@a
2202         \or

```

Material before name 2:

```

2203         \ifnum\series@total<\thr@@
2204             \let\@tempa\series@add@b
2205         \else
2206             \let\@tempa\series@add@c
2207         \fi
2208     \else

```

Material before names 3, 4, 5,...

```

2209         \ifnum\series@index=\series@total
2210             \let\@tempa\series@add@d
2211         \else
2212             \ifnum\series@index<\series@total
2213                 \let\@tempa\series@add@c
2214             \else
2215                 \let\@tempa\@gobble
2216             \fi
2217         \fi
2218     \fi
2219 \fi
2220 \@tempa{#1}%
2221 }

```

\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.

```
2222 \def\SwapBreak#1{%
2223     \relax\ifvmode\leavevmode\fi
2224     \@tempcnta\@MM
2225     \toks@{#1}%
```

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

```
2226     \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.

```
2227     \ifnum \lastkern>\@ne \unkern \fi
```

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

```
2228     \ifnum\lastpenalty=\z@
2229     \else
2230         \@tempcnta\lastpenalty
2231         \unpenalty
2232     \fi
```

Now we add the punctuation, *unless* one of the following conditions is true:

1. The last item on the horizontal list was a kern of 1 sp, indicating that we’re at the very beginning of a bibliography item.
2. The current space factor is equal to the `\sfcode` of the punctuation mark we are adding, meaning that the mark is already on the list.
3. The current space factor is equal to the special value `\@nopunctsfcode`, meaning that `\nopunct` was specified.

This relies on distinct punctuation marks having distinct space factors, as established by our definition of `\frenchspacing`.

```
2233     \edef\@tempa{%
2234         \@nx\deferredquoteslogical
2235         \ifnum\lastkern=\@ne
2236         \else
2237             \ifnum\spacefactor=\sfcode\@xp\@xp\@xp'\@xp\@car\string#1)\@nil
2238             \else
2239                 \ifnum\spacefactor=\@nopunctsfcode
2240                 \else
2241                     \the\toks@
2242                 \fi
2243             \fi
```

```

2244         \fi
2245         \@nx\deferredquotes
2246         \ifnum\@tempcnta=\@MM \else \penalty\number\@tempcnta\space \fi
2247         \ifnum\lastkern=\@ne \ignorespaces \fi
2248     }%
2249     \@tempa
2250 }

```

`\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.

```

2251 \newcommand{\SingularPlural}[2]{#1}
2252 \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.)

```

2253 \BibSpec{nameLE}{
2254     +{}{}{given}
2255     +{}{\IfEmptyBibField{given}}{}{ }}{surname}
2256     +{}{ }{jr}
2257 }

```

`\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.”

```

2258 \BibSpec{nameBE}{
2259     +{}{}{surname}
2260     +{}{ }{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.

```

2261 %     +{}{ }{jr}
2262 }

```

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

```

2263 \BibSpec{nameinverted}{
2264     +{} {} {surname}
2265     +{,}{ } {given}
2266     +{,}{ } {jr}
2267 }

```

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.

```
2268 \newcommand{\PrintNames}{%
2269     \@ifstar{\PrintNames@a\set@othername}{\PrintNames@a\set@firstname}%
2270 }
```

\PrintNames@a

```
2271 \newcommand{\PrintNames@a}[4]{%
2272     \PrintSeries{\name}
2273         {#1}
2274         {}{ and \set@othername}
2275         {,}{ \set@othername}
2276         {,}{ and \set@othername}
2277         {#2}{#4}{#3}%
2278 }
```

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

```
2279 \def\set@firstname#1{%
2280     \set@name{#1}\setbib@nameLE
2281 }
```

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

```
2282 \def\set@othername#1{%
2283     \set@name{#1}\setbib@nameLE
2284 }
```

`\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.

```
2285 \def\set@name#1{%
2286     \name@split#1,,,\@nil
2287     \set@name@a
2288 }
```

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

```
2289 \def\set@name@a#1{%
2290     \begingroup
2291         \get@current@properties
2292         \select@auxlanguage
2293         \def\@tempa{yes}%
2294         \ifx\@tempa\prop'inverted
2295             \setbib@nameBE
2296         \else
2297             #1%
2298         \fi
2299     \endgroup
2300 }
```

`\PrintPrimary`

```
2301 \def\PrintPrimary{%
2302     \ifx\current@primary\@empty
2303         \EmptyPrimaryWarning
2304     \else
2305         \print@primary\current@primary
2306     \fi
2307 }
```

`\EmptyPrimaryWarning`

```
2308 \def\EmptyPrimaryWarning{%
2309     \amsrefs@warning{No authors, editors or translators}%
2310 }
```

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

```
2311 \newcommand{\PrintAuthors}[1]{%
2312     \ifx\previous@primary\current@primary
2313         \sameauthors\@empty
2314     \else
2315         \def\current@bibfield{\bib'author}%
2316         \PrintNames{}-{}{#1}%
2317     \fi
2318 }
```

`\sameauthors`

```
2319 \newcommand{\sameauthors}[1]{\bysame#1}
```

```

\bysame
2320 \def\bysame{%
2321     \leavevmode\hbox to3em{\hrulefill}\thinspace
2322     \kern\z@
2323 }

```

\PrintNameList This just prints the names without any additional information.

```
2324 \newcommand{\PrintNameList}{\PrintNames{}}{}}
```

\PrintEditorsC

```

2325 \newcommand{\PrintEditorsC}[1]{%
2326     \PrintNames{Edited by }{ }{#1}%
2327 }

```

\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.

```

2328 \newcommand{\PrintEditorsA}[1]{%
2329     \ifx\previous@primary\current@primary
2330         \sameauthors{(ed\Plural{s}.)}%
2331     \else
2332         \def\current@bibfield{\bib'editor}%
2333         \PrintNames{{ (ed\Plural{s}.)}{#1}%
2334     \fi
2335     \erase@field\bib'editor
2336 }

```

\PrintEditorsB

```

2337 \newcommand{\PrintEditorsB}{%
2338     \PrintNames*{{}\{ \SwapBreak{,}~ed\Plural{s}.)}%
2339 }

```

\PrintContributions

```

2340 \newcommand{\PrintContributions}[1]{%
2341     \PrintSeries
2342     {\fld@elt}
2343     {\print@contribution}
2344     {}{ and \print@contribution}
2345     {,}{ \print@contribution}
2346     {,}{ and \print@contribution}{ }{#1}{ }%
2347 }

```

\print@contribution

```

2348 \newcommand{\print@contribution}[1]{%
2349     \in@={#1}%
2350     \ifin@
2351         \ifnum\series@index=\@one with \fi
2352         \RestrictedSetKeys{{}\{bib}{ }%

```

```

2353         \bib@print@inner\setbib@contribution{\the\rsk@toks}%
2354     }{#1}%
2355     \else
2356         #1%
2357     \fi
2358 }

```

\resolve@inner

```

2359 \def\resolve@inner#1#2{%
2360     \in@=#2}%
2361     \ifin@
2362         \RestrictedSetKeys-{#1}{\the\rsk@toks}{#2}%
2363     \else
2364         \@ifundefined{bi@#2}{%
2365             \XRefWarning{#2}%
2366         }{%
2367             #1{\csname bi@#2\endcsname}%
2368         }%
2369     \fi
2370 }

```

\PrintConference

```

2371 \def\PrintConference{%
2372     \resolve@inner{\bib@print@inner\setbib@conference}
2373 }

```

\PrintConferenceDetails

```

2374 \def\PrintConferenceDetails#1{%
2375     \ifx\@empty\bib'address
2376         \ifx\@empty\bib'date
2377             \else
2378                 \PrintConferenceDetails@
2379             \fi
2380         \else
2381             \PrintConferenceDetails@
2382         \fi
2383 }

```

\PrintConferenceDetails@

```

2384 \def\PrintConferenceDetails@{%
2385     \ifnum\lastkern=\@ne\else\space\fi(\kern 1sp
2386     \ifx\@empty\bib'address
2387         \else
2388             \bib'address
2389         \fi
2390     \ifx\@empty\bib'date
2391         \else
2392             \SwapBreak{,}\space
2393             \print@date

```

```

2394     \fi
2395     )%\spacefactor\sfcode'\,%
2396 }

```

\PrintBook

```

2397 \def\PrintBook{%
2398     \resolve@inner{\bib@print@inner\setbib@innerbook}
2399 }

```

\PrintReprint

```

2400 \def\PrintReprint{%
2401     \resolve@inner{\bib@reprint}
2402 }

```

\bib@reprint

```

2403 \def\bib@reprint#1{%
2404     \begingroup
2405     #1\relax           % execute definitions locally
2406     \bib@resolve@xrefs
2407     \bib@field@patches
2408     \bib'setup
2409     \IfEmptyBibField{copula}{reprinted in}{\bib'copula} \nopunct
2410     \let\bib'language\@empty
2411     \setbib@book
2412     \endgroup
2413 }

```

\PrintTranslation

```

2414 \def\PrintTranslation{%
2415     \resolve@inner{\bib@translation}
2416 }

```

\bib@translation

```

2417 \def\bib@translation#1{%
2418     \begingroup
2419     #1\relax           % execute definitions locally
2420     \bib@resolve@xrefs
2421     \bib@field@patches
2422     \bib'setup
2423     \let\PrintPrimary\@empty
2424     \bib@append{;}{ % keep this space!
2425         \IfEmptyBibField{language}{English}{\bib'language} transl.%
2426         \IfEmptyBibField{pages}{ in \kern\@ne sp}{, }%
2427     }\bib'transition
2428     \let\bib'language\@empty
2429     \setbib@@
2430     \endgroup
2431 }

```

`\PrintTranslatorsC`

```
2432 \newcommand{\PrintTranslatorsC}[1]{%
2433   \PrintNames{translated by }{ }{#1}%
2434 }
```

`\PrintTranslatorsA`

```
2435 \newcommand{\PrintTranslatorsA}[1]{%
2436   \ifx\previous@primary\current@primary
2437     \sameauthors{(trans.)}%
2438   \else
2439     \def\current@bibfield{\bib'translator}%
2440     \PrintNames{{ (trans.)}{#1}%
2441   \fi
2442   \erase@field\bib'translator
2443 }
```

`\PrintTranslatorsB`

```
2444 \newcommand{\PrintTranslatorsB}[1]{
2445   \PrintNames*{({}\SwapBreak{,}~tran\Plural{s}.)}%
2446 }
```

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

```
2447 \DefineName{alii}{\etaltext}
2448 \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’.)

```
2449 \newcommand{\etaltext}{et al.}
2450 \newcommand{\SubEtal}[1]{\etaltext}
```

6.23 The partial field

`\print@partial`

```
2451 \newcommand{\print@partial}{%
2452   \resolve@inner{\bib@print@inner\setbib@partial}
2453 }
```

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.

```
2454 \newcommand{\parenthesize}[1]{%
2455   \leavevmode\push@bracket\upn{({#1\pop@bracket
2456 }
```

`\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.)

```
2457 \providecommand{\upn}[1]{#1}
```

```

\push@bracket
\pop@bracket 2458 \let\bracket@stack\@empty
                2459
                2460 \def\push@bracket#1{%
                2461     \xdef\bracket@stack{#1\bracket@stack}%
                2462 }
                2463
                2464 \def\pop@bracket{%
                2465     \iffalse{\fi
                2466     \xp\pop@bracket@a\bracket@stack \@empty}%
                2467 }
                2468
                2469 \def\pop@bracket@a#1{%
                2470     \leavevmode\/\upn{#1}%
                2471     \xdef\bracket@stack{\iffalse}\fi
                2472 }

\bibquotes
                2473 \newcommand{\bibquotes}[1]{%
                2474     \textquotedblleft#1%
                2475     \gdef\deferredquotes{%
                2476         \global\let\deferredquotes\@empty
                2477         \textquotedblright
                2478     }%
                2479 }

\mdash Cf. textcmds, where there's also a penalty added.
\ndash 2480 \providecommand{\mdash}{\textendash}
                2481 \providecommand{\ndash}{\textendash}

\@scanforMR
                2482 \def\@scanforMR#1#2#3#4\@nil{%
                2483     \def\@tempa{#1#2#3#4}%
                2484     \if#1M%
                2485         \if#2R%
                2486             \def\@tempa{#3#4}%
                2487         \fi
                2488     \fi
                2489 }

\MR
                2490 \def\MR#1{%
                2491     \relax\ifhmode\unskip\spacefactor3000 \space\fi
                2492     \begingroup
                2493         \@scanforMR#1\@nil
                2494         \edef\@tempa{MR\@nx\MRhref{\@tempa}{\@tempa}}%
                2495     \exp\endgroup
                2496     \@tempa
                2497 }

```

```
2498 \providecommand{\MRhref}[2]{#1}
```

```

2499 \newcommand{\PrintReviews}[1]{%
2500     \PrintSeries{\fld@elt}{},{ }{ },{ }{ }{ }\#1}{}%
2501 }

```

```

2502 \newcommand{\PrintPartials}[1]{%
2503   \PrintSeries
2504     {\fld@elt}
2505     {\print@partial}
2506     {;}{ \print@partial}
2507     {;}{ \print@partial}
2508     {;}{ \print@partial}{-}{#1}{-}%
2509 }

```

```

2510 \newcommand{\PrintISBNs}[1]{%
2511     \PrintSeries{\fld@elt}{-}{ }{-}{ }{-}{ }{-}ISBN }{#1}{}%
2512 }

```

```
2513 \newcommand{\voltext}{\IfEmptyBibField{series}{Vol.~}{vol.~}}
```

```
2514 \newcommand{\issuetext}{no.~}
```

```

2515 \newcommand{\DashPages}[1]{%
2516     p\pp@scan@a#1\ndash p\ndash{\pp@scan#1@-p@-}\@nil}\@nil.~#1%
2517 }
2518
2519 \def\pp@scan#1-#2@-#3#4\@nil{#3}
2520
2521 \def\pp@scan@a#1\ndash#2\ndash#3#4\@nil{#3}

```

```

2522 \newcommand{\eprintpages}[1]{%
2523     #1\IfEmptyBibField{eprint}{-}\IfEmptyBibField{journal}{ pp.}{-}}%
2524 }

```

\PrintThesisType

```

2525 \def\PrintThesisType#1{%
2526     \thesis@type#1?\@nil{#1}%
2527 }
2528
2529 \def\thesis@type#1#2\@nil#3{%
2530     \ifx p#1%
2531         Ph.D. Thesis%
2532     \else
2533         \ifx m#1%
2534             Master's Thesis%
2535         \else
2536             #3%
2537         \fi
2538     \fi
2539 }

```

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

```

2540 \newcommand{\PrintDOI}[1]{%
2541     DOI #1%
2542     \IfEmptyBibField{volume}{, (to appear in print)}{}%
2543 }

```

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

```

2544 \newcommand{\PrintDatePV}[1]{%
2545     \IfEmptyBibField{doi}{%
2546         \let\@tempa\PrintDate
2547     }{%
2548         \IfEmptyBibField{volume}{%
2549             \let\@tempa\PrintDatePosted
2550         }{%
2551             \let\@tempa\PrintDate
2552         }%
2553     }%
2554     \@tempa{#1}%
2555 }

```

\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.

```

2556 \newcommand{\PrintDate}[1]{(\print@date)}

```

\PrintDateB The same, but without the parentheses.

```

2557 \newcommand{\PrintDateB}[1]{\print@date}

```

`\print@date`

```
2558 \def\print@date{%
2559     \ifx\bib@month\@empty
2560     \else
2561         \print@month@day
2562     \fi
2563     \bib@year
2564 }
```

`\print@month@day`

```
2565 \def\print@month@day{%
2566     \bib@monthname
2567     \ifx\@empty\bib@day \else \nobreakspace\number 0\bib@day,\fi
2568     \space
2569 }
```

`\bib@monthname` With the Babel package, month names for a given language are typically available in a macro `\month@language`:

```
\def\month@german{\ifcase\month\or
    Januar\or Februar\or M"arz\or April\or Mai\or Juni\or
    Juli\or August\or September\or Oktober\or November\or Dezember\fi}
```

However this is not true for English.

```
2570 \newcommand{\bib@monthname}{%
2571     \ifcase 0\bib@month
2572     \or January\or February\or March\or April\or May\or June\or
2573     July\or August\or September\or October\or November\or December\or
2574     Winter\or Spring\or Summer\or Fall\else Unknown Month%
2575     \fi
2576 }
```

`\PrintYear` You can use `\PrintYear` if you want to suppress month/day even when supplied in the data.

```
2577 \newcommand{\PrintYear}[1]{\bib@year}
```

`\PrintDatePosted` This one is special for AMS use.

```
2578 \newcommand{\PrintDatePosted}[1]{\unskip, posted on \print@date}
```

`\PrintEdition`

```
2579 \newcommand{\PrintEdition}[1]{%
2580     \afterassignment\print@edition
2581     \count@ 0#1\relax\@nil
2582 }
```

`\print@edition` If the number assignment swept up all the contents, produce a cardinal number from `\count@`.

```
2583 \def\print@edition#1#2\@nil{%
2584     \ifx\relax#1\relax
```

```

2585         \ifnum\count@>\z@
2586             \CardinalNumeric\count@
2587         \else
2588             ??th%
2589         \fi
2590     \ \editiontext
2591 \else
2592     \ifnum \count@>\z@ \number\count@ \fi
2593     #1#2\relax
2594 \fi
2595 }

\editiontext
2596 \newcommand{\editiontext}{ed.}

\CardinalNumber
2597 \newcommand{\CardinalNumeric}[1]{%
2598     \number#1\relax
2599     \if
2600         \ifnum#1<14
2601             \ifnum#1>\thr@@ T\else F\fi
2602         \else
2603             F%
2604         \fi
2605         T%
2606         th%
2607     \else
2608         \@xp\keep@last@digit\@xp#1\number#1\relax
2609         \ifcase#1th\or st\or nd\or rd\else th\fi
2610     \fi
2611 }

\keep@last@digit
2612 \def\keep@last@digit#1#2{%
2613     \ifx\relax#2%
2614         \@xp\@gobbletwo
2615     \else
2616         #1=#2\relax
2617     \fi
2618     \keep@last@digit#1%
2619 }

\SentenceSpace Note how careful we are here to preserve \frenchspacing.
2620 \newcommand{\SentenceSpace}{\relax\ifhmode\spacefactor'\. \fi}

\epprint For now, this does nothing. Could do a url/hyperlink or something.
2621 \newcommand{\epprint}[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.

```
2622 \let\bibliographystyle\@gobble
```

`\bibtex@style`

```
2623 \def\bibtex@style{amsrn}

2624 \AtBeginDocument{
2625   \if@filesw
2626     \immediate\write\@auxout{\string\bibstyle{\bibtex@style}}%
2627   \fi
2628 }
```

6.26 Implementing package options

6.26.1 The alphabetic option

```
2629 \IfOption{alphabetic}{%
2630   \def\bibtex@style{amsra}%
2631   \def\alpha@label{%
2632     \ifx\@empty\bib'label
2633       \def\thebib{\CurrentBib}%
2634     \else
2635       \let\thebib\bib'label
2636     \fi
2637   }%
2638   \let\generate@label\generate@alphalabel
2639   \let\process@citelist\process@citelist@unsorted
2640   \def\numeric@refs{01}%
2641 }{}
```

6.26.2 The shortalphabetic option

```
2642 \IfOption{shortalphabetic}{%
2643   \def\bibtex@style{amsrs}%
2644   \def\alpha@label{%
2645     \ifx\@empty\bib'label
2646       \def\thebib{\CurrentBib}%
2647     \else
2648       \let\thebib\bib'label
2649     \fi
2650   }%
2651   \let\@suffix@format\@arabic
2652   \def\calc@author@part{%
2653     \xp\@multiauthorlabel\@xp{\@tempa}%
```

```

2654 }%
2655 \let\append@label@year\empty
2656 \let\generate@label\generate@alphalabel
2657 \let\process@citelist\process@citelist@unsorted
2658 \def\numeric@refs{01}%
2659 }{}

```

6.26.3 The backrefs option

```

2660 \IfOption{backrefs}{%
2661   \let\PrintBackRefs\print@backrefs
2662   \@ifundefined{Hy@backout}{%
2663     \amsrefs@warning{backref option requires hyperref package}%
2664   }{%
2665     \let\BackCite\back@cite
2666     \AtBeginDocument{%
2667       \let\ams@saved@toc\tableofcontents
2668       \@starttoc{brf}{}%
2669       \let\tableofcontents\ams@saved@toc
2670     }%
2671   }%
2672 }{}
2673 }

```

6.26.4 The citation-order option

```

2674 \IfOption{citation-order}{%
2675   \IfOption{alphabetic}{%
2676     \amsrefs@warning@nl{%
2677       The 'citation-order' and 'alphabetic' options are
2678       incompatible%
2679     }%
2680   }{
2681     \def\bibtex@style{amsru}%
2682   }
2683 }{}

```

6.26.5 The initials option

```

2684 \IfOption{initials}{% TRUE:
2685   \BibSpec{nameLE}{
2686     +{}{}{initials}
2687     +{}{\IfEmptyBibField{initials}{ } }{surname}
2688     +{}{ }{jr}
2689   }
2690
2691   \BibSpec{nameBE}{
2692     +{}{}{surname}
2693     +{}{ }{initials}
2694     % +{}{ }{jr}
2695   }
2696
2697   \BibSpec{nameinverted}{
2698     +{} {} {surname}

```

```

2699      +{,}{ } {initials}
2700      +{,}{ } {jr}
2701    }
2702 }{% initials? FALSE:
2703 %   \let\extract@initials\@gobble
2704 } % end conditional code for initials option

```

6.26.6 The logical-quotes option

`\deferredquotes`

```

2705 \let\deferredquotes\@empty

```

`\deferredquoteslogical`

```

2706 \IfOption{logical-quotes}{%
2707   \def\deferredquoteslogical{\deferredquotes}%
2708 }{%
2709   \let\deferredquoteslogical\relax
2710 }

```

6.26.7 The non-compressed-cites option

```

2711 \IfOption{non-compressed-cites}{%
2712   \let\cite@compress\cite@print
2713 }{}

```

6.26.8 The non-sorted-cites option

```

2714 \IfOption{non-sorted-cites}{%
2715   \let\process@citelist\process@citelist@unsorted
2716 }{}

```

6.26.9 The short-journals option

```

2717 \IfOption{short-journals}{%
2718   \renewcommand{\DefineJournal}[4]{%
2719     \bib*{#1}{periodical}{
2720       issn={#2},
2721       journal={#3},
2722     }%
2723   }
2724 }{}

```

6.26.10 The short-publishers option

```

2725 \IfOption{short-publishers}{%
2726   \renewcommand{\DefinePublisher}[4]{%
2727     \bib*{#1}{publisher}{%
2728       publisher={#2},%

```

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

```

2729       address={#4},
2730     }%
2731   }%
2732 }{}

```

6.26.11 The short-months option

```

2733 \IfOption{short-months}{%
2734     \renewcommand{\bib@monthname}{%
2735         \ifcase 0\bib@month
2736         \or Jan.\or Feb.\or Mar.\or Apr.\or May\or June\or
2737         July\or Aug.\or Sep.\or Oct.\or Nov.\or Dec.\or
2738         Winter\or Spring\or Summer\or Fall\else Unknown Month%
2739     \fi
2740 }%
2741 }{}

```

6.26.12 The y2k option

```

2742 \IfOption{y2k}{%
2743     \IfOption{alphabetic}{%
2744         \def\year@short#1\@nil{#1}%
2745         \def\bibtex@style{amsry}%
2746     }{%
2747         \amsrefs@warning@nl{%
2748             The 'y2k' option can only be used with the^^J%
2749             'alphabetic' option%
2750         }%
2751     }
2752 }{}

```

6.26.13 The bibtex-style option

```

2753 \IfOption{bibtex-style}{%
2754     \RequirePackage{amsbst}
2755 }{}

```

6.26.14 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.

```

2756 \IfOption{author-year}{}{\PopCatcodes \endinput}

```

`\generate@label`

```

2757 \def\generate@label{%
    If the user supplied an explicit label field, we use it. Otherwise, we generate
    our own.
2758     \ifx\bib'label\@empty
2759         \begingroup
    We begin by saving the previous stem and initializing the current stem to the
    empty string.
2760         \global\let\previous@stem\current@stem
2761         \global\let\current@stem\@empty
2762         \global\let\previous@year\current@year
2763         \global\let\current@year\bib@year
    The list of primary contributors is available to us in \current@primary in the
    form

```

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

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

```

2764      \@apply\auto@protect\amsrefs@textsymbols
2765      \@apply\auto@protect\amsrefs@textaccents
2766      \def\name##1{\@nx\name{\lncan@a##1,\@nil}}%
2767      \auto@protect\etaltext
2768      \normalize@edef\current@stem{\current@primary}%
2769      \xdef\current@stem{\current@stem}%

```

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

```

2770      \calc@alpha@suffix

```

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

```

2771      \edef\@tempa{%
2772          \def\@nx\cite@label{\current@stem}%
2773          \def\@nx\bib@label@year{%
2774              \current@year
2775              \alpha@label@suffix
2776          }%
2777      }
2778      \@xp\endgroup
2779      \@tempa
2780  \fi
2781 }

```

$\backslash\text{lncan@a}$

```

2782 \def\lncan@a#1,#2\@nil{#1}

```

$\backslash\text{citesel@author}$

```

2783 \def\citesel@author#1#2#3#4#5{\PrintCiteNames{#3}}

```

$\backslash\text{citesel@authoryear}$

```

2784 \def\citesel@authoryear#1#2#3#4#5{\PrintCNY{#3}{#4}}

```

$\backslash\text{citesel@object}$

```

2785 \def\citesel@object#1#2#3#4#5{\PrintCiteNames{#3} \citeleft#4}

```

$\backslash\text{citesel}$

```

2786 \let\citesel\citesel@authoryear

```

$\backslash\text{numeric@refs}$

```

2787 \def\numeric@refs{01}%

```

```

\citeleft
2788 \def\citeleft{()%

\citeright
2789 \def\citeright{)%

\@citeleft
2790 \def\@citeleft{\ifx\citesel\citesel@object\else\citeleft\fi}%

\citepunct
2791 \def\citepunct{; }

\BibLabel
2792 \def\BibLabel{%
2793   \hyper@anchorstart{cite.\CurrentBib}\relax\hyper@anchorend
2794 }

\process@citelist
2795 \let\process@citelist\process@citelist@unsorted

\ycite
2796 \DeclareRobustCommand{\ycite}[1]{%
2797   \star@{\cite@a\citesel@year{#1}}{}}%
2798 }

\ycites
2799 \DeclareRobustCommand{\ycites}[1]{%
2800   \begingroup
2801     \def\citepunct{, }%
2802     \let\citesel\citesel@year
2803     \cites{#1}%
2804   \endgroup
2805 }

\ocite
2806 \DeclareRobustCommand{\ocite}[1]{%
2807   \star@{\cite@a\citesel@object{#1}}{}}%
2808 }

\ocites
2809 \DeclareRobustCommand{\ocites}[1]{%
2810   \begingroup
2811     \let\@citelist\@ocitelist
2812     \cites{#1}%
2813   \endgroup
2814 }

```

\ocitelist

```
2815 \def\@ocitelist#1{%
2816     \PrintSeries{\InnerCite}%
2817     {\ocite}%
2818     {}{ 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.

```
2819     {,}{ \ocite}%
2820     {,}{ and \ocite}%
2821     {}%
2822     {#1}%
2823     {}%
2824 }
```

\citeauthor

```
2825 \DeclareRobustCommand{\citeauthor}[1]{%
2826     \star@\{\cite@a\citesel@author{#1}\}%
2827 }
```

\citeauthory

```
2828 \DeclareRobustCommand{\citeauthory}[1]{%
2829     \citeauthor{#1} \ycite{#1}%
2830 }
```

\fullcite

```
2831 \DeclareRobustCommand{\fullcite}[1]{%
2832     \begingroup
2833     \let\print@citenames\CiteNamesFull
2834     \star@\{\cite@a\citesel@authoryear{#1}\}%
2835     \endgroup
2836 }
```

\fullocite

```
2837 \DeclareRobustCommand{\fullocite}[1]{%
2838     \begingroup
2839     \let\print@citenames\CiteNamesFull
2840     \star@\{\cite@a\citesel@object{#1}\}%
2841     \endgroup
2842 }
```

Invert the first author’s name.

```
2843 \def\set@firstname#1{%
2844     \set@name{#1}\setbib@nameinverted
2845 }
```

\PrintCNY

```
2846 \def\PrintCNY#1#2{%
```

```

2847 \PrintCiteNames{#1}%
2848 \@ifnotempty{#2}{\@addpunct{,} #2}%
2849 }

```

\PrintCiteNames

```

2850 \def\PrintCiteNames#1{%
2851   \leavevmode
2852   \def\@tempa{#1}%
2853   \ifx\@tempa\prev@names
2854   \else
2855     \gdef\prev@names{#1}%
2856     \@xp\ifx\@car#1.\@nil\CitePrintUndefined
2857       #1\relax
2858   \else
2859     \print@citenames{#1}%
2860   \fi
2861 \fi
2862 }

```

\CiteNames

```

2863 \newcommand{\CiteNames}[1]{%
2864   \PrintSeries{\name}%
2865   {}%
2866   {}{ 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.

```

2867   {}{\@gobble}%
2868   {}{ \etaltext\@gobble}%
2869   {}%
2870   {#1}%
2871   {}%
2872 }

```

\print@citenames

```

2873 \let\print@citenames\CiteNames

```

\CiteNamesFull

```

2874 \newcommand{\CiteNamesFull}[1]{%
2875   \PrintSeries{\name}%
2876   {}%
2877   {}{ 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.

```

2878   {,}{ }%
2879   {,}{ and }%
2880   {}%

```

```

2881      {#1}%
2882      {}%
2883 }

```

`\PrintDate` No parentheses around the year.

```

2884 \renewcommand{\PrintDate}[1]{\bib@label@year}

```

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

```

2885 \def\print@date{%
2886   \IfEmptyBibField{date}{%
2887     \IfEmptyBibField{year}{\BibField{status}}{\bib@year}%
2888   }{%
2889     \bib@year
2890   }%
2891 }

2892 \BibSpec{article}{%
2893   +{ } {\PrintAuthors}           {author}
2894   +{.} { \PrintDate}             {date}
2895   +{.} { \textit}                {title}
2896   +{.} { }                       {part}
2897   +{:} { \textit}                {subtitle}
2898   +{,} { \PrintContributions}    {contribution}
2899   +{.} { \PrintPartials}         {partial}
2900   +{,} { }                       {journal}
2901   +{ } { \textbf}                {volume}
2902   +{,} { \issuetext}             {number}
2903   +{,} { \eprintpages}          {pages}
2904   +{,} { }                       {status}
2905   +{,} { \PrintDOI}              {doi}
2906   +{,} { available at \eprint}   {eprint}
2907   +{ } { \parenthesize}          {language}
2908   +{ } { \PrintTranslation}      {translation}
2909   +{;} { \PrintReprint}          {reprint}
2910   +{.} { }                       {note}
2911   +{.} { }                       {transition}
2912   +{ } { \SentenceSpace \PrintReviews} {review}
2913 }
2914
2915 \BibSpec{book}{%
2916   +{ } {\PrintPrimary}           {transition}
2917   +{.} { \PrintDate}             {date}
2918   +{.} { \textit}                {title}
2919   +{.} { }                       {part}
2920   +{:} { \textit}                {subtitle}
2921   +{,} { \PrintEdition}          {edition}
2922   +{ } { \PrintEditorsB}         {editor}
2923   +{,} { \PrintTranslatorsC}     {translator}
2924   +{,} { \PrintContributions}    {contribution}
2925   +{,} { }                       {series}

```

```

2926 +{,} { \voltext} {volume}
2927 +{,} { } {publisher}
2928 +{,} { } {organization}
2929 +{,} { } {address}
2930 +{,} { } {status}
2931 +{ } { \parenthesize} {language}
2932 +{ } { \PrintTranslation} {translation}
2933 +{;} { \PrintReprint} {reprint}
2934 +{.} { } {note}
2935 +{.} {} {transition}
2936 +{ } {\SentenceSpace \PrintReviews} {review}
2937 }
2938
2939 \BibSpec{collection.article}{%
2940 +{ } {\PrintAuthors} {author}
2941 +{.} { \PrintDate} {date}
2942 +{.} { \textit} {title}
2943 +{.} { } {part}
2944 +{:} { \textit} {subtitle}
2945 +{,} { \PrintContributions} {contribution}
2946 +{,} { \PrintConference} {conference}
2947 +{ } {\PrintBook} {book}
2948 +{,} { } {booktitle}
2949 +{,} { pp.~} {pages}
2950 +{,} { } {status}
2951 +{,} { \PrintDOI} {doi}
2952 +{,} { available at \eprint} {eprint}
2953 +{ } { \parenthesize} {language}
2954 +{ } { \PrintTranslation} {translation}
2955 +{;} { \PrintReprint} {reprint}
2956 +{.} { } {note}
2957 +{.} {} {transition}
2958 +{ } {\SentenceSpace \PrintReviews} {review}
2959 }
2960
2961 \BibSpec{report}{%
2962 +{ } {\PrintPrimary} {transition}
2963 +{.} { \PrintDate} {date}
2964 +{.} { \textit} {title}
2965 +{.} { } {part}
2966 +{:} { \textit} {subtitle}
2967 +{,} { \PrintEdition} {edition}
2968 +{,} { \PrintContributions} {contribution}
2969 +{,} { Technical Report } {number}
2970 +{,} { } {series}
2971 +{,} { } {organization}
2972 +{,} { } {address}
2973 +{,} { \eprint} {eprint}
2974 +{,} { } {status}
2975 +{ } { \parenthesize} {language}

```

```

2976 +{ } { \PrintTranslation} {translation}
2977 +{;} { \PrintReprint} {reprint}
2978 +{.} { } {note}
2979 +{.} {} {transition}
2980 +{ } {\SentenceSpace \PrintReviews} {review}
2981 }
2982
2983 \BibSpec{thesis}{%
2984 +{ } {\PrintAuthors} {author}
2985 +{.} { \PrintDate} {date}
2986 +{.} { \textit} {title}
2987 +{:} { \textit} {subtitle}
2988 +{,} { \PrintThesisType} {type}
2989 +{,} { } {organization}
2990 +{,} { } {address}
2991 +{,} { \eprint} {eprint}
2992 +{,} { } {status}
2993 +{ } { \parenthesize} {language}
2994 +{ } { \PrintTranslation} {translation}
2995 +{;} { \PrintReprint} {reprint}
2996 +{.} { } {note}
2997 +{.} {} {transition}
2998 +{ } {\SentenceSpace \PrintReviews} {review}
2999 }
3000 \PopCatcodes
3001 \</pkg>

```

6.27 The amsbst package

```

3002 \*bst>
3003 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3004 \ProvidesPackage{amsbst}[2004/03/29 v1.68]
3005 %\RequirePackage{amsrefs}[2004/03/29]
3006 \BibSpec{article}{%
3007 +{ } {\PrintAuthors} {author}
3008 +{.} { } {title}
3009 +{.} { } {part}
3010 +{:} { } {subtitle}
3011 +{.} { \PrintContributions} {contribution}
3012 +{.} { \PrintPartials} {partial}
3013 +{.} { \emph} {journal}
3014 +{ } { } {volume}
3015 +{ } { \parenthesize} {number}
3016 +{:} {} {pages}
3017 +{,} { \PrintDateB} {date}
3018 +{,} { } {status}
3019 +{.} { \PrintTranslation} {translation}
3020 +{.} { Reprinted in \PrintReprint} {reprint}
3021 +{.} { } {note}
3022 +{.} {} {transition}

```

```

3023 }
3024
3025 \BibSpec{partial}{%
3026     +{ } { } {part}
3027     +{:} { } {subtitle}
3028     +{.} { \PrintContributions} {contribution}
3029     +{.} { \emph} {journal}
3030     +{ } { } {volume}
3031     +{ } { \parenthesize} {number}
3032     +{:} { } {pages}
3033     +{,} { \PrintDateB} {date}
3034 }
3035
3036 \BibSpec{book}{%
3037     +{ } { \PrintPrimary} {transition}
3038     +{.} { \emph} {title}
3039     +{.} { } {part}
3040     +{:} { \emph} {subtitle}
3041     +{.} { } {series}
3042     +{,} { \voltext} {volume}
3043     +{.} { Edited by \PrintNameList} {editor}
3044     +{.} { Translated by \PrintNameList} {translator}
3045     +{.} { \PrintContributions} {contribution}
3046     +{.} { } {publisher}
3047     +{.} { } {organization}
3048     +{,} { } {address}
3049     +{,} { \PrintEdition} {edition}
3050     +{,} { \PrintDateB} {date}
3051     +{.} { } {note}
3052     +{.} { } {transition}
3053     +{.} { \PrintTranslation} {translation}
3054     +{.} { Reprinted in \PrintReprint} {reprint}
3055     +{.} { } {transition}
3056 }
3057
3058 \BibSpec{collection.article}{%
3059     +{ } { \PrintAuthors} {author}
3060     +{.} { } {title}
3061     +{.} { } {part}
3062     +{:} { } {subtitle}
3063     +{.} { \PrintContributions} {contribution}
3064     +{.} { \PrintConference} {conference}
3065     +{.} { \PrintBook} {book}
3066     +{.} { In } {booktitle}
3067     +{,} { pages~} {pages}
3068     +{.} { \PrintDateB} {date}
3069     +{.} { \PrintTranslation} {translation}
3070     +{.} { Reprinted in \PrintReprint} {reprint}
3071     +{.} { } {note}
3072     +{.} { } {transition}

```

```

3073 }
3074
3075 \BibSpec{conference}{%
3076     +{ } { } {title}
3077     +{ } {\PrintConferenceDetails} {transition}
3078 }
3079
3080 \BibSpec{innerbook}{%
3081     +{.} { \emph} {title}
3082     +{.} { } {part}
3083     +{:} { \emph} {subtitle}
3084     +{.} { } {series}
3085     +{,} { \voltext} {volume}
3086     +{.} { Edited by \PrintNameList} {editor}
3087     +{.} { Translated by \PrintNameList} {translator}
3088     +{.} { \PrintContributions} {contribution}
3089     +{.} { } {publisher}
3090     +{.} { } {organization}
3091     +{,} { } {address}
3092     +{,} { \PrintEdition} {edition}
3093     +{,} { \PrintDateB} {date}
3094     +{.} { } {note}
3095     +{.} { } {transition}
3096 }
3097
3098 \BibSpec{report}{%
3099     +{ } {\PrintPrimary} {transition}
3100     +{.} { \emph} {title}
3101     +{.} { } {part}
3102     +{:} { \emph} {subtitle}
3103     +{.} { \PrintContributions} {contribution}
3104     +{.} { Technical Report } {number}
3105     +{,} { } {series}
3106     +{.} { } {organization}
3107     +{,} { } {address}
3108     +{,} { \PrintDateB} {date}
3109     +{.} { \PrintTranslation} {translation}
3110     +{.} { Reprinted in \PrintReprint} {reprint}
3111     +{.} { } {note}
3112     +{.} { } {transition}
3113 }
3114
3115 \BibSpec{thesis}{%
3116     +{ } {\PrintAuthors} {author}
3117     +{,} { \emph} {title}
3118     +{:} { \emph} {subtitle}
3119     +{.} { \PrintThesisType} {type}
3120     +{.} { } {organization}
3121     +{,} { } {address}
3122     +{,} { \PrintDateB} {date}

```

```

3123   +{.} { \PrintTranslation}           {translation}
3124   +{.} { Reprinted in \PrintReprint}   {reprint}
3125   +{.} { }                             {note}
3126   +{.} {}                             {transition}
3127 }

```

`\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.

```

3128 \renewcommand{\PrintEditorsA}[1]{%
3129   \def\current@bibfield{\bib'editor}%
3130   \PrintNames{[, editor\Plural{s}]}{#1}%
3131   \erase@field\bib'editor
3132 }

```

`\PrintTranslatorsA`

```

3133 \renewcommand{\PrintTranslatorsA}[1]{%
3134   \def\current@bibfield{\bib'translator}%
3135   \PrintNames{[, translator\Plural{s}]}{#1}%
3136   \erase@field\bib'translator
3137 }

```

3138 `\bst`

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

3139 `\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			
<code>\</code> 64, 1642	<code>.bbl</code> . . . 7, 8, 27, 29, 45	<code>\@MM</code> . . . 2197, 2224, 2246	
<code>\&</code> 25	<code>.bib</code> 7	<code>\@addpunct</code> 5,	
<code>\'</code> 40, 64, 1643	<code>.dtx</code> 7	14, <u>107</u> , 1310, 2848	
<code>\+</code> 24	<code>.ltx</code> 27, 29, 45	<code>\@alpha</code> 2122	
<code>\,</code> 24, 119, 2395	<code>\:</code> . . . 24, 117, 611, 627	<code>\@apply</code> <u>68</u>	
<code>\-</code> 24	<code>\;</code> 24, 118	<code>\@arabic</code> 2651	
<code>\.</code> . . 24, 114, 1644, 2620	<code>\=</code> 24, 1645	<code>\@bibdef</code> 28, 29, 562,	
<code>.aux</code> . 7-9, 48, 59, 61, 62	? option 16	572, <u>577</u> , 589,	
	<code>\@undefined</code> 1347	1123, 1126, 1142	

- \@biblist 475, 480, 484
 - \@car . 2059, 2237, 2856
 - \@carcube 2060
 - \@chomp . . 85, 1837,
1849, 1902, 1903
 - \@chomper . . 14, 88, 102
 - \@citeleft
. 1374, 1382, 2790
 - \@citelist 51,
1297, 1298, 2811
 - \@citestyle . 1301, 1318
 - \@citex 53
 - \@close@bbl@file .
. 1139,
1211, 1220, 1230
 - \@concat 59, 2027
 - \@count@chars 2011, 2016
 - \@currentlabel 986, 1028
 - \@currentxt 135
 - \@currname 137
 - \@empty 13, 18–20, 46, 47
 - \@endbiblist
. . . . 477, 482, 504
 - \@extract@initial
. 80,
1977, 1979, 1991
 - \@filef@und . . 1156,
1159, 1162, 1176
 - \@firstofone
. . 15, 39, 962,
1274, 1296, 1339
 - \@firstoftwo
. . . . 48, 976, 2180
 - \@firstone . 2059, 2061
 - \@firstthree 2060, 2083
 - \@for 36
 - \@gobblefour 679
 - \@gobblethree . . 1392
 - \@ifempty 45,
256, 1044,
1243, 1268,
1354, 1526,
1532, 1548, 1600
 - \@ifnextchar 18
 - \@ifnotempty 45, 227,
644, 668, 1255,
1376, 2064, 2848
 - \@ifpackageloaded
. 1624, 1627, 1632
 - \@ifpackagewith . . 5, 8
 - \@ifsame@patterns
. 965, 971
 - \@ifsamepat . . 972, 974
 - \@input@ 1143
 - \@lappend
61, 1640, 1669,
1670, 1672, 1673
 - \@latex@warning . .
. 506, 1362
 - \@let@token 73, 80
 - \@listctr 493
 - \@m 110, 498, 537
 - \@marknames . 2063, 2075
 - \@maxlabelwidth 491,
509, 512, 513, 521
 - \@multiauthorlabel
. 2056, 2093, 2653
 - \@nametoken . . 2061,
2069, 2081, 2083
 - \@nameuse 67
 - \@nil 13,
39, 69, 71, 76, 77
 - \@nilgobble
71, 82, 1779, 2074
 - \@nmbrrlisttrue . . 492
 - \@noitemerr 506
 - \@nopunctsfcode . .
. . . 122, 123, 2239
 - \@ocitelist . 2811, 2815
 - \@oneauthorlabel .
. 2054, 2078
 - \@open@bbl@file 602,
1136, 1210, 1226
 - \@pass@options . . . 134
 - \@prev@cite 1503
 - \@scanforMR . 2482, 2493
 - \@secondoftwo . . .
. . . . 48, 978, 2182
 - \@sptoken . . 1791,
1866, 1916, 1964
 - \@starttoc 2668
 - \@suffix@format . .
. 2122,
2133, 2144, 2651
 - \@tabacckludge . 67,
1699, 1711, 1730
 - \@tempa 72
 - \@tempcnta 13, 59
 - \@tempcntb 59
 - \@tempdima . . . 511–513
 - \@tempswafalse . . .
. . 565, 1019, 2124
 - \@tempswatrue . 561,
1037, 1040, 2128
 - \@temptokena 72
 - \@testdef 43
 - \@threeauthors . . .
. 2089, 2103
 - \@uclclist 66
 - \@urlbreak 25, 27
 - \@urlfix 24, 32
 - \@urlsetup 20, 22
 - \@xifempty 45, 47
 - \^ 106, 1646
 - \' 1647
 - \~ 29, 1648
 - _ 1295, 2590
- A**
- \A 535
 - abbrv BibT_EX style . . 15
 - \add@toks@ 60
 - address field 2, 44, 105
 - \addto@defer@list
. 690, 694
 - \addto@hook . . . 60, 808
 - \AE 1678
 - \ae 66, 1678
 - \after@deleting@token
. . . . 41, 1798,
1808, 1876,
1914, 1930, 1973
 - \afterassignment 11,
42, 82, 701, 2580
 - \aftergroup . . . 21, 279
 - alpha BibT_EX style . . 15
 - \alpha@label . . 987,
990, 2631, 2644
 - \alpha@label@suffix
. 2045,
2118, 2133,
2140, 2144, 2775
 - \alpha@suffix 2120,
2121, 2132–
2134, 2141, 2144
 - alphabetic option .
. . . 15, 82, 85, 103

- \ams@saved@toc ...
 - 2667, 2669
- amsbook class 51
- amsbst package 113
- amsgen package 14
- amsproc class 51
- amsrdoc.tex 1
- amsrefs package ...
 - ... 1–6, 8, 16,
 - 45, 61, 89, 106, 116
- \amsrefs@bibcite .
 - . 1571, 1576, 1580
- \amsrefs@error ...
 - 105, 246, 296, 582
- \amsrefs@textaccents
 - 65,
 - 838, 1018, 1638,
 - 1640, 1698,
 - 1712, 1729,
 - 1830, 2034, 2765
- \amsrefs@textsymbols
 - 67, 837, 1017,
 - 1667, 1669,
 - 1670, 1672,
 - 1673, 1697,
 - 1713, 1727,
 - 1829, 2033, 2764
- \amsrefs@warning .
 - 104,
 - 881, 882, 1051,
 - 1065, 1147,
 - 1173, 1251,
 - 1256, 1261,
 - 1443, 1823,
 - 1895, 2309, 2663
- \amsrefs@warning@nl
 - 4, 6, 9, 2676, 2747
- amsxport BibT_EX style
 - 81, 82
- \append@label@year
 - . 2040, 2108, 2655
- \append@to@stem ..
 - .. 2027, 2085,
 - 2091, 2101, 2111
- \ar@hyperlink
 - . 1375, 1619, 1628
- \ar@SK@cite
 - . 1373, 1622, 1633
- article entry type . 3, 6
- \AtBeginDocument .
 - 577, 578, 1572,
 - 1623, 2624, 2666
- \AtEndDocument .. 1081
- author field 4
- author-year option
 - .. 51, 85, 92, 106
- \AuthorList 6
- \auto@protect 12, 57,
 - 68, 837, 838,
 - 999, 1016–1018,
 - 1697, 1712,
 - 1720, 1727,
 - 1826, 1827,
 - 1829, 1830,
 - 2033–2036,
 - 2764, 2765, 2767
- B**
- \b 1649
- \b@ 41, 53, 54, 57, 61, 62
- \b@* 1433
- \b@moo 7, 8, 48
- \b@whatever ... 8, 9, 54
- babel package 40
- \back@cite . 1608, 2665
- \BackCite
 - . 1327, 1607, 2665
- backrefs option . 8, 103
- \bbl@copy ... 607, 661
- \bbl@copy@a .. 665, 667
- \bbl@out 1208,
 - 1218, 1219,
 - 1221, 1229, 1232
- \bbl@write
 - 604, 608, 671,
 - 1209, 1221, 1231
- \bblname 1130, 1152, 1219
- \begin 8
- \bgroup
 - 89, 1748, 1816,
 - 1879, 1933, 1976
- \bi@ 32
- \bib .. 2, 4, 6, 9, 10,
 - 27, 40, 44, 45,
 - 47, 48, 53, 67, 558
- \bib' 36
- \bib'bar 19
- \bib'date 39
- \bib'given 70, 71
- \bib'initials 71
- \bib'jr 71
- \bib'language 38
- \bib'pages 19
- \bib'surname 71
- \bib* 27, 29
- \bib@append 19,
 - 20, 234, 247, 2424
- \bib@author 40
- \bib@bar 19
- \bib@cite . 40, 714, 988
- \bib@cite@a .. 993, 996
- \bib@cite@check ..
 - 1008, 1011
- \bib@citeyear 2110, 2114
- \bib@day
 - 933, 940, 941, 2567
- \bib@dbfile
 - .. 1153, 1156,
 - 1159, 1162,
 - 1181, 1184, 1187
- \bib@end .. 34, 731, 748
- \bib@exec .. 10, 28,
 - 32, 578, 586, 591
- \bib@field@patches
 - 710, 737,
 - 892, 2407, 2421
- \bib@label ... 83, 107
- \bib@label@year ..
 - .. 1002, 1035,
 - 1049, 1056,
 - 1063, 2773, 2884
- \bib@language
 - 38, 954,
 - 956, 959, 961,
 - 962, 965–967, 981
- \bib@language@fixup
 - 929, 951
- \bib@mklab .. 494, 510
- \bib@month
 - 932, 939, 942,
 - 2559, 2571, 2735
- \bib@monthname ...
 - . 2566, 2570, 2734
- \bib@parsedate ...
 - 919, 922, 934
- \bib@parsedate@a .
 - 935, 937
- \bib@print
 - .. 28, 33, 578, 705

- \bib@print@inner 733, 2353, 2372, 2398, 2452
- \bib@reprint 2401, 2403
- \bib@reset 842
- \bib@resolve@xrefs 709, 736, 756, 2406, 2420
- \bib@selectlanguage 711, 964
- \bib@start 83, 107, 706, 744
- \bib@store 28, 591, 700
- \bib@translation 2415, 2417
- \bib@year . 40, 917, 926, 931, 938, 943, 2109, 2112, 2563, 2577, 2763, 2887, 2889
- \BibAbbrevWarning 790, 825, 881
- bibchapter environment 1084
- \bibchapter 1097
- \bible 8, 9, 43, 61, 1078, 1081, 1576, 1580
- \bible@ . 1571, 1582
- \bible@b . 1587, 1591
- \bible@write 994, 1069
- \bibdata 7, 8
- \bibdefer@list 693, 697, 1140, 1141
- bibdiv environment 1094
- \bibdiv 1100
- \BibField . 6, 173, 2887
- \BibItem . 563, 566, 569
- \bibitem 8
- \BibLabel 489, 545, 549, 2792
- \biblanguage@default 954, 981
- \biblanguage@English 981
- \bibliography . . . 7, 8
- \bibliographystyle 7, 102, 2622
- biblist environment 25, 473
- \biblist 1101
- biblist* environment 479
- \bibname 43, 1082, 1084
- \bibquotes . 2, 5, 2473
- bibsection environment 1089
- \bibsection 1095
- \BibSelect 1124, 1127, 1130
- \bibselect 27, 31, 32, 55, 1121, 1148, 1149
- \bibselect* 29, 55
- \bibselect@msg . . 1133
- \BibSpec . . 2–7, 19, 225
- \bbspec@scan 18, 19, 228, 233
- \bbspec@scan@error 242, 246
- \BibSpecAlias 21, 293, 449–457
- \bibstyle . . 7, 8, 2626
- bibtex-style option 106
- \bibtex@style 2623, 2626, 2630, 2643, 2681, 2745
- book class 44
- \bracket@stack 2458, 2461, 2466, 2471
- bst 1
- \bud 1692
- \bysame . . 6, 2319, 2320
- C**
- \c 1650
- \c@bib@env 521, 526, 529
- \calc@alpha@suffix . . 2041, 2123, 2770
- \calc@author@part . . 2039, 2052, 2652
- \CardinalNumber . 2597
- \CardinalNumeric 6, 2586, 2597
- \catcode 27
- \cdprime 1691
- \ch@ck 1223
- \check@mathfonts . . 23
- \chomp@ 95, 100
- \citation . . . 7–9, 1385
- citation-order option 82, 104
- cite package 50
- \cite 7–9, 41, 48, 49, 51, 54, 1241
- \cite@a . . 52, 1325, 1326, 1431, 2797, 2807, 2826, 2834, 2840
- \cite@b . 52, 1336, 1343
- \cite@bc 52, 1344, 1346
- \cite@begingroup . . . 1303, 1319, 1328, 1405, 1411
- \cite@cb . . . 1372, 1383, 1420, 1507
- \cite@cj 52, 58, 59, 1369
- \cite@cjs . . 1455, 1516
- \cite@compress 1456, 1524, 1544, 1568, 2712
- \cite@compress@a . . . 1528, 1531, 1561
- \cite@compress@b 1537, 1547
- \cite@dash 59, 1465, 1519, 1522, 1552, 1556, 1564
- \cite@dest . 1592, 1593
- \cite@dest@b 1597, 1599
- \cite@endgroup 1308, 1320, 1380, 1406, 1412, 1426
- \cite@label 986, 1002, 1028, 1029, 1031, 1046, 1049, 1056, 2772
- \cite@nobib@test 1349, 1353
- \cite@print . . 1506, 1513, 1525, 1536, 1539, 1543, 1554, 1559, 1567, 2712
- \cite@single 1242, 1266
- \cite@sort . 1459, 1469
- \cite@sort@a 1473, 1476
- \cite@sort@b 1482, 1492
- \cite@sorted@e 58, 1436, 1462

- \cite@sorted@s ... 58, 1436, 1452
 - \CiteAltPunct 50
 - \citeAltPunct 1295, 1306
 - \citeauthor
 - . 1401, 2825, 2829
 - \citeauthor 1402, 2828
 - \citedest 9, 61, 62, 1592
 - \citeform 1296
 - \citeleft 1291, 1302,
 - 1310, 1332,
 - 1382, 1414,
 - 2785, 2788, 2790
 - \citelist
 - 54, 1256, 1277,
 - 1297, 1379, 1407
 - \citemid 50, 1291, 1376
 - \citen 1394
 - \CiteNames . 2863, 2873
 - \CiteNamesFull ...
 - . 2833, 2839, 2874
 - \CitePrintUndefined
 - .. 1020, 1358,
 - 1365, 1368,
 - 1454, 1515, 2856
 - \citepunct 50, 1291,
 - 1302, 1306,
 - 1310, 2791, 2801
 - \citeright
 - .. 1291, 1311,
 - 1315, 1333,
 - 1377, 1415, 2789
 - \cites .. 1252, 1393,
 - 1396, 1398,
 - 1408, 2803, 2812
 - \cites@a 1247,
 - 1268, 1269, 1393
 - \cites@b ... 1271, 1280
 - \cites@c ... 1284, 1286
 - \cites@init .. 1309,
 - 1321, 1329, 1413
 - \cites@o ... 1244, 1267
 - \cites@oo .. 1267, 1268
 - \CITSEL 52
 - \citesel
 - 41, 48, 53, 54,
 - 61, 62, 1239, 2786
 - \citesel@author ..
 - . 1331, 2783, 2826
 - \citesel@authoryear
 - . 2784, 2786, 2834
 - \citesel@number ..
 - . 1237, 1239, 1435
 - \citesel@object ..
 - 2785,
 - 2790, 2807, 2840
 - \citesel@other ...
 - 1431, 1432
 - \citesel@update ..
 - 1234, 1388
 - \citesel@write ...
 - 1071, 1075
 - \citesel@year
 - . 1238, 2797, 2802
 - \CloseBBLFile 1220, 1228
 - \closein 1181
 - \clubpenalty . 499, 500
 - cmr font 51
 - cmsy font 5
 - collection entry type 6
 - \compare@stems ...
 - 2125, 2149
 - \copy@bibdef
 - .. 28, 55, 594,
 - 684, 1123, 1423
 - \copy@bibdef@a ...
 - 598, 601, 647
 - \count 13, 1215
 - \count@ 101
 - \count@chars 1999, 2008
 - \cprime 1690
 - \CPU@normal . 1368, 1515
 - \CPU@sort .. 1451, 1454
 - \cs 67, 68
 - \CurLine 46, 47, 1187,
 - 1188, 1194, 1205
 - \current@bibfield
 - 21, 255,
 - 283, 2315, 2332,
 - 2439, 3129, 3134
 - \current@citekey .
 - 742, 746, 753, 2142
 - \current@primary 82,
 - 93, 106, 883,
 - 889, 895, 898,
 - 902, 2037, 2302,
 - 2305, 2312,
 - 2329, 2436, 2768
 - \current@stem
 - 82, 106,
 - 2024, 2027,
 - 2031, 2032,
 - 2044, 2125,
 - 2760, 2761,
 - 2768, 2769, 2772
 - \current@year
 - 85, 2026, 2127,
 - 2762, 2763, 2774
 - \CurrentBib
 - 544, 546, 750,
 - 989, 1067, 1078,
 - 2633, 2646, 2793
 - \CurrentBibType 29, 580
 - \cydot 1693
- ## D
- \d 1651
 - \DashPages ... 5, 2515
 - \dateLANGUAGE 40
 - \Dbar 1687
 - \dbar 1687
 - \DeclareBooleanOption
 - 127–132, 142–146
 - \DeclareExclusiveOptions
 - 125, 126, 139–141
 - \DeclareNameAccent
 - . 1639, 1642–1666
 - \DeclareNameSymbol
 - . 1668, 1676–1694
 - \DeclareOption ... 133
 - \DeclareRobustCommand
 - 18, 1297, 1365,
 - 1393, 2796,
 - 2799, 2806,
 - 2809, 2825,
 - 2828, 2831, 2837
 - \def 21, 53, 54
 - \defer@bibdef
 - ... 596, 686, 1178
 - \deferredquotes ..
 - .. 2245, 2475,
 - 2476, 2705, 2707
 - \deferredquoteslogical
 - 2234, 2706
 - \DefineAdditiveKey
 - 177–184
 - \DefineDummyKey .. 222

- `\DefineJournal` 2, 15,
 35, 44, 1109, 2718
`\DefineName`
 . 2, 34, 44, 47,
 1106, 2447, 2448
`\DefinePublisher` 15,
 35, 44, 1115, 2726
`\DefineSimpleKey` .
 185–221, 223, 224
`\DH` 1681
`\dh` 1681
`\DJ` 1682
`\dj` 1682
`\do` 12, 17, 64
`docstrip` 116
`\dospecials` 13
`\DSK@append` 840
`\DSK@def` 839
`\dudot` 1660
`\DuplicateBibKeyWarning`
 1013, 1064
- E**
- `\edef` 67
`\editiontext`
 . 5, 6, 2590, 2596
`editor field` 20
`\emph` 3013,
 3029, 3038,
 3040, 3081,
 3083, 3100,
 3102, 3117, 3118
`\empty@cite` . 1348, 1352
`\EmptyNameWarning`
 1773, 1823
`\EmptyPrimaryWarning`
 2303, 2308
`\endbibchapter` .. 1097
`\endbibdiv` 1104
`\endbiblist` 1103
`\endbibsection` .. 1095
`\endinput` 47, 116
`\endlist` 507
environments:
 bibchapter ... 1084
 bibdiv 1094
 biblist ... 25, 473
 biblist* 479
 bibsection ... 1089
 tabbing 67
- thebibliography**
 1099
`\eprint` 317, 379, 422,
 440, 2621, 2906,
 2952, 2973, 2991
`\eprintpages` .. 314,
 334, 2522, 2903
`\erase@field` ... 20,
 21, 278, 2335,
 2442, 3131, 3136
`\errmessage` 251
`\etalchar` 51,
 65, 66, 1665,
 2091, 2096, 2106
`\etaltext` 65,
 999, 1016,
 1666, 2036,
 2095, 2191,
 2447, 2448,
 2449, 2767, 2868
`\ExecuteOptions` .. 147
`export-bibtex pack-`
 age 29
`\extr@cite` .. 1051,
 1358, 1363,
 1385, 1392,
 1610, 1620, 1622
`\extract@initial` .
 1951,
 1956, 1973, 1988
`\extract@initial@a`
 ... 76, 1958, 1960
`\extract@initial@b`
 1965, 1972
`\extract@initial@c`
 1967, 1975
`\extract@initial@d`
 1982, 1984
`\extract@initials`
 . 1777, 1824, 2703
`\extract@surnames`
 . 2073, 2080, 2099
- F**
- `\fld@elt` 175,
 180–184, 2342,
 2500, 2504, 2511
`\font` 23
`\fontseries` 1366
`\fontshape` 1366
- `\foo` 67
`\frenchspacing` ...
 . 90, 102, 113, 501
`\fsa@n` 76
`\fullcite` .. 1399, 2831
`\fullocite` . 1400, 2837
`\futurelet` 71
- G**
- `\G@refundefinedtrue`
 1355
`\g@undef` . 58, 651, 1140
`\generate@alphalabel`
 . 2028, 2638, 2656
`\generate@label` ..
 712, 985,
 2638, 2656, 2757
`\get@current@properties`
 ... 264, 281, 2291
`\get@namepart` 71, 72,
 76, 1764, 1767,
 1770, 1781, 2074
`\get@namepart@a` ..
 72, 76,
 1784, 1786, 1798
`\get@namepart@b` ..
 72, 76, 1788, 1790
`\get@namepart@c` ..
 ... 77, 1792, 1797
`\get@namepart@d` ..
 ... 71, 1794, 1800
`\get@namepart@e` ..
 1802, 1804
`\get@namepart@f` ..
 73, 77, 1812, 1815
`\get@nth@property` 283
`\get@numberof` .. 72,
 2038, 2081, 2177
`\get@property` 806
`getwidth package` ... 16
- H**
- `\H` 1652
`har.bst` 7
`\hbox` 2321
`\hfil` 515
`hij.bib` 7
`\hrulefill` 2321
`\Hy@backout` 1612
`\hyper@link` 1620

- `\hyper@anchorend` . . . 546, 1630, 2793
- `\hyper@anchorstart` . . . 546, 1629, 2793
- hyperref package 61, 63
- `\hyph@to@space` 2062, 2075
- `\hyphenpenalty` 30
- I**
- `\i` . . 66, 68, 1676, 1731
- `\if@tempswa` . . . 595, 605, 1043, 2131
- `\ifBR@verbose` 1606, 1609
- `\IfEmptyBibField` 6, 18, 174, 2255, 2409, 2425, 2426, 2513, 2523, 2542, 2545, 2548, 2687, 2886, 2887
- `\IfFileExists` 11, 153, 156, 1155, 1158, 1161
- `\ifin@` 657, 862, 1021, 1305, 1996, 2350, 2361
- ifoption package . . . 14
- `\ifsame@stems` 2126, 2148
- `\ifx` 12
- `\in@` 656, 861, 1020, 1304, 1995, 2349, 2360
- `\ini@hyphen` 1826, 1843, 1849, 1880, 1882, 1992, 1993
- initials option 71, 104
- `\InnerCite` . . 1282, 1287, 1312, 1325, 1416, 2816
- inputenc package . . . 67
- `\interlinepenalty` 498
- inverted attribute . . 71
- isbn field 6
- `\issuetext` . 5, 313, 333, 2514, 2902
- `\itemindent` 553
- J**
- `\j` . . 66, 68, 1677, 1732
- journal field 35
- K**
- `\k` 1653
- `\keep@last@digit` 2608, 2612
- `\kern` 715, 2322, 2385, 2426
- klm.bib 7
- L**
- `\L` 1683
- `\l` 1683
- label field 82, 106
- `\labelsep` 488, 531, 555
- `\labelwidth` 529, 530, 549–551
- `\languagename` 965, 981
- `\lasp` 1688
- `\lastkern` 257, 2227, 2235, 2247, 2385
- `\lastpenalty` 2228, 2230
- `\lc@do` 1713, 1718
- `\lc@edef` 1709, 2153, 2154
- `\lccode` 29
- `\lcsymbol` 65
- `\leftmargin` 530, 531, 550, 552, 553, 555
- `\let` 21, 53, 54
- `\lfhook` 1661
- `\linebreak` 4
- `\list` 489
- lite option 16
- `\lncan@a` . . 2766, 2782
- logical-quotes option 5, 104
- `\lowercase` 29, 637, 784, 821, 961, 1714
- ltoutenc.dtx 67
- M**
- `\m@ne` 254, 975
- `\m@th` 1318
- `\macrotext` . 52, 55, 573
- `\makelabel` 494
- `\mathchar` 30
- `\mathcode` 28, 33
- mathscinet package 16, 65, 66
- `\mathsurround` 51
- `\mdash` 7, 2480
- `\mddefault` 1366
- `\mdseries` 1318
- `\meaning` 20, 52
- `\MessageBreak` 106, 1253, 1262
- `\MessageBreakNS` 106, 1067
- `\modify@xref@fields` 613, 631
- `\ModifyBibLabel` 2136, 2163
- `\month@` 101
- `\moo` 19
- `\MR` 2490
- `\MRhref` 2494, 2498
- `\MultipleBibSelectWarning` 1144, 1146
- `\MultipleCiteKeyWarning` 1242, 1250
- N**
- name entry type 2
- `\name` . . 82, 106, 175, 177–179, 771, 787, 791, 797, 814, 2035, 2038, 2079, 2089, 2090, 2094, 2097, 2272, 2766, 2864, 2875
- `\name@split` 71, 1763, 2286
- `\name@split@finish` 1770, 1772
- `\name@split@given` 71, 1764, 1766
- `\name@split@jr` 1767, 1769
- `\NameCheck` 1775
- `\ndash` 7, 99, 2480, 2516, 2521
- `\NeedsTeXFormat` 2, 3003
- `\newcount` . . 2120, 2166
- `\newcounter` . . 472, 517
- `\newdimen` 509
- `\newif` 2148

- \newlabel 521
- \newread 1153
- \NG 1684
- \ng 1684
- \NoBibDBFile 1165, 1172
- \nobreak 75
- \nobreakspace 1827, 2567
- nobysame option ... 128
- \nocite 55, 1403
- non-compressed-cites
option 105
- non-sorted-cites op-
tion 15, 105
- \NonNumericCiteWarning
..... 1438, 1442
- \nopunct ... 5, 122,
1302, 1521, 2409
- \normal@bibdef . 28,
577, 579, 592, 1142
- \NormalCatcodes ... 37
- \normalfont 486
- \normalize@edef ..
..... 1695,
1831, 2037, 2768
- \number
30, 2246, 2567,
2592, 2598, 2608
- \numeric@refs . 704,
2640, 2658, 2787
- O**
- \O 1680
- \o 1680
- \ObsoleteCiteOptionWarning
..... 1246, 1260
- \ocite 1394,
1397, 1400,
2806, 2817–2820
- \ocitelist 2815
- \ocites . 51, 1398, 2809
- \OE 66, 1679
- \oe 66, 1679
- \on@line 14
- \OpenBBLFile 1136, 1212
- \openin 1156, 1159, 1162
- \or 2200, 2202,
2572–2574,
2609, 2736–2738
- organization field .. 2
- \OtherCite . 1416, 1431
- \othercitelist ...
..... 1407, 1410
- \othercites . 1403, 1404
- \output@inner@xref@
..... 620, 655
- \output@xref@
619, 639, 643, 658
- \output@xref@a ...
..... 614–618, 635
- \overshadebox ... 1618
- P**
- \PackageError 105
- \PackageInfo 1610
- \PackageWarning .. 104
- \PackageWarningNoLine
..... 4
- \parenthesize
..... 4, 5, 97,
318, 358, 380,
424, 442, 2454,
2907, 2931,
2953, 2975,
2993, 3015, 3031
- partial field 97
- pcatcode package ... 16
- place field 2
- \Plural 6, 88,
90, 2251, 2330,
2333, 2338,
2445, 3130, 3135
- \polhk 1663
- \pop@bracket 2455, 2458
- \pop@bracket@a ...
..... 2466, 2469
- \PopCatcodes 2756, 3000
- \pp@scan ... 2516, 2519
- \pp@scan@a . 2516, 2521
- \prev@cite 58,
59, 1466, 1501,
1509, 1536,
1554, 1557,
1558, 1565, 1566
- \prev@cite@cb
... 59, 1502, 1553
- \prev@citekey
... 743, 753, 2136
- \prev@names
... 1322, 2853, 2855
- \previous@primary
.. 93, 884, 889,
2312, 2329, 2436
- \previous@stem 2023,
2031, 2125, 2760
- \previous@year . 85,
2025, 2127, 2762
- \prime 5
- \print@backrefs ..
..... 1614, 2661
- \print@citenames .
..... 2833,
2839, 2859, 2873
- \print@contribution
.. 2343–2346, 2348
- \print@date .. 2393,
2556, 2557,
2558, 2578, 2885
- \print@edition ...
..... 2580, 2583
- \print@month@day .
..... 2561, 2565
- \print@one@dash ..
..... 1520, 1552
- \print@partial ...
.. 2451, 2505–2508
- \print@primary ...
896, 899, 903, 2305
- \PrintAuthors .. 4,
5, 304, 367,
433, 465, 903,
2311, 2893,
2940, 2984,
3007, 3059, 3116
- \PrintBackRefs ...
.. 750, 1617, 2661
- \PrintBook ... 373,
2397, 2947, 3065
- \PrintCiteNames ..
..... 2783,
2785, 2847, 2850
- \PrintCNY .. 2784, 2846
- \PrintConference .
..... 372,
2371, 2946, 3064
- \PrintConferenceDetails
.. 390, 2374, 3077
- \PrintConferenceDetails@
.. 2378, 2381, 2384

- \PrintContributions
 - 308, 329,
 - 350, 371, 400,
 - 416, 2340, 2898,
 - 2924, 2945,
 - 2968, 3011,
 - 3028, 3045,
 - 3063, 3088, 3103
- \PrintDate 7,
 - 2546, 2551,
 - 2556, 2884,
 - 2894, 2917,
 - 2941, 2963, 2985
- \PrintDateB ... 356,
 - 375, 406, 421,
 - 439, 2557,
 - 3017, 3033,
 - 3050, 3068,
 - 3093, 3108, 3122
- \PrintDatePosted .
 - 2549, 2578
- \PrintDatePV
 - ... 312, 332, 2544
- \PrintDOI 316, 378,
 - 2540, 2905, 2951
- \PrintEdition
 - 5, 6, 347, 397,
 - 415, 2579, 2921,
 - 2967, 3049, 3092
- \PrintEditorsA ...
 - 5, 6, 92,
 - 93, 899, 2328, 3128
- \PrintEditorsB ...
 - 6, 348,
 - 398, 2337, 2922
- \PrintEditorsC 6, 2325
- \PrinteSeries@a . 2176
- \PrintISBNs 2510
- \PrintNameList 339,
 - 2324, 3043,
 - 3044, 3086, 3087
- \PrintNames 92,
 - 2268, 2316,
 - 2324, 2326,
 - 2333, 2338,
 - 2433, 2440,
 - 2445, 3130, 3135
- \PrintNames@a 2269, 2271
- \PrintPartials 309,
 - 2502, 2899, 3012
- \PrintPrimary .. 5,
 - 343, 411, 2301,
 - 2423, 2916,
 - 2962, 3037, 3099
- \PrintReprint . 320,
 - 360, 382, 426,
 - 444, 2400, 2909,
 - 2933, 2955,
 - 2977, 2995,
 - 3020, 3054,
 - 3070, 3110, 3124
- \PrintReviews .. 6,
 - 323, 363, 385,
 - 429, 447, 2499,
 - 2912, 2936,
 - 2958, 2980, 2998
- \PrintSeries 86, 88,
 - 92, 2167, 2272,
 - 2341, 2500,
 - 2503, 2511,
 - 2816, 2864, 2875
- \PrintSeries@a ...
 - ... 88, 2174, 2176
- \PrintThesisType .
 - 436,
 - 2525, 2988, 3119
- \PrintTranslation
 - 319,
 - 359, 381, 425,
 - 443, 2414, 2908,
 - 2932, 2954,
 - 2976, 2994,
 - 3019, 3053,
 - 3069, 3109, 3123
- \PrintTranslatorsA
 - 6,
 - 93, 896, 2435, 3133
- \PrintTranslatorsB
 - 6, 2444
- \PrintTranslatorsC
 - 6, 349,
 - 399, 2432, 2923
- \PrintYear 7, 101, 2577
- proceedings entry
 - type 6
- \process@bare@dot
 - 1914, 1924
- \process@citelist
 - 1313,
 - 1450, 2639,
 - 2657, 2715, 2795
- \process@citelist@sorted
 - 1434, 1450
- \process@citelist@unsorted
 - 1439,
 - 1447, 2639,
 - 2657, 2715, 2795
- \process@dots
 - 75, 79, 1835, 1897
- \process@hyphens 75,
 - 77, 79, 1834, 1844
- \process@names ...
 - 1836, 1948
- \process@one@dot .
 - 78, 1900,
 - 1908, 1930, 1945
- \process@one@dot@a
 - 1910, 1912
- \process@one@dot@b
 - 1917, 1929
- \process@one@dot@c
 - 1919, 1932
- \process@one@dot@d
 - . 1927, 1939, 1941
- \process@one@hyphen
 - 76, 1847,
 - 1854, 1876, 1891
- \process@one@hyphen@a
 - 78, 79, 1856, 1858
- \process@one@hyphen@b
 - . 1864, 1867, 1875
- \process@one@hyphen@c
 - 1869, 1878
- \process@one@hyphen@d
 - ... 76, 1885, 1887
- \process@xrefs 603, 629
- \ProcessExclusiveOptions
 - 151
- \ProcessOptions .. 149
- \prop ... 273, 275, 2294
- \prop@reset 286
- \protected@edef ..
 - 1000,
 - 1054, 1086, 1091
- \ProvidesFile ... 1176
- \ProvidesPackage 3, 3004
- publisher field 35
- \purge@accent 1729, 1760

- `\purge@edef`
 . [1725](#), [2151](#), [2152](#)
`\purge@edef@`
 . [1735](#), [1740](#),
 [1750](#), [1751](#), [1755](#)
`\purge@edef@a`
 . [69](#), [1742](#), [1744](#)
`\push@bracket` [2455](#), [2458](#)
`\PushCatcodes` [37](#)
- R**
- `\r` [1654](#)
`\rasp` [1689](#)
`\read` [1187](#)
`\ReadBibData` [1137](#), [1154](#)
`\ReadBibData@a` ...
 . [1166](#), [1170](#), [1175](#)
`\ReadBibLoop`
 . [1179](#), [1183](#), [1197](#)
`\ReadBibLoop@a` ...
 [1188](#), [1192](#)
`\ReadBibLoop@b` ...
 [1199](#), [1203](#)
`\ReadBibLoop@e` .. [1192](#)
`\refname` [43](#), [1083](#), [1089](#)
`\RequirePackage` [3](#),
 [12](#), [36](#), [124](#), [154](#),
 [157](#), [167](#), [170](#),
 [171](#), [2754](#), [3005](#)
`\reset@nth@property` [809](#)
`\resetbiblist`
 . [527](#), [548](#), [1101](#)
`\ResetCapSFCodes` .
 [502](#), [534](#)
`\resolve@inner` [2359](#),
 [2372](#), [2398](#),
 [2401](#), [2415](#), [2452](#)
`\restore@labelwidth`
 [490](#), [525](#)
`\RestrictedSetKeys`
 [10](#), [20](#),
 [285](#), [585](#), [607](#),
 [631](#), [2352](#), [2362](#)
`review field` [4](#), [6](#)
`rkeyval package` ...
 . [4](#), [10](#), [20](#), [76](#)
`\rkvd@DSAK` [20](#)
`\rkvd@setter` .. [857](#), [877](#)
`\rkvIfAdditive` ... [263](#)
`\rkvIfEmpty` ... [17](#), [174](#)
- S**
- `\safe@set` [81](#),
 [1470](#), [1477](#),
 [1493](#), [1527](#),
 [1534](#), [1550](#), [2109](#)
`\same@stemsfalse` [2158](#)
`\same@stemstrue` . [2156](#)
`\sameauthors`
 . [5](#), [6](#), [2313](#),
 [2319](#), [2330](#), [2437](#)
`\save@labelwidth` .
 [505](#), [518](#)
`\save@primary` [752](#), [885](#)
`\selbibdef@a` . [676](#), [678](#)
`\selbibdef@b` . [680](#), [683](#)
`\select@auxlanguage`
 . [265](#), [272](#), [2292](#)
`\selective@bibdef`
 [28](#),
 [55](#), [675](#), [1126](#), [1423](#)
`\selectlanguage` ..
 . [38](#), [40](#), [275](#), [967](#)
`\SentenceSpace` [323](#),
 [363](#), [385](#), [429](#),
 [447](#), [1615](#), [2620](#),
 [2912](#), [2936](#),
 [2958](#), [2980](#), [2998](#)
`\series@add` . [2185](#), [2189](#)
`\series@add@...` ... [88](#)
`\series@add@a` [2169](#), [2201](#)
`\series@add@b` [2170](#), [2204](#)
`\series@add@c`
 . [2171](#), [2206](#), [2213](#)
`\series@add@d` [2172](#), [2210](#)
`\series@add@e` [2173](#), [2195](#)
`\series@index`
 [21](#), [254](#),
 [283](#), [770](#), [782](#),
 [809](#), [2166](#), [2184](#),
 [2190](#), [2192](#),
 [2197](#), [2199](#),
 [2209](#), [2212](#), [2351](#)
- `\series@total`
 . [2178](#), [2179](#),
 [2203](#), [2209](#), [2212](#)
`\set` [614](#)–[624](#)
`\set@firstname` ...
 . [2269](#), [2279](#), [2843](#)
`\set@name` ... [2280](#),
 [2283](#), [2285](#), [2844](#)
`\set@name@a`
 . [92](#), [2287](#), [2289](#)
`\set@namea` [2289](#)
`\set@othername` [2269](#),
 [2274](#)–[2276](#), [2282](#)
`\setbib@` [28](#), [29](#), [31](#)–[33](#)
`\setbib@@`
 [707](#), [716](#), [721](#),
 [724](#), [727](#), [730](#), [2429](#)
`\setbib@article` .. [716](#)
`\setbib@book` [2411](#)
`\setbib@conference` [2372](#)
`\setbib@contribution`
 [2353](#)
`\setbib@incollection`
 [458](#), [721](#), [724](#), [727](#)
`\setbib@innerbook` [2398](#)
`\setbib@inproceedings`
 [461](#)
`\setbib@misc` [583](#)
`\setbib@nameBE` ...
 [2258](#), [2295](#)
`\setbib@nameinverted`
 [2263](#), [2844](#)
`\setbib@nameLE` ...
 . [2253](#), [2280](#), [2283](#)
`\setbib@partial` . [2452](#)
`\setbib@TYPE` [18](#)
`\setcounter` [474](#)
`\settowidth` .. [511](#), [549](#)
`\sfcode` [90](#), [114](#)–[119](#),
 [537](#), [2237](#), [2395](#)
`\shade@cite` .. [1618](#),
 [1620](#), [1625](#), [1628](#)
`shaderef package` ... [63](#)
`short-journals op-`
 tion [15](#), [105](#)
`short-months option` [105](#)
`short-publishers op-`
 tion [15](#), [105](#)

- shortalphabetic op-
 tion 103
 showkeys package 49, 63
 \SingularPlural ..
 6, 88, 90,
 2180, 2182, 2251
 \sixt@on ... 1217, 1223
 \SK@ ... 992, 1622, 1635
 \SK@@label .. 992, 1634
 \SK@@ref 1622
 \soft 1664
 sorted option 15
 \space 76
 \spacefactor .. 110,
 122, 2237, 2239,
 2395, 2491, 2620
 \SS 1685
 \ss 1685
 \star@ 665,
 1267, 1325,
 2797, 2807,
 2826, 2834, 2840
 \star@bibdef ... 28,
 562, 590, 647, 1178
 \stepcounter 485
 \string 67
 \strip@prefix .. 20, 52
 \SubEtal 2193, 2195, 2449
 \surround@names ..
 2064, 2066
 \SwapBreak 89,
 259, 2170–
 2173, 2222,
 2338, 2392, 2445
 \symbol 65
- T**
- \t 1655
 tabbing environment 67
 \textbf 4, 311, 331, 2901
 textcmds package ..
 7, 16, 98
 \textemdash ... 7, 2480
 \textendash 7, 1521, 2481
 \textfont 23
 \textit 305, 307, 328,
 344, 346, 368,
 370, 412, 414,
 434, 435, 2895,
- 2897, 2918,
 2920, 2942,
 2944, 2964,
 2966, 2986, 2987
 \textquotedblleft 2474
 \textquotedblright 2477
 \textsc 87
 \textsubscript 5
 \textsuperscript .. 5
 \texttt 20
 \TH 1686
 \th 1686
 \the 29, 32, 33
 \thebib 546,
 549, 990, 2633,
 2635, 2646, 2648
 thebibliography envi-
 ronment ... 1099
 \thesis@type 2526, 2529
 \thinspace 2321
 \thr@@ 2203, 2601
 \toks@ 18, 72
 \tprime 5
 \TrailingHyphenWarning
 1863, 1894
 transition field 4, 18, 34
 \tsub 5
 \tsup 5
 \ttfamily 1366
 \tw@ 2134, 2192
 \typeout 161, 1132, 1134
- U**
- \u 1656
 \uarc 1659
 \udot 1662
 \UndefinedCiteWarning
 1356, 1361
 \unkern 2227
 \unpenalty 2231
 \unskip .. 75, 1843,
 2226, 2491, 2578
 \uparrow 1615
 \upn 97, 2455, 2457, 2470
 upref package 97
 \upshape 1318
 \url 18, 2621
 \use@accent
 . 1699, 1705, 1711
- \utilde 1658
- V**
- \v 1657
 \vdef ... 53, 1272, 1337
 \voltext 5,
 352, 402, 2513,
 2926, 3042, 3085
- W**
- \widowpenalty 500
 \wrap@accent
 ... 69, 1698, 1706
 wrapper macros, defi-
 nition 12
- X**
- xref field 36
 \xref@add@toks 839, 854
 \xref@append . 840, 875
 \xref@check@a
 758–760, 764
 \xref@check@aa 771, 781
 \xref@check@ab 793, 800
 \xref@check@b
 761, 762, 817
 \xref@check@c
 . 36, 757, 828, 833
 \xref@ID . 843–845, 847
 \XRefWarning
 ... 845, 882, 2365
 \XrefWarning 882
 xyz.aux 7
 xyz.bbl 8
 xyz.tex 7
- Y**
- y2k option . 81, 105, 127
 \ycite ... 54, 1395,
 1397, 1401,
 1402, 2796, 2829
 \ycites 1396, 2799
 \year@short
 . 2107, 2114, 2744
- Z**
- \Z 539
 \zap@space . 1274, 1339