

The `amsrefs` package

Michael Downes and David M. Jones
American Mathematical Society

Version 2.03, 2007/10/22

Contents

1	Introduction	2
2	Package options	3
3	More about the <code>\bib</code> command	3
3.1	Field names for the <code>\bib</code> command	3
3.2	Bibliography entry types	4
4	Customizing the bibliography style	4
5	Miscellaneous commands provided by the <code>amsrefs</code> package	5
6	Implementation	8
6.1	Overview	8
6.1.1	Normal L ^A T _E X processing of cites	8
6.2	How cites are processed by <code>amsrefs</code>	9
6.3	Data structures	10
6.4	Preliminaries	11
6.5	Utilities	12
6.6	Declaring package options	14
6.6.1	The <code>?</code> option	16
6.7	Loading auxiliary packages	16
6.7.1	The <code>lite</code> option	16
6.8	Key-value setup	16
6.8.1	Standard field names (the <code>bib</code> group)	16
6.8.2	Auxiliary properties (the <code>prop</code> group)	17
6.9	Bibliography type specifications	18
6.10	The standard bibliography types	20
6.11	The <code>\biblist</code> environment	23
6.12	Processing bibliography entries	25
6.12.1	<code>\@bibdef</code> Implementations	26
6.12.2	<code>\bib@exec</code> Implementations	29
6.12.3	Resolving cross-references	31
6.12.4	Bib field preprocessing	34
6.12.5	Date setup	35
6.12.6	Language setup	36
6.12.7	Citation label setup	36
6.12.8	Printing the bibliography	39
6.13	Name, journal and publisher abbreviations	40
6.14	Processing <code>.ltb</code> files	41
6.15	Citation processing	43
6.15.1	The <code>\citesel</code> structure	43
6.15.2	The basic <code>\cite</code> command	44
6.15.3	Fancier <code>\cite</code> commands	49
6.15.4	The <code>\nocite</code> command	49
6.15.5	Citation sorting	50

6.15.6	Range compression	52
6.15.7	Munging the <code>.aux</code> file	54
6.15.8	Back references	56
6.15.9	<code>hyperref</code> , <code>showkeys</code> and <code>shaderef</code> support . . .	56
6.16	Lexical structure of names	57
6.16.1	Text accents	57
6.16.2	Text symbols	58
6.16.3	<code>\edef</code> -like macros for names	59
6.17	Name parsing	61
6.18	Extracting initials	65
6.18.1	The algorithm	66
6.18.2	The implementation	66
6.19	Generating alphabetic labels	71
6.19.1	The algorithm	71
6.19.2	The implementation	72
6.20	Generating short alphabetic labels	76
6.21	Formatting series	76
6.22	Formatting names and series of names	79
6.23	The <code>partial</code> field	85
6.24	Special formatting for other fields	85
6.25	BibTeX support	89
6.26	Implementing package options	89
6.26.1	The <code>alphabetic</code> option	89
6.26.2	The <code>shortalphabetic</code> option	90
6.26.3	The <code>backrefs</code> option	90
6.26.4	The <code>citation-order</code> option	90
6.26.5	The <code>initials</code> option	91
6.26.6	The <code>logical-quotes</code> option	91
6.26.7	The <code>non-compressed-cites</code> option	91
6.26.8	The <code>non-sorted-cites</code> option	91
6.26.9	The <code>short-journals</code> option	91
6.26.10	The <code>short-publishers</code> option	92
6.26.11	The <code>short-months</code> option	92
6.26.12	The <code>y2k</code> option	92
6.26.13	The <code>bibtex-style</code> option	92
6.26.14	The <code>msc-links</code> option	92
6.26.15	The <code>author-year</code> option	93
6.27	The <code>amsbst</code> package	99

1 Introduction

The `amsrefs` package is a L^AT_EX package for bibliographies that provides an archival data format similar to the format of BibTeX database files, but adapted to make direct processing by L^AT_EX 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 L^AT_EX document. This takes on its greatest significance when

archiving documents in L^AT_EX form or transmitting them to another user (such as a publisher).

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

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

2 Package options

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

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

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

3 More about the `\bib` command

3.1 Field names for the `\bib` command

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

fulljournal Used internally by `\DefineJournal`.

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

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

The following fields are included for backwards compatibility:

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

place A synonym for `address`. In earlier versions of `amsrefs`, `place` was preferred and `address` was considered as an alias for `place`. However, this seemed like a gratuitous incompatibility with BibT_EX to me, so I have 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.

¹There is one drawback: If you don’t want to get the prime symbol for `\tprime` from the `cmsy` font, you will need to redefine `\tprime` in some suitable way.

`\PrintEditorsB` This is similar to `\PrintEditorsA` but puts parentheses around the entire list of editors. It's used by, for example, the `article` type to print the editors of a `proceedings` or `collection`.

`\PrintEditorsC` Similar to `\PrintEditorsA` but precedes the editors by `Edited by`. It's used when the editors should be treated as subsidiary contributors, rather than the primary contributor.

`\PrintTranslatorsA` This is similar to `\PrintEditorsA` but adds `(trans.)` following the translators.

`\PrintTranslatorsB` This is similar to `\PrintEditorsB`. It's not currently used, but is provided for symmetry.

`\PrintTranslatorsC` Similar to `\PrintEditorsC` but precedes the translators by `Translated by`.

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

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

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

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

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

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

`\IfEmptyBibField` If one writes

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

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

`\PrintEdition` If a bibliography entry has

```
edition={2}
```

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

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

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

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

et cetera ... [mjd,2002-01-03] See the `.dtx` files for further possibilities that I have not managed to get properly documented yet!

6 Implementation

6.1 Overview

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

6.1.1 Normal \LaTeX processing of cites

First \LaTeX pass Various commands are written to the `.aux` file that are mostly used by 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@moo` contains the corresponding citation label, but since this is the first pass, the label won't be known yet, so \LaTeX emits an 'Undefined citation' warning and prints a placeholder (i.e., `???`) instead of the citation label.
2. A `\bibliographystyle{har}` command writes one line to the `.aux` file: `\bibstyle{har}`. This indicates to BibTeX that it should use `har.bst` to determine the style for sorting and formatting the bibliography items.
3. A `\bibliography{hij,klm,...}` command writes one line to the `.aux` file: `\bibdata{hij,klm,...}`. This indicates to BibTeX that it should look in `hij.bib`, `klm.bib`, ... for bibliographic data. The `\bibliography` also tries to input the `.bbl` file, but on the first pass it won't exist yet.

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

BibTeX pass For a document named `xyz.tex`, the command `bibtex xyz` is used to invoke BibTeX . It looks in `xyz.aux` to find the citation information written there by \LaTeX . For each `\citation` line, BibTeX searches for a corresponding entry in the specified `.bib` files and formats it. The entire list is then

sorted in whatever way dictated by the bibliography style, and written out to the file `xyz.bbl`. This normally produces entries that look something like:

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

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

1. Instances of `\cite` still print question marks.
2. The `\bibliography` command causes \LaTeX to input `xyz.bbl` and typeset its contents.
3. A `\bibitem{moo}` command writes one line to the `.aux` file: `\bibcite{moo}{9}`, where 9 is the current item number.
4. A `\bibitem[Moody]{moo}` command writes one line to the `.aux` file: `\bibcite{moo}{Moody}`, using the supplied label instead of a number.

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

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

6.2 How cites are processed by `amsrefs`

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

- Each time an item is cited in the body of the document, a `backref` entry is added to the info of that item. The `backref` info is the current page and section location. Section location is a bit hard to get right without better support from the document class. So we provide a hook to allow it to work better when the support is there.
- When a cite occurs, if the info is undefined then a warning is issued and the info structure is created. A `\citation` command and a `\citedest` command (providing backref info) are written to the `.aux` file. Because the backref info includes page number, it has to be a non-immediate write. An undefined info structure would normally happen only on a first pass when no `.aux` file exists, or when a new cite is added. I.e., when the corresponding `\citation` command is not yet present in the `.aux` file.

- When a citation command occurs in the `.aux` file, it initializes the info structure if necessary, setting the “bib-info-present” flag to 0.
- When a `\citedest` command occurs in the `.aux` file, it initializes the info structure if necessary—but this shouldn’t happen: if the corresponding `\citation` command did not already get processed, then something is wrong. So normally, the `\citedest` command merely needs to add its backref info to the existing info structure.
- When a `\bibcite` command occurs in the `.aux` file, it will normally find that `\b@whatever` is already defined, if the bibliography occurs after all the `\cite` commands. What it must do is fill in the appropriate blank slots in the info structure set up by a previous `\citation` command.
- The `.aux` file is actually processed two times, once at the beginning of the document and once at the end. In the latter case, `\bibcite` should give a warning if the backref-list is empty, since that means there were no `\cite` commands for the given key.
- When processing the bibliography: The `\bib` command needs to check if it is using a key that is already used by another `\bib` command.

We therefore have

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

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

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

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

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

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

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

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

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

6.3 Data structures

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

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

The scanned result is to assign

```
\global\rsk@toks{%
  \set:bib'author{Miller, G.}{}%
  \set:bib'title{Eine Bemerkung zur Darstellung von Polynomen
```

```

        \{"u}ber Verb\"{a}nden}{language={german}}}%
\set:bib'journal{J. Math. Sent.}{}%
\set:bib'volume{10}{}%
\set:bib'year{1983}{}%
\set:bib'pages{26\ndash 30}{}%
}

```

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

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

6.4 Preliminaries

```
1 < *pkg >
```

Standard declaration of package name and date.

```
2 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
```

```
3 \ProvidesPackage{amsrefs}[2007/10/22 v2.03]
```

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

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

Backward handling for beta and jpa options.

```
5 \ifpackagewith{amsrefs}{beta}{%
```

```
6   \amsrefs@warning@nl{The beta option is obsolete}%
```

```
7 }{}
```

```
8 \ifpackagewith{amsrefs}{jpa}{%
```

```
9   \amsrefs@warning@nl{The jpa option is obsolete}%
```

```
10 }{}
```

```
11 \IfFileExists{url.sty}{%
```

```
12   \RequirePackage{url}\relax
```

```
13   \@gobble
```

```
14 }{%
```

```
15   \@firstofone
```

```
16 }
```

```
17 {
```

```
18   \DeclareRobustCommand{\url}[1]{%
```

```
19     \def\@tempa{#1}%
```

```
20     \texttt{\@urlsetup $\expandafter\strip@prefix\meaning\@tempa$}%
```

```
21   }%
```

```
22   \def\@urlsetup{%
```

```
23     \check@mathfonts \textfont\@ne\the\font \textfont\z@\the\font
```

```
24     \@apply\@urlfix{\do\+\do\=\do\:\do\-\do\.\do\, \do\;}%
```

```
25     \@apply\@urlbreak{\do\&\do\/\do\?}%
```

```
26   }%
```

```
27   \def\@urlbreak#1{%
```

```
28     \mathcode' #1="8000
```

```
29     \begingroup \lccode'\~=' #1 \lowercase{\endgroup \edef~}%
```

```
30     {\mathchar\number' #1\penalty\hyphenpenalty}%
```

```
31   }%
```

```
32   \def\@urlfix#1{%
```

```
33     \mathcode' #1=' #1\relax
```

```
34   }%
```

```
35 }
```

```
36 \ifundefined{NormalCatcodes}{\RequirePackage{pcatcode}\relax}{}
```

```
37 \PushCatcodes\NormalCatcodes
```

```
38 %% WARNING WARNING WARNING: Catcode of apostrophe ' is letter
```

```
39 %% throughout this file.
```

```
40 \catcode'\='11 % letter
```

6.5 Utilities

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

<code>\after@deleting@token</code>	Similar in concept to <code>\afterassignment</code> , except it deletes the next token in the stream before putting its argument back into the input. Useful for skipping past tokens during parsing.
	<pre> 41 \def\after@deleting@token#1{% 42 \afterassignment#1% 43 \let\@let@token= % Don't delete this space! 44 }</pre>
<code>\ifempty</code>	Some frequently used tests for empty arguments. Note that an argument consisting entirely of spaces (e.g., <code>\ifempty{ }</code>) counts as empty.
<code>\ifnotempty</code>	<pre> 45 \long\def\ifempty#1{\xifempty#1@@..\@nil} 46 47 \long\def\xifempty#1#2@#3#4#5\@nil{% 48 \ifx#3#4\@xp\@firstoftwo\else\@xp\@secondoftwo\fi 49 } 50 51 \long\def\ifnotempty#1{\ifempty{#1}{}}</pre>
<code>\macrotext</code>	<pre> 52 \def\macrotext{\expandafter\strip@prefix\meaning}</pre>
<code>\vdef</code>	“Verbatim” def.
	<pre> 53 \def\vdef#1#2{% 54 \def#1{#2}% 55 \edef#1{\macrotext#1}% 56 }</pre>
<code>\auto@protect</code>	Sometimes it’s convenient to render a given control sequence unexpandable for a time. <code>\auto@protect</code> provides a way to do that. ²
	An earlier version of this code read simply <code>\let#1\relax</code> but that had the disadvantage of making all <code>\auto@protected</code> macros compare equal via <code>\ifx</code> . This version allows macros to keep their identities under comparisons.
	<pre> 57 \def\auto@protect#1{\def#1{\@nx#1}}</pre>
<code>\g@undef</code>	Globally undefine a control sequence.
	<pre> 58 \def\g@undef#1{\global\let#1\relax}</pre>
<code>\@concat</code>	Concatenate onto the end of a token list. Expands everything.
	<pre> 59 \def\@concat#1#2{\edef#1{#1#2}}</pre>
<code>\add@toks@</code>	This saves a few tokens of main memory and a lot of typing.
	<pre> 60 \def\add@toks@{\addto@hook\toks@}</pre>
<code>\@lappend</code>	Append an element to a <code>\do</code> -delimited list. As long as the element to be appended (<code>#2</code>) is a single token, nothing is expanded. If it contains multiple tokens, all tokens after the first will be expanded.
	<pre> 61 \def\@lappend#1#2{% 62 \begingroup 63 \def\do{\@nx\do\@nx}% 64 \edef\@tempa{\def\@nx#1{#1\do#2}}%</pre>

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

```

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

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

y2k

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

nobysame

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

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

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

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

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

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

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

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

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

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

```
135   \@currentx
```

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

```
137   \@currname
```

```
138 }
```

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

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

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

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

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

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

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

Option for giving information about the available options:

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

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

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

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

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

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

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

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

```
149
```

```
150 \ProcessOptions\relax
```

```
151
```

```
152 \ProcessExclusiveOptions
```

```

153 \IfOption{backrefs}{%
154   \IfFileExists{backref.sty}{%
155     \RequirePackage{backref}[1999/05/30]
156   }{%
157     \amsrefs@warning@nl{The backrefs option cannot be used^^J%
158       unless the backref package is also installed.^^J%
159       (backref is part of the hyperref package)}%
160   }%
161 }{}
162
163 \IfOption{msc-links}{%
164   \IfFileExists{hyperref.sty}{%
165     \RequirePackage{hyperref}[1999/07/08]
166   }{
167     \amsrefs@warning@nl{The msc-links option cannot be used^^J%
168       unless the hyperref package is installed}%
169   }%
170 }{}

```

6.6.1 The ? option

```

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

```

6.7 Loading auxiliary packages

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

```

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

```

6.7.1 The lite option

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

```

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

```

6.8 Key-value setup

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

```

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

```

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

```

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

```

6.8.1 Standard field names (the bib group)

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

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

`\name`


```

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

```

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

```

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

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

```

233 \DefineDummyKey{bib}{transition}

```

6.8.2 Auxiliary properties (the prop group)

```

234 \DefineSimpleKey{prop}{inverted}

```

```
235 \DefineSimpleKey{prop}{language}
```

6.9 Bibliography type specifications

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

```
236 \newcommand{\BibSpec}[2]{%
237   \toks@{\emptytoks
238   \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.

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

`\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.

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

`\bibspeg@scan@error`

```
257 \def\bibspeg@scan@error{\amsrefs@error{Bad BibSpec: Expected '+'}}
```

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

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

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

```

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

```

`\select@auxlanguage`

```

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

```

`\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).

```

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

```

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

```

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

```

`\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.

```

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

```

6.10 The standard bibliography types

```

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

```

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

```

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

```

```

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

```

```

395     +{.} {} {transition}
396     +{} {} \SentenceSpace \PrintReviews} {review}
397 }
398
399 \BibSpec{conference}{%
400     +{} {} {title}
401     +{} {} \PrintConferenceDetails} {transition}
402 }
403
404 \BibSpec{innerbook}{%
405     +{,} { } {title}
406     +{.} { } {part}
407     +{:} { } {subtitle}
408     +{,} { \PrintEdition} {edition}
409     +{} { \PrintEditorsB} {editor}
410     +{,} { \PrintTranslatorsC} {translator}
411     +{,} { \PrintContributions} {contribution}
412     +{,} { } {series}
413     +{,} { \voltext} {volume}
414     +{,} { } {publisher}
415     +{,} { } {organization}
416     +{,} { } {address}
417     +{,} { \PrintDateB} {date}
418     +{.} { } {note}
419 }
420
421 \BibSpec{report}{%
422     +{} {} \PrintPrimary} {transition}
423     +{,} { \textit} {title}
424     +{.} { } {part}
425     +{:} { \textit} {subtitle}
426     +{,} { \PrintEdition} {edition}
427     +{,} { \PrintContributions} {contribution}
428     +{,} { Technical Report } {number}
429     +{,} { } {series}
430     +{,} { } {organization}
431     +{,} { } {address}
432     +{,} { \PrintDateB} {date}
433     +{,} { \eprint} {eprint}
434     +{,} { } {status}
435     +{} { \parenthesize} {language}
436     +{} { \PrintTranslation} {translation}
437     +{;} { \PrintReprint} {reprint}
438     +{.} { } {note}
439     +{.} {} {transition}
440     +{} {} \SentenceSpace \PrintReviews} {review}
441 }
442
443 \BibSpec{thesis}{%
444     +{} {} \PrintAuthors} {author}
445     +{,} { \textit} {title}
446     +{:} { \textit} {subtitle}
447     +{,} { \PrintThesisType} {type}
448     +{,} { } {organization}
449     +{,} { } {address}
450     +{,} { \PrintDateB} {date}
451     +{,} { \eprint} {eprint}
452     +{,} { } {status}
453     +{} { \parenthesize} {language}

```

```

454   +{} { \PrintTranslation}           {translation}
455   +{;} { \PrintReprint}              {reprint}
456   +{.} { }                          {note}
457   +{.} {}                            {transition}
458   +{} { \SentenceSpace \PrintReviews} {review}
459 }

460 \BibSpecAlias{periodical}{book}
461 \BibSpecAlias{collection}{book}
462 \BibSpecAlias{proceedings}{book}
463 \BibSpecAlias{manual}{book}
464 \BibSpecAlias{miscellaneous}{book}
465 \BibSpecAlias{misc}{miscellaneous}
466 \BibSpecAlias{unpublished}{book}
467 \BibSpecAlias{proceedings.article}{collection.article}
468 \BibSpecAlias{techreport}{report}

```

`\setbib@incollection`

```

469 \edef\setbib@incollection{%
470   \exp\@nx\csname setbib@collection.article\endcsname
471 }

```

`\setbib@inproceedings`

```

472 \edef\setbib@inproceedings{%
473   \exp\@nx\csname setbib@collection.article\endcsname
474 }

```

Some more entry types for implementing abbreviations.

```

475 \BibSpec{name}{%
476   +{} { \PrintAuthors}   {name}
477 }
478
479 \BibSpec{publisher}{%
480   +{,} { } {publisher}
481   +{,} { } {address}
482 }

```

6.11 The biblist environment

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

Use a standard \LaTeX counter for numbering bibliography items.

```

483 \newcounter{bib}

```

`biblist`

```

484 \newenvironment{biblist}{%
485   \setcounter{bib}\z@
486   \@biblist
487 }{%
488   \@endbiblist
489 }

```

`biblist*`

```

490 \newenvironment{biblist*}{%
491   \@biblist
492 }{%
493   \@endbiblist
494 }

```

`\@biblist`

```

495 \newcommand\@biblist[1] [] {%

```

```

496 \stepcounter{bib@env}
497 \normalfont
498 \footnotesize
499 \labelsep .5em\relax
500 \list{\BibLabel}{%
501     \restore@labelwidth
502     \@maxlabelwidth\z@
503     \@nmbrlisttrue
504     \def\@listctr{bib}%
505     \let\makelabel\@bib@mklab
506     #1\relax
507 }%
508 \sloppy

```

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

```

509 \interlinepenalty\@m
510 \clubpenalty\@M
511 \widowpenalty\clubpenalty
512 \frenchspacing
513 \ResetCapSFCodes
514 }

```

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

```

515 \def\@endbiblist{%
516     \save@labelwidth
517     \def\@noitemerr{\@latex@warning{Empty bibliography list}}%
518     \endlist
519 }

```

`\@maxlabelwidth`

```

520 \newdimen\@maxlabelwidth

```

`\bib@mklab`

```

521 \def\bib@mklab#1{%
522     \settowidth\@tempdima{#1}%
523     \ifdim \@tempdima > \@maxlabelwidth
524         \global\@maxlabelwidth\@tempdima
525     \fi
526     #1\hfil
527 }

528 \newcounter{bib@env}

```

`\save@labelwidth`

```

529 \def\save@labelwidth{%
530     \if@filesw
531         \immediate\write\@auxout{%
532             \string\newlabel{[bibenv:\the\c@bib@env]}{\the\@maxlabelwidth}%
533         }%
534     \fi
535 }

```

`\restore@labelwidth`

```

536 \def\restore@labelwidth{%
537     \@xp\ifx \csname r@[bibenv:\the\c@bib@env]\endcsname \relax
538     \resetbiblist{00}%

```



```

539   \else
540     \@xp\labelwidth\csname r@[bibenv:\the\c@bib@env]\endcsname
541     \leftmargin\labelwidth
542     \advance\leftmargin\labelsep
543   \fi
544 }

```

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

```

545 \providecommand{\ResetCapSFCodes}{%
546   \count@='A
547   \def\@tempa{%
548     \sfcode\count@=\@m
549     \advance\count@\@ne
550     \ifnum\count@>'Z\relax \expandafter\@gobble \fi
551   \@tempa
552 }%
553 \@tempa
554 }

```

`\CurrentBib` In case this is undefined sometimes.

```

555 \def\CurrentBib{??}

```

`\BibLabel`

```

556 \newcommand{\BibLabel}{%
557   [\hyper@anchorstart{cite.\CurrentBib}\relax\thebib\hyper@anchorend]%
558 }

```

`\resetbiblist`

```

559 \newcommand{\resetbiblist}[1]{%
560   \settowidth\labelwidth{\def\thebib{#1}\BibLabel}%
561   \leftmargin\labelwidth
562   \ifdim\labelwidth=\z@
563     \leftmargin=1em
564     \itemindent=-\leftmargin
565   \else
566     \advance\leftmargin\labelsep
567   \fi
568 }

```

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.

```

569 \newcommand{\bib}{%
570   \begingroup
571     \@ifstar{%
572       \@tempwattrue
573       \let\@bibdef\star@bibdef
574     \BibItem
575   }%

```

```

576          \@tempswafalse
577          \BibItem
578      }%
579 }

```

`\BibItem` *Arguments:*

```

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

```

```

580 \newcommand{\BibItem}[2]{%
581     \def\@tempa{#1}%
582     \edef\@tempb{%
583         \@nx\@bibdef\@xp\@nx\csname setbib@#2\endcsname{#2}%
584         {\macrotext\@tempa}%
585     }%
586     \@tempb
587 }

```

`\@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.

```

```

588 \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.

```

```

589 \AtBeginDocument{\let\bib@exec\bib@print}

```

6.12.1 `\@bibdef` Implementations

`\normal@bibdef` *Arguments:*

```

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

```

```

590 \def\normal@bibdef#1#2#3{%

```

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

```

591     \def\CurrentBibType{#2}%
592     \ifx\relax#1%
593         \amsrefs@error{Undefined entry type: #2}\@ehc

```

```

594     \let#1\setbib@misc
595     \fi
596     \RestrictedSetKeys{}{bib}%
597     {\bib@exec{#3}{\the\rsk@toks}{#1}\endgroup}%
598 }
599
600 \let\@bibdef\normal@bibdef

```

`\star@bibdef` *Arguments:*

```

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

601 \def\star@bibdef{%
602     \let\bib@exec\bib@store
603     \normal@bibdef
604 }

```

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

```

605 \def\copy@bibdef{%
606     \if@tempwa
607         \@xp\defer@bibdef
608     \else
609         \@xp\copy@bibdef@a
610     \fi
611 }

```

`\copy@bibdef@a`

```

612 \def\copy@bibdef@a#1#2#3#4{%
613     \@open@bbl@file
614     \process@xrefs{#4}%
615     \bbl@write{%
616         \string\bib\if@tempwa*\fi{#3}{#2}\string{\iffalse}\fi
617     }%

```

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.

```

618     \RestrictedSetKeys{\global\let\rsk@set\bbl@copy}\@empty
619     {\bbl@write{\iffalse{\fi\string^^J}%
620         \endgroup}{#4}%
621 }

```

```

622 \catcode'\:=11
623
624 \def\modify@xref@fields{%
625     \let\set:bib'author\output@xref@a
626     \let\set:bib'editor\output@xref@a
627     \let\set:bib'translator\output@xref@a
628     \let\set:bib'journal\output@xref@a
629     \let\set:bib'publisher\output@xref@a
630     \def\set:bib'xref##1##2{\output@xref@{##1}\@empty}%
631     \def\set:bib'book##1##2{\output@inner@xref@{##1}\@empty}%
632     \let\set:bib'conference\set:bib'book
633     \let\set:bib'partial\set:bib'book
634     \let\set:bib'reprint\set:bib'book
635     \let\set:bib'translation\set:bib'book
636 }
637

```

```

638 \catcode'\:=12
639
640 \def\process@xrefs#1{%
641   \begingroup
642     \RestrictedSetKeys{\modify@xref@fields}{bib}{\the\rsk@toks}{#1}%
643   \endgroup
644 }
645
646 \def\output@xref@a#1#2{%
647   \def\@tempa{#1}%
648   \lowercase{\def\@tempb{#1}}%
649   \ifx\@tempa\@tempb
650     \output@xref@{#1}%
651   \fi
652 }
653
654 \def\output@xref@#1{%
655   \@ifnotempty{#1}{%
656     \@ifundefined{bi@#1}{}{%
657       \begingroup
658         \let\star@bibdef\copy@bibdef@a
659         \csname bi@#1\endcsname
660       \endgroup
661     }%
662     \xp@g@undef\csname bi@#1\endcsname
663   }%
664 }
665
666 \def\output@inner@xref@#1{%
667   \in@={#1}%
668   \ifin@
669     \output@xref@{#1}%
670   \fi
671 }

```

\bbl@copy

```

672 \def\bbl@copy#1\endcsname#2{%
673   \begingroup
674     \def\@tempa{#1}%
675     \toks@{{#2}}%
676     \star@{\bbl@copy@a}{}%
677 }

```

\bbl@copy@a

```

678 \def\bbl@copy@a#1{%
679   \@ifnotempty{#1}{%
680     \add@toks@{*{#1}}%
681   }%
682   \bbl@write{ \space\@tempa=\the\toks@,}%
683 \endgroup
684 \rsk@resume
685 }

```

\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.
686 \def\selective@bibdef#1#2#3{%
687   \exp\selbibdef@a\csname b@#3\endcsname{#1}{#2}{#3}%
688 }

```

\selbibdef@a

```

689 \def\selbibdef@a#1{%
690   \def\@tempa{\endgroup\@gobblefour}%
691   \ifx\relax#1\else \exp\selbibdef@b#1\@nil \fi
692   \@tempa
693 }

```

\selbibdef@b

```

694 \def\selbibdef@b#1#2#3\@nil{%
695   \ifx 1#2\let\@tempa\copy@bibdef\fi
696 }

```

\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.

```

```

697 \def\defer@bibdef#1#2#3#4{%
698   \exp\gdef\csname bi@#3\endcsname{%
699     \bib*{#3}{#2}{#4}%
700   }%
701   \exp\addto@defer@list \csname bi@#3\endcsname
702   \endgroup
703 }

```

\bibdefer@list

```

704 \let\bibdefer@list\@empty

```

\addto@defer@list

```

705 \def\addto@defer@list#1{%
706   \begingroup
707   \def\do{\@nx\do\@nx}%
708   \xdef\bibdefer@list{\bibdefer@list\do#1}%
709   \endgroup
710 }

```

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.

```

711 \def\bib@store#1{%
712   \afterassignment\@gobble
713   \exp\xdef\csname bi@#1\endcsname
714 }

```

\numeric@refs

```

715 \def\numeric@refs{00}

```

```

\bib@print Arguments:
    #1 <- citekey.
    #2 <- \the\rsk@toks.
    #3 <- \setbib@bibtype.

716 \def\bib@print#1#2#3{%
717     \bib@start{#1}%
718     \let\setbib@@#3%
719     #2\relax      % execute definitions locally
720     \bib@resolve@xrefs
721     \bib@field@patches
722     \bib@selectlanguage
723     \generate@label
724     \bib'setup
725     \bib@cite{#1}%
726     \kern\@ne sp
727     \ifx\setbib@@\setbib@article
728         \ifx\bib'booktitle\@empty
729             \ifx\bib'book\@empty
730                 \ifx\bib'conference\@empty
731                     \else
732                         \let\setbib@@\setbib@incollection
733                     \fi
734                 \else
735                     \let\setbib@@\setbib@incollection
736                 \fi
737             \else
738                 \let\setbib@@\setbib@incollection
739             \fi
740         \fi
741         \setbib@@
742     \bib@end
743 }

```

`\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.

744 \def\bib@print@inner#1#2{%
745     \begingroup
746     #2\relax      % execute definitions locally
747     \bib@resolve@xrefs
748     \bib@field@patches
749     \bib'setup
750     #1%
751     \endgroup
752 }

```

```

\current@citekey

753 \let\current@citekey\@empty

```

```

\prev@citekey

754 \let\prev@citekey\@empty

```

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

```

755 \def\bib@start#1{%
756     \begingroup

```

```

757     \def\current@citekey{#1}%
758 }

```

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

```

759 \def\bib@end{%
760     \relax
761     \@xp\PrintBackRefs\@xp{\CurrentBib}%
762     \par
763     \save@primary
764     \global\let\prev@citekey\current@citekey
765     \endgroup
766 }

```

6.12.3 Resolving cross-references

`\bib@resolve@xrefs`

```

767 \def\bib@resolve@xrefs{%
768     \xref@check@c\bib'xref
769     \xref@check@a\bib'author
770     \xref@check@a\bib'editor
771     \xref@check@a\bib'translator
772     \xref@check@b\bib'journal
773     \xref@check@b\bib'publisher
774 }

```

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

```

775 \def\xref@check@a#1{%
776     \ifx\@empty#1\relax
777     \else
778         \begingroup
779             \toks@{\@emptytoks
780                 \@temptokenb\@emptytoks
781                 \series@index\z@
782                 \def\name{\xref@check@aa#1}%
783                 #1\relax
784                 \edef\@tempa{%
785                     \def\@nx#1{\the\toks@}%
786                     \the\@temptokenb
787                 }%
788             \@xp\endgroup
789             \@tempa
790         \fi
791 }

```

`\xref@check@aa`

```

792 \def\xref@check@aa#1#2{%
793     \advance\series@index\@ne
794     \def\@tempa{#2}%
795     \lowercase{\def\@tempb{#2}}%
796     \ifx\@tempa\@tempb
797         \ifx\@tempa\@empty
798             \add@toks@{\name{}}%
799         \else
800             \@ifundefined{bi@#2}{%
801                 \BibAbbrevWarning{#2}%
802                 \add@toks@{\name{#2}}%
803             }{%

```

```

804         \xref@check@ab#1{#2}%
805     }%
806     \fi
807     \else
808         \add@toks@{\name{#2}}%
809     \fi
810 }

```

\xref@check@ab

```

811 \def\xref@check@ab#1#2{%
812     \csname bi@#2\endcsname
813     \ifx\@empty\bib'name
814         \@temptokena{#2}%
815     \else
816         \@temptokena\exp{\bib'name}%
817         \get@property\@tempa\bib'name
818         \edef\@tempa{%
819             \@nx\addto@hook\@temptokenb{%
820                 \@nx\reset@nth@property\@nx#1\the\series@index{\@tempa}%
821             }%
822         }%
823         \@tempa
824     \fi
825     \edef\@tempa{\@nx\add@toks@{\@nx\name{\the\@temptokena}}}%
826     \@tempa
827 }

```

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

```

828 \def\xref@check@b#1{%
829     \ifx\@empty#1%
830     \else
831         \toks@\exp{#1}%
832         \edef\@tempb{\lowercase{\def\@nx\@tempa{\the\toks@}}}%
833         \@tempb
834         \ifx\@tempa#1\relax % all lowercase
835             \@ifundefined{bi@#1}{%
836                 \BibAbbrevWarning{#1}%
837             }{%

```

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.

```

838         \let#1\@empty
839         \xref@check@c\@tempa
840     }%
841     \fi
842 \fi
843 }

```

\xref@check@c Resolve an xref field.

```

844 \def\xref@check@c#1{%
845     \ifx#1\@empty
846     \else
847         \begingroup
848             \@apply\auto@protect\amsrefs@textsymbols
849             \@apply\auto@protect\amsrefs@textaccents
850             \let\DSK@def\xref@add@toks
851             \let\DSK@append\xref@append

```



```

852      \toks@\@emptytoks
853      \let\bib@reset\@empty
The \@for here is just a fancy way of expanding #1. (Or is it?)
854      \@for\xref@ID:=#1\do{%
855          \ifundefined{bi@\xref@ID}{%
856              \XRefWarning{\xref@ID}%
857          }{%
858              \csname bi@\xref@ID\endcsname
859          }%
860      }%
861      \edef\@tempa{\endgroup\the\toks@}%
862      \@tempa
863      \fi
864  }

```

`\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.

```

```

865 \def\xref@add@toks#1#2#3{%
866     \ifx#1\@empty
867         \edef\@tempa{%
868             \@nx\add@toks@{\@xp\@nx\csname\rkv@setter#1\endcsname{#2}{#3}}%
869         }%
870         \@tempa
871     \else
872         \in@\bib'title{#1}%
873         \ifin@
874             \ifx\bib'booktitle\@empty
875                 \edef\@tempa{%
876                     \@nx\add@toks@{%
877                         \@xp\@nx\csname set:bib'booktitle\endcsname
878                     }%
879                 }%
880                 \@tempa
881                 \add@toks@{{#2}{#3}}%
882             \fi
883         \fi
884     \fi
885 }

886 \def\xref@append#1#2#3#4{%
887     \edef\@tempa{%
888         \@nx\add@toks@{\@xp\@nx\csname\rkv@setter#2\endcsname{#3}{#4}}%
889     }%
890     \@tempa
891 }

```

`\BibAbbrevWarning`

```

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

```

`\XrefWarning`

```

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

```

6.12.4 Bib field preprocessing

```

\current@primary
894 \let\current@primary\@empty

\previous@primary
895 \let\previous@primary\@empty

\save@primary
896 \IfOption{nobysame}{%
897   \let\save@primary\@empty
898 }{%
899   \def\save@primary{%
900     \global\let\previous@primary\current@primary
901   }%
902 }

```

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

```

903 \def\bib@field@patches{%
904   \ifx\bib'author\@empty
905     \ifx\bib'editor\@empty
906       \let\current@primary\bib'translator
907       \let\print@primary\PrintTranslatorsA
908     \else
909       \let\current@primary\bib'editor
910       \let\print@primary\PrintEditorsA
911     \fi
912   \else
913     \let\current@primary\bib'author
914     \let\print@primary\PrintAuthors
915   \fi
916   \ifx\bib'address\@empty
917     \let\bib'address\bib'place
918   \fi
919   \ifx\bib'organization\@empty
920     \ifx\bib'institution\@empty
921       \let\bib'organization\bib'school
922     \else
923       \let\bib'organization\bib'institution
924     \fi
925   \fi
926   \ifx\bib'date\@empty
927     \ifx\bib'year\@empty
928       \let\bib@year\bib'status
929     \else
930       \bib@parsedate\bib'year
931     \fi
932   \else
933     \bib@parsedate\bib'date
934   \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.

```

935   \def\@tempa{year}%
936   \ifx\bib'volume\@tempa
937     \let\bib'volume\bib@year

```

```

938      \let\bib'date\@empty
939    \fi

```

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

```

940    \ifx\setbib@@\setbib@article
941      \ifx\bib'volume\@empty
942        \ifx\bib'number\@empty\else
943          \let\bib'volume\bib'number
944          \let\bib'number\@empty
945        \fi
946      \fi
947    \fi

```

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

```

948    \bib@language@fixup
949  }

```

6.12.5 Date setup

`\bib@year`

```

950 \let\bib@year\@empty

```

`\bib@month`

```

951 \let\bib@month\@empty

```

`\bib@day`

```

952 \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.

```

953 \def\bib@parsedate#1{%
954   \@xp\bib@parsedate@a#1---\@nil
955 }

```

`\bib@parsedate@a`

```

956 \def\bib@parsedate@a#1-#2-#3-#4\@nil{%
957   \def\bib@year{#1}%
958   \def\bib@month{#2}%
959   \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.

```

960   \ifx\@empty\bib@day
961     \ifx\@empty\bib@month
962       \let\bib'date\bib@year
963     \else
964       \def\bib'date{#1-#2}%
965     \fi
966   \else
967     \def\bib'date{#1-#2-#3}%
968   \fi
969 }

```

6.12.6 Language setup

`\bib@language@fixup`

```

970 \def\bib@language@fixup{%
971   \ifx\bib'hyphenation\@empty
972     \ifx\bib'language\@empty
973       \let\bib@language\biblanguagedefault
974     \else
975       \let\bib@language\bib'language
976     \fi
977   \else
978     \let\bib@language\bib'hyphenation
979   \fi
980   \def\@tempa##1 ##2\@nil{\lowercase{\def\bib@language{##1}}}%

```

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

```

981   \@firstofone{\@xp\@tempa\bib@language} \@nil
982 }

```

`\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.

```

983 \def\bib@selectlanguage{%
984   \@ifsame@patterns{\language}{\bib@language}{\}%
985   \@ifundefined{date\bib@language}{\}%
986   \@xp\selectlanguage\@xp{\bib@language}%
987   }%
988 }%
989 }

```

`\@ifsame@patterns`

```

990 \def\@ifsame@patterns#1#2{%
991   \@xp\@ifsamepat\csname l@#1\@xp\endcsname\csname l@#2\endcsname
992 }

```

`\@ifsamepat`

```

993 \def\@ifsamepat#1#2{%
994   \ifnum \ifx\relax#1\m@ne\else#1\fi = \ifx\relax#2\m@ne\else#2\fi
995     \@xp\@firstoftwo
996   \else
997     \@xp\@secondoftwo
998   \fi
999 }

```

`\language`

`\biblanguageEnglish`

```
1000 \providecommand{\language}{english}
```

`\biblanguagedefault`

```
1001 \def\biblanguageEnglish{english}
```

`\bib@language`

```

1002 \let\biblanguagedefault\biblanguageEnglish
1003 \let\bib@language\@empty

```

6.12.7 Citation label setup

`\generate@label`

```
1004 \let\generate@label\relax
```

```

\cite@label
1005 \def\cite@label{\@currentlabel}

\alpha@label
1006 \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.

1007 \def\bib@cite#1{%
1008   \def\CurrentBib{#1}%
1009   \alpha@label           % modify \thebib if necessary
1010   \item\leavevmode
1011   \SK@\SK@@label{#1}%
1012   \@xp\bib@cite@a\csname b@#1\endcsname
1013   \bibcite@write{#1}%
1014 }

1015 \def\bib@cite@a#1{%
1016   \ifx\relax#1%
1017     \begingroup
1018     \auto@protect\etaltext
1019     \protected@edef\@tempa{%
1020       \gdef\@nx#1{%
1021         \@nx\citesel 01{\cite@label}{\bib@label@year}{}%
1022       }%
1023     }%
1024     \@xp\endgroup
1025     \@tempa
1026   \else
1027     \@xp\bib@cite@check\@xp#1#1\@empty\@empty\@empty\@empty\@empty
1028   \fi
1029 }

\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.

1030 \def\bib@cite@check#1#2#3#4#5#6#7{%
1031   \ifx 1#4\relax
1032     \DuplicateBibKeyWarning
1033   \else

```

This has gotten *way* out of hand.

```

1034     \begingroup
1035         \auto@protect\etaltext
1036         \@apply\auto@protect\amsrefs@textsymbols
1037         \@apply\auto@protect\amsrefs@textaccents
1038         \@tempswafalse
1039         \in@CitePrintUndefined{#5}%
1040         \ifin@
1041             \let\@tempa\@empty
1042         \else
1043             \def\@tempa{#5}%
1044         \fi
1045         \ifx\@tempa\@empty
1046         \else
1047             \exp\ifx\@xp\@currentlabel\cite@label
1048                 \edef\@tempb{\cite@label}%
1049             \else
1050                 \let\@tempb\cite@label
1051             \fi
1052             \ifx\@tempa\@tempb
1053                 \def\@tempa{#6}%
1054                 \ifx\@tempa\bib@label@year
1055                     \else
1056                         \@tempswatrue
1057                     \fi
1058                 \else
1059                     \@tempswatrue
1060                 \fi
1061             \fi
1062             \if@tempswa
1063                 \@ifempty{#6}{%
1064                     \def\@tempa{#5}%
1065                     \let\@tempb\cite@label
1066                 }{%
1067                     \def\@tempa{#5, #6}%
1068                     \def\@tempb{\cite@label, \bib@label@year}%
1069                 }%
1070             \amsrefs@warning{Citation label for \extr@cite#1 is
1071                 changing from '\@tempa ' to '\@tempb '}%
1072             \fi
1073             \protected@edef\@tempa{%
1074                 \gdef\@nx#1{%
1075                     \@nx\citesel #31{\cite@label}{\bib@label@year}{#7}%
1076                 }%
1077             }%
1078         \@xp\endgroup
1079         \@tempa
1080     \fi
1081 }

\bib@label@year

1082 \let\bib@label@year\@empty

\DuplicateBibKeyWarning

1083 \def\DuplicateBibKeyWarning{%
1084     \amsrefs@warning{%
1085         Duplicate \protect\bib\space key
1086         '\CurrentBib ' detected\MessageBreakNS}%
1087 }
```

`\bibcite@write`

```
1088 \def\bibcite@write#1{%
1089     \if@filesw
1090         \begingroup
1091             \let\citesel\citesel@write
1092             \csname b@#1\endcsname
1093         \endgroup
1094     \fi
1095 }
```

`\citesel@write`

```
1096 \def\citesel@write#1#2#3#4#5{%
1097     \toks@{{#3}-{#4}}%
1098     \immediate\write\@auxout{\string\bibcite{\CurrentBib}{\the\toks@}}%
1099 }
```

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

```
1100 \AtEndDocument{\let\bibcite\@gobbletwo}
```

6.12.8 Printing the bibliography

`\bibname`

```
1101 \providecommand{\bibname}{Bibliography}
```

`\refname`

```
1102 \providecommand{\refname}{References}
```

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

```
1103 \let\bib@div@mark\@gobble
```

This is verbose, but probably safer than any alternative.

```
1104 \@ifclassloaded{amsbook}{}{%
1105     \@ifclassloaded{amsart}{}{%
1106         \@ifclassloaded{amsproc}{}{%
1107             \def\bib@div@mark#1{%
1108                 \@mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}%
1109             }%
1110         }%
1111     }%
1112 }
```

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

```
1113 \newenvironment{bibchapter}[1][\bibname]{%
1114     \begingroup
1115         \protected@edef\@{%
1116             \endgroup
1117             \protect\chapter*{#1}%
1118             \protect\bib@div@mark{#1}%
1119         }%
1120     \@
1121 }\par}
```

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

```
1122 \newenvironment{bibsection}[1][\refname]{%
1123     \begingroup
```

```

1124     \protected@edef\@{%
1125         \endgroup
1126     \protect\section*{#1}%
1127     \protect\bib@div@mark{#1}%
1128 }%
1129 \@
1130 }\par}

```

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

```

1131 \ifundefined{chapter}{%
1132     \newenvironment{bibdiv}{\bibsection}{\endbibsection}
1133 }{%
1134     \newenvironment{bibdiv}{\bibchapter}{\endbibchapter}
1135 }

```

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

```

1136 \renewenvironment{thebibliography}[1]{%
1137     \bibdiv
1138     \biblist[\resetbiblist{#1}]%
1139 }{%
1140     \endbiblist
1141     \endbibdiv
1142 }

```

6.13 Name, journal and publisher abbreviations

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

\DefineName

```

1143 \newcommand{\DefineName}[2]{%
1144     \bib*{#1}{name}{name={#2}}%
1145 }

```

\DefineJournal

```

1146 \newcommand{\DefineJournal}[4]{%
1147     \bib*{#1}{periodical}{
1148         issn={#2},
1149         journal={#4}
1150     }%
1151 }

```

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

```

1152 \newcommand{\DefinePublisher}[4]{%
1153     \bib*{#1}{publisher}{%
1154         publisher={#3},
1155         address={#4}
1156     }%
1157 }

```


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`

```
1158 \newcommand{\bibselect}{%
1159   \@ifstar{%
1160     \let\@bibdef\copy@bibdef
1161     \BibSelect
1162   }{%
1163     \let\@bibdef\selective@bibdef
1164     \BibSelect
1165   }%
1166 }
```

`\BibSelect`

```
1167 \newcommand{\BibSelect}[2][\bblname]{%
1168   \if@files
1169     \typeout{Trying to create bbl file '#1.bbl' ...}%
1170     \def\bibselect@msg{%
1171       \typeout{ ... rats. Unable to create bbl file.}%
1172     }%
1173     \let\@open@bbl@file\OpenBBLFile
1174     \@for\@tempa:=#2\do{\ReadBibData{\@tempa}}%
1175   \fi
1176   \@close@bbl@file
1177   \@apply@g@undef\bibdefer@list
1178   \global\let\bibdefer@list\@empty
1179   \let\@bibdef\normal@bibdef
1180   \input{#1.bbl}%
1181   \let\BibSelect\MultipleBibSelectWarning
1182 }
```

Now read the .bbl file we just created.

`\MultipleBibSelectWarning`

```
1183 \newcommand\MultipleBibSelectWarning[2][{%
1184   \amsrefs@warning{%
1185     Multiple \string\bibselect 's found (only one
1186     \string\bibselect\space per biblist environment is allowed)%
1187   }%
1188 }
```

`\bblname`

```
1189 \def\bblname{\jobname}
```

`\bib@dbfile`

```
1190 \newread\bib@dbfile
```

`\ReadBibData`

```
1191 \newcommand{\ReadBibData}[1]{%
1192   \IfFileExists{#1.ltb}{%
1193     \openin\bib@dbfile=\@filef@und \relax
```

```

1194   }{%
1195       \IfFileExists{#1.ltx}{%
1196           \openin\bib@dbfile=\@filef@und \relax
1197       }{%
1198           \IfFileExists{#1.tex}{%
1199               \openin\bib@dbfile=\@filef@und \relax
1200           }{%
1201               \begingroup
1202                   \NoBibDBFile{#1}%
1203                   \let\ReadBibData@a\endgroup
1204           }%
1205       }%
1206   }%
1207   \ReadBibData@a
1208 }

```

\NoBibDBFile

```

1209 \def\NoBibDBFile#1{%
1210     \amsrefs@warning{No data file #1.ltb (.ltx, .tex) found}%
1211 }

```

\ReadBibData@a

```

1212 \def\ReadBibData@a{%
1213     \ProvidesFile{\@filef@und}\relax
1214     \begingroup
1215         \let\star@bibdef\defer@bibdef
1216         \ReadBibLoop
1217     \endgroup
1218     \closein\bib@dbfile
1219 }

```

\ReadBibLoop

```

1220 \def\ReadBibLoop{%
1221     \ifeof\bib@dbfile
1222         \@xp\@gobble
1223     \else
1224         \read\bib@dbfile to\CurLine
1225         The \@empty is in case \CurLine is empty.
1226         \@xp\ReadBibLoop@a\CurLine\@empty\@nil
1227     \fi
1228 }

```

\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.

```

1229 \long\def\ReadBibLoop@a#1#2\@nil{%
1230     \ifx\bib#1%
1231         \CurLine % just exec it
1232     \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.

```

1233     \ifx\endinput#1%
1234         \let\ReadBibLoop\@empty
1235     \else

```

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

```

1236      \@xp\ReadBibLoop@b#1#2\@empty\@nil
1237      \fi
1238      \fi
1239  }
```

`\ReadBibLoop@b`

```

1240 \long\def\ReadBibLoop@b#1#2\@nil{%
1241     \ifx\bib#1%
1242         \CurLine % just exec it
1243     \fi
1244 }

1245 \let\bbl@out=\relax
1246 \let\bbl@write\@gobble
1247 \let\@open@bbl@file\relax
1248 \let\@close@bbl@file\relax
```

`\OpenBBLFile`

```

1249 \def\OpenBBLFile{%
1250     \if@filesw
1251         % Just use the next unused output stream
1252         \count@\count17
1253         \advance\count@\@ne
1254         \ifnum\count@<\sist@\@n
1255             \global\chardef\bbl@out=\count@
1256             \immediate\openout\bbl@out=\bblname.bbl\relax
1257             \global\let\@close@bbl@file\CloseBBLFile
1258             \gdef\bbl@write{\immediate\write\bbl@out}%
1259         \else
1260             \ch@ck\count@\sist@\@n\write
1261         \fi
1262     \fi
1263     \global\let\@open@bbl@file\relax
1264 }
```

`\CloseBBLFile`

```

1265 \def\CloseBBLFile{%
1266     \immediate\closeout\bbl@out\relax
1267     \global\let\@close@bbl@file\relax
1268     \global\let\bbl@write\@gobble
1269     \global\let\bbl@out\relax
1270 }
```

6.15 Citation processing

6.15.1 The `\citesel` structure

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

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

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

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

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

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

`\citesel@update`

```
1271 \def\citesel@update#1#2#3#4#5#6{%
1272   \gdef#6{\citesel 1#2{#3}{#4}{#5}}%
1273 }
```

`\citesel@number`

```
1274 \def\citesel@number#1#2#3#4#5{#3}
```

`\citesel@year`

```
1275 \def\citesel@year#1#2#3#4#5{#4}
```

`\citesel`

```
1276 \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 `\citel` prevents certain problems that could arise with the `showkeys` package.

```
1277 \expandafter\let\csname cite \endcsname\relax
```

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

```
1278 \renewcommand{\cite}[2] [] {%
1279   \if\cite@single#2,\@gobble \else\MultipleCiteKeyWarning{#2}{#1}\fi
1280   \@ifempty{#1}{%
1281     \cites@o{#2}%
1282   }{%
1283     \ObsoleteCiteOptionWarning
1284     \cites@a{*{#1}}{#2}%
1285   }%
1286 }
```

`\MultipleCiteKeyWarning`

```
1287 \def\MultipleCiteKeyWarning#1#2{%
1288   \amsrefs@warning{%
1289     Use of \string\cites\space is recommended instead of %
1290     \string\cite\space\MessageBreak
1291     for multiple cites '#1'%
1292   \@ifnotempty{#2}{%
1293     \amsrefs@warning{Star option requires \string\citelist\space here}%
1294   }%
1295   \global\let\MultipleCiteKeyWarning\@gobbletwo
1296 }
```

ObsoleteCiteOptionWarning

```

1297 \def\ObsoleteCiteOptionWarning{%
1298   \amsrefs@warning{%
1299     The form \string\cite{...}*{...} is recommended\MessageBreak
1300     instead of \string\cite[...]{...}%
1301   \global\let\ObsoleteCiteOptionWarning\@empty
1302 }

```

\cite@single

```

1303 \edef\cite@single#1,#2{\iffalse{\fi\iffalse{\fi\string}#2.\string}}

```

\cites@o

```

1304 \def\cites@o#1{\star@{\cites@oo{#1}}{}}

```

\cites@oo

```

1305 \def\cites@oo#1#2{\@ifempty{#2}{\cites@a{#1}}{\cites@a{*#2}{#1}}}

```

\cites@a

```

1306 \def\cites@a#1#2{%
1307   \begingroup
1308   \toks@{\endgroup \cites@b{#1}}%
1309   \vdef\@tempa{#2}%
1310   \edef\@tempa{%
1311     \the\toks@ \@firstofone{\@xp\zap@space\@tempa} \@empty
1312   }%
1313   \@tempa,\@empty
1314   \edef\@tempa{\endgroup\@nx\citelist{\the\toks@}}%
1315   \@tempa
1316 }

```

\cites@b

```

1317 \def\cites@b#1#2,#3{%
1318   \begingroup
1319   \toks@{\InnerCite{#2}{#1}}%
1320   \ifx\@empty#3\@xp\@gobble\fi
1321   \cites@c#3%
1322 }

```

\cites@c

```

1323 \def\cites@c#1,#2{%
1324   \add@toks@{\InnerCite{#1}}%
1325   \ifx\@empty#2\@xp\@gobble\fi
1326   \cites@c#2%
1327 }

```

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

```

1328 \def\citeleft{[]
1329 \def\citeright{[]

```

```

1330 \def\citimid{\penalty9999 \space}
1331 \def\citepunct{\penalty9999 \hskip.13em plus.1em minus.05em\relax}

```

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

```
1332 \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.

```
1333 \providecommand{\citeform}{\@firstofone}
```

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

```
1334 \DeclareRobustCommand{\citelist}{\@citelist}
```

`\@citelist`

```
1335 \def\@citelist#1{%
1336     \leavevmode
1337     \begingroup
1338         \@citestyle
1339         \citeleft\nopunct    % suppress first \citepunct
1340         \cite@begingroup
1341         \in@*{#1}%
1342         \ifin@
1343             \let\citepunct\CiteAltPunct
1344         \fi
1345         \let\cite@endgroup\@empty
1346         \cites@init
1347         \def\citeleft{\@addpunct{\citepunct}}%
1348         \let\citeright\ignorespaces
1349         \def\cite{\InnerCite}%
1350         \process@citelist{#1}%
1351     \endgroup
1352     \citeright
1353 \endgroup
1354 }
```

`\@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).

```
1355 \providecommand{\@citestyle}{\m@th\upshape\mdseries}
```

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

```
1356 \def\cite@begingroup{\begingroup\let\cite@begingroup\relax}
```

`\cite@endgroup`

```
1357 \let\cite@endgroup\endgroup
```

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

```
1358 \def\cites@init{%
1359     \gdef\prev@names{???}%
1360     \let\cites@init\@empty
1361 }
```

`\InnerCite`

```
1362 \newcommand{\InnerCite}[1]{\star@\cite@a\citesel{#1}}{}}
```

Arguments:

```

1363 \def\cite@a#1#2{%
1364     \BackCite{#2}%
1365     \cite@begingroup
1366         \cites@init
1367         \let\citesel#1\relax
1368         \ifx\citesel\citesel@author
1369             \let\citeleft\@empty
1370             \let\citeright\@empty
1371         \fi
1372     \begingroup
1373         \toks@{\endgroup \cite@b}%
1374         \vdef\@tempa{#2}%
1375         \edef\@tempa{%
1376             \the\toks@{\@firstofone{\@xp\zap@space\@tempa} \@empty}%
1377         }%
1378     \@tempa
1379 }
```

```

1380 \def\cite@b#1#2{%
1381     \exp\cite@bc\csname b@#1\exp\endcsname {#1}#{2}%
1382 }

```

```

1383 \def\cite@bc#1#2{%
1384     \ifx#1\@@undefined \global\let#1\relax \fi
1385     \ifx#1\relax \global\let#1\empty@cite \fi
1386     \@xp\cite@nobib@test#1{}{}{}{}@nil#1%
1387     \cite@cj#1%
1388 }

```

```
1389 \def\empty@cite{\citesel 00{}{}{}}
```

Arguments:

```

1390 \def\cite@nobib@test#1#2#3#4#5\@nil#6{%
1391     \@ifempty{#4}{%
1392         \G@refundefinedtrue

```

```

1393      \UndefinedCiteWarning#6%
1394      \xdef#6{\@nx\citesel #2#3{%
1395          \@nx\CitePrintUndefined{\extr@cite#6}}-{}{}%
1396      }{}%
1397 }

```

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

```

1398 \def\UndefinedCiteWarning#1{%
1399     \@latex@warning{%
1400         Citation ‘\extr@cite#1’ on page \thepage\space undefined}%
1401 }

```

`\CitePrintUndefined`

```

1402 \DeclareRobustCommand{\CitePrintUndefined}[1]{%
1403     \begingroup\fontshape{n}\fontseries\mddefault \ttfamily ?#1\endgroup
1404 }

```

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

```

1405 \let\CPU@normal\CitePrintUndefined

```

`\cite@cj` *Arguments:*

```

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

```

```

1406 \def\cite@cj#1#2{%
1407     \leavevmode
1408     \begingroup
1409         \cite@cb#1% write info to aux file
1410         \ar@SK@cite#1%
1411         \@citeleft
1412         \ar@hyperlink{#1}%
1413         \@ifnotempty{#2}{\citamid{#2}}%
1414         \citeright
1415     \endgroup
1416     \ignorespaces % ignore spaces inside \citelist
1417 \cite@endgroup
1418 }

```

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

```

1419 \def\@citeleft{\citeleft}

```

`\cite@cb`

```

1420 \def\cite@cb#1{%
1421     \if@files@w
1422         \immediate\write\@auxout{\string\citation{\extr@cite#1}}%
1423     \fi

```

Define `\citesel` to make `\b@whatever` update itself.

```

1424     \begingroup
1425         \let\citesel\citesel@update
1426         #1#1%
1427     \endgroup
1428 }

```

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

```

1429 \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.)

```
1430 \DeclareRobustCommand{\cites}{\cites@a{}}
```

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

```
1431 \providecommand{\citen}{\ocite}
```

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

```
1432 \def\ycite{\cite}
```

\ycites

```
1433 \let\ycites\cites
```

\ocite

```
1434 \let\ocite\ycite
```

\ocites

```
1435 \let\ocites\cites
```

\fullcite

```
1436 \let\fullcite\cite
```

\fullocite

```
1437 \let\fullocite\ocite
```

\citeauthor

```
1438 \let\citeauthor\ycite
```

\citeauthorhy

```
1439 \let\citeauthorhy\ycite
```

6.15.4 The \nocite command

\nocite

```
1440 \renewcommand{\nocite}[1]{\othercites{#1}}
```

\othercites

```
1441 \newcommand{\othercites}[1]{%
1442   \cite@begingroup
1443     \let\BackCite\@gobble
1444     \let\cite@endgroup\@empty
1445     \def\citelist{\othercitelist}%
1446     \cites{#1}%
1447 }
```

\othercitelist

```
1448 \newcommand{\othercitelist}[1]{%
1449   \cite@begingroup
1450     \let\cite@endgroup\@empty
1451     \cites@init
1452     \let\citeleft\relax
1453     \let\citeright\ignorespaces
1454     \def\InnerCite{\OtherCite}%

```

```

1455      \def\cite@cj ##1##2{%
1456      \begingroup
1457      \@xp\citesel##1%
1458      \cite@cb ##1%
1459      \endgroup

```

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

```

1460      \@xp\ifx\csname b@*\endcsname ##1%
1461      \global\let\selective@bibdef\copy@bibdef
1462      \fi
1463      \ignorespaces
1464      \cite@endgroup
1465      }%
1466      #1\relax
1467      \endgroup
1468 }

```

`\OtherCite`

```

1469 \def\OtherCite#1{\cite@a\citesel@other{#1}{}}

```

`\citesel@other`

```

1470 \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.

```

1471 \@namedef{b@*}{\citesel 11{*}{*}{*}}

```

6.15.5 Citation sorting

`\process@citelist@sorted`

```

1472 \def\process@citelist@sorted#1{%
1473   \ifx\citesel\citesel@number
1474     \cite@sorted@s #1\cite@sorted@e
1475   \else
1476     \NonNumericCiteWarning
1477     \process@citelist@unsorted{#1}%
1478   \fi
1479 }

```

`\NonNumericCiteWarning`

```

1480 \def\NonNumericCiteWarning{%
1481   \amsrefs@warning{%
1482     Unable to confirm that cite keys are numeric: not sorting%
1483   }%
1484 }

```

`\process@citelist@unsorted`

```

1485 \def\process@citelist@unsorted#1{%
1486   \ignorespaces#1\relax
1487 }

```

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

```

1488 \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.

```

1489 \def\CPU@sort#1{2147483647}

```

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

```

1490 \def\cite@sorted@s{%
1491     \begingroup
1492     \let\CitePrintUndefined\CPU@sort
1493     \let\cite@cjs\cite@cj
1494     \let\cite@cj\cite@compress
1495     \begingroup
1496     \toks@{\emptytoks}
1497     \let\cite@cj\cite@sort
1498     \ignorespaces
1499 }

```

`\cite@sorted@e`

```

1500 \def\cite@sorted@e{%
1501     \@xp\endgroup
1502     \the\toks@
1503     \cite@dash
1504     \prev@cite
1505     \endgroup
1506 }

```

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

Arguments:

```

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

```

```

1507 \def\cite@sort#1#2{%
1508     \safe@set\@tempcnta#1% highest number so far
1509     \toks@{\cite@cj#1{#2}}%
1510     \@temptokena\toks@
1511     \let\cite@cj\cite@sort@a
1512     \ignorespaces
1513 }

```

`\cite@sort@a`

```

1514 \def\cite@sort@a#1#2{%
1515     \safe@set\@tempcntb#1%
1516     \ifnum\@tempcntb > \@tempcnta
1517         \add@toks@{\cite@cj#1{#2}}%
1518         \@tempcnta\@tempcntb
1519     \else
1520         \let\cite@cj\cite@sort@b
1521         \toks@{\emptytoks}
1522         \def\@tempb{\add@toks@{\cite@cj#1{#2}}}%
1523         \the\@temptokena
1524         \@tempb
1525         \let\cite@cj\cite@sort@a
1526     \fi
1527     \@temptokena\toks@
1528     \ignorespaces
1529 }

```

`\cite@sort@b`

```

1530 \def\cite@sort@b#1#2{%
1531     \safe@set\count@#1%
1532     \ifnum\@tempcntb < \count@
1533         \@tempb
1534         \let\@tempb\empty

```

```

1535 \fi
1536 \add@toks@{\cite@cj#1{#2}}%
1537 \ignorespaces
1538 }

```

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
```

```
1539 \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).

```

1540 \def\prev@cite@cb{%
1541     \ifx\@prev@cite\@empty
1542     \else
1543         \begingroup
1544             \def\cite@print##1##2{%
1545                 \cite@cb##1%
1546             }%
1547             \prev@cite
1548         \endgroup
1549     \fi
1550 }
```

`\cite@print`

```

1551 \def\cite@print#1#2{%
1552     \begingroup
1553         \let\CitePrintUndefined\CPU@normal
1554         \cite@cjs#1{#2}%
1555     \endgroup
1556 }
```

`\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.

```
1557 \let\cite@dash\@empty
```

`\print@one@dash`

```

1558 \def\print@one@dash{%
1559     \textendash \nopunct
1560     \let\cite@dash\@empty
1561 }
```

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:

```

1562 \def\cite@compress#1#2{%
1563     \cite@print#1{#2}%
1564     \@ifempty{#2}{%
1565         \safe@set\@tempcnta#1%
1566         \let\cite@cj\cite@compress@a
1567     }{}%
1568 }
```

`\cite@compress@a` State 1:

```

1569 \def\cite@compress@a#1#2{%
1570     \@ifempty{#2}{%
1571         \advance\@tempcnta\@ne
1572         \safe@set\@tempcntb#1%
1573         \ifnum\@tempcnta=\@tempcntb
```

```

1574         \def\prev@cite{\cite@print#1{}}%
1575         \let\cite@cj\cite@compress@b
1576     \else
1577         \cite@print#1{}%
1578         \@tempcnta\@tempcntb
1579     \fi
1580 }{%
1581     \cite@print#1{#2}%
1582     \let\cite@cj\cite@compress
1583 }%
1584 }

```

\cite@compress@b State 2:

```

1585 \def\cite@compress@b#1#2{%
1586     \@ifempty{#2}{%
1587         \advance\@tempcnta\@ne
1588         \safe@set\@tempcntb#1%
1589         \ifnum\@tempcnta=\@tempcntb
1590             \let\cite@dash\print@one@dash
1591             \prev@cite@cb
1592             \def\prev@cite{\cite@print#1{}}%
1593         \else
1594             \cite@dash
1595             \prev@cite
1596             \let\prev@cite\@empty
1597             \cite@print#1{}%
1598             \@tempcnta\@tempcntb
1599             \let\cite@cj\cite@compress@a
1600         \fi
1601     }{%
1602         \cite@dash
1603         \prev@cite
1604         \let\prev@cite\@empty
1605         \cite@print#1{#2}%
1606         \let\cite@cj\cite@compress
1607     }%
1608 }

```

6.15.7 Munging the .aux file

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

```

1609 \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.

```

1610 \AtBeginDocument{%
1611     \if@files
1612         \immediate\write\@auxout{%
1613             \string\ifundefined{amsrefs@bibcite}{-}{-}%
1614             \string\let\string\bibcite\string\amsrefs@bibcite
1615         }%
1616     }%
1617 \fi

```

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

```

1618     \let\bibcite\amsrefs@bibcite
1619 }

```

`\bibcite@a` *Arguments:*

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

```
1620 \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.

```
1621   \ifx\relax#1%
1622     \gdef#1{\citesel 00#2{}}%
1623   \else
1624     \begingroup
1625       \exp\bibcite@b\exp#1#1#2}%
1626     \endgroup
1627   \fi
1628 }
```

`\bibcite@b` *Arguments:*

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

```
1629 \def\bibcite@b#1#2#3#4#5#6#7#8{\gdef#1{\citesel#3#4#8{#7}}}
```

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

```
1630 \newcommand{\citedest}[1]{\exp\cite@dest\csname b@#1\endcsname}
```

`\cite@dest`

```
1631 \def\cite@dest#1{%
1632   \ifx\relax#1%
1633     \gdef#1{\citesel 00{}{}{}}%
1634   \fi
1635   \exp\cite@dest@b\exp#1#1%
1636 }
```

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

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

```
1637 \def\cite@dest@b#1#2#3#4#5#6#7#8{%
1638   \ifempty{#7}{%
1639     \def#1{\citesel #3#4{#5}{#6}{#8}}%
1640   }{%
1641     \gdef#1{\citesel #3#4{#5}{#6}{#7,#8}}%
1642   }%
1643 }
```

6.15.8 Back references

```

\ifBR@verbose
1644 \@ifundefined{ifBR@verbose}{\let\ifBR@verbose\iffalse \let\fi\fi}{}%

\BackCite
1645 \let\BackCite\@gobble

\back@cite
1646 \def\back@cite#1{%
1647   \ifBR@verbose
1648     \PackageInfo{backref}{back cite \string '\extr@cite#1'}%
1649   \fi
1650   \Hy@backout{#1}%
1651 }

\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.
1652 \def\print@backrefs#1{%
1653   \space\SentenceSpace$\uparrow$\csname br@#1\endcsname
1654 }

\PrintBackRefs
1655 \let\PrintBackRefs\@gobble

```

6.15.9 hyperref, showkeys and shaderef support

```

\shade@cite
1656 \newcommand{\shade@cite}{\overshadebox{shaderef-gray}}

\ar@hyperlink
1657 \def\ar@hyperlink#1{%
1658   \hyper@link [cite]{}{cite.\extr@cite#1}{\shade@cite{#1}}%
1659 }

\ar@SK@cite
1660 \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.
1661 \AtBeginDocument{%
1662   \@ifpackageloaded{shaderef}{}%
1663   \let\shade@cite\@empty
1664 }%
1665 \@ifpackageloaded{hyperref}{}%
1666   \let\ar@hyperlink\shade@cite
1667   \let\hyper@anchorstart\@gobble
1668   \let\hyper@anchorend\relax
1669 }%
1670 \@ifpackageloaded{showkeys}{%
1671   \@ifpackagewith{showkeys}{notcite}{%
1672     \let\ar@SK@cite\@gobble
1673   }{}
1674 }{%
1675   \let\ar@SK@cite\@gobble
1676   \let\SK@@label\@gobble
1677   \let\SK@\@gobbletwo
1678 }%
1679 }

```


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 72).³

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

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

```
1680 \let\amsrefs@textaccents\@empty
```

`\DeclareNameAccent` *Arguments:*

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

```
1681 \def\DeclareNameAccent{%
1682   \@lappend\amsrefs@textaccents
1683 }
```

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

```
1684 \DeclareNameAccent\"
```

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

```

1685 \DeclareNameAccent\'
1686 \DeclareNameAccent\.
1687 \DeclareNameAccent\=
1688 \DeclareNameAccent\^
1689 \DeclareNameAccent\`
1690 \DeclareNameAccent\~%
1691 \DeclareNameAccent\b
1692 \DeclareNameAccent\c
1693 \DeclareNameAccent\d
1694 \DeclareNameAccent\H
1695 \DeclareNameAccent\k
1696 \DeclareNameAccent\r
1697 \DeclareNameAccent\t
1698 \DeclareNameAccent\u
1699 \DeclareNameAccent\v

```

From mathscinet:

```

1700 \DeclareNameAccent\utilde
1701 \DeclareNameAccent\uarc
1702 \DeclareNameAccent\dudot
1703 \DeclareNameAccent\lfhook
1704 \DeclareNameAccent\udot
1705 \DeclareNameAccent\polhk
1706 \DeclareNameAccent\soft

```

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

```

1707 \DeclareNameAccent\etalchar
1708 %\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`.

```
1709 \let\amsrefs@textsymbols\@empty
```

`\DeclareNameSymbol` *Arguments:*

#1 <- *symbol*.

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

```

#2 <- lowercase.
1710 \def\DeclareNameSymbol#1#2{%
1711     \@lappend\amsrefs@textsymbols#1%
1712     \@lappend\amsrefs@textsymbols#2%
1713     \ifx#1#2\else
1714         \@lappend\amsrefs@textsymbols#2%
1715         \@lappend\amsrefs@textsymbols#2%
1716     \fi
1717 }

```

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.

```

1718 \DeclareNameSymbol\i\i
1719 \DeclareNameSymbol\j\j
1720 \DeclareNameSymbol\AE\ae
1721 \DeclareNameSymbol\OE\oe
1722 \DeclareNameSymbol\O\o
1723 \DeclareNameSymbol\DH\dh
1724 \DeclareNameSymbol\DJ\dj
1725 \DeclareNameSymbol\L\l
1726 \DeclareNameSymbol\NG\ng
1727 \DeclareNameSymbol\SS\ss
1728 \DeclareNameSymbol\TH\th

```

From mathscinet:

```

1729 \DeclareNameSymbol\Dbar\dbar
1730 \DeclareNameSymbol\lasp\lasp
1731 \DeclareNameSymbol\rasp\rasp
1732 \DeclareNameSymbol\cprime\cprime
1733 \DeclareNameSymbol\cdprime\cdprime
1734 \DeclareNameSymbol\bud\bud
1735 \DeclareNameSymbol\cydot\cydot

```

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

```

1736 \DeclareNameSymbol~~%

```

6.16.3 `\edef`-like macros for names

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

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

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

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

```
\normalize@edef\cs{P{'o}lya}
```

`\cs` will contain `P{'o}lya`. (This might result in a redundant layer of braces if the original text contained, say, “`P{'o}lya`”, but that’s ok.) This lets us extract the first n characters from a name by using T_EX’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.

```

1737 \def\normalize@edef#1#2{%
1738     \begingroup
1739     \@apply\auto@protect\amsrefs@textsymbols
1740     \@apply\wrap@accent\amsrefs@textaccents

```

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

```

1741      \let\@tabacckludge\use@accent
1742      \let~\space
1743      \edef\@tempa{\def\@nx#1{#2}}%
1744      \@xp\endgroup
1745      \@tempa
1746 }

```

`\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

```

1747 \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}`.

```

1748 \def\wrap@accent#1{%
1749     \def#1##1{{\@nx#1##1}}%
1750 }

```

`\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.

```

1751 \def\lc@edef#1#2{%
1752     \begingroup
1753     \let\@tabacckludge\use@accent %%??
1754     \@apply\auto@protect\amsrefs@textaccents
1755     \@apply\lc@do\amsrefs@textsymbols
1756     \edef\@tempa{\lowercase{\def\@nx#1{#2}}}%
1757     \@xp\endgroup
1758     \@tempa
1759 }

```

`\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.

```

1760 \def\lc@do#1\do#2{%
1761     \ifx#1#2%
1762         \auto@protect#1%
1763     \else
1764         \def#1{#2}%
1765     \fi
1766 }

```

`\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`.

```

1767 \def\purge@edef#1#2{%

```

```

1768 \begingroup
1769 \let\auto@protect\amsrefs@textsymbols
1770 \let\space
1771 \let\purge@accent\amsrefs@textaccents
1772 \let\@tabacckludge\@gobble

```

As mentioned above (page 59), `\i` and `\j` are semantically like text accents; hence, they require special treatment here.

```

1773 \def\i{i}%
1774 \def\j{j}%
1775 \edef\@tempa{#2}%
1776 \toks@{\emptytoks}
1777 \exp\purge@edef@ \@tempa \@nil
1778 \edef\@tempa{\def\@nx#1{the\toks@}}%
1779 \exp\endgroup
1780 \@tempa
1781 }

```

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

```

1782 \def\purge@edef@{%
1783 \futurelet\@let@token
1784 \purge@edef@a
1785 }

```

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

```

1786 \def\purge@edef@a#1{%

```

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

```

1787 \ifx\@let@token\@nil
1788 \let\@tempa\empty
1789 \else

```

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

```

1790 \ifx\@let@token\bgroup
1791 \def\@tempa{%
1792 \purge@edef@ #1\@nil
1793 \purge@edef@
1794 }%
1795 \else

```

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

```

1796 \add@toks@{#1}%
1797 \let\@tempa\purge@edef@
1798 \fi
1799 \fi
1800 \@tempa
1801 }

```

`\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).

```

1802 \def\purge@accent#1{%
1803 \def#1##1{##1}%
1804 }

```

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 57), 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.)

```
1805 \def\name@split{%
1806     \get@namepart\bib'surname\name@split@given
1807 }
```

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

```
1808 \def\name@split@given{%
1809     \get@namepart\bib'given\name@split@jr
1810 }
```

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

```
1811 \def\name@split@jr{%
1812     \get@namepart\bib'jr\name@split@finish
1813 }
```

`\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.

```
1814 \def\name@split@finish{%
1815     \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.

```
1816 %%     \ifx\@empty\bib'given
1817 %%         \NameCheck \bib'surname ??\@nil
1818 %%     \else
1819         \extract@initials\bib'given
1820 %%     \fi
1821     \@nilgobble
1822 }
```

`\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.

```
1823 \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.

```
1824     \toks@{#1}%
1825     \@temptokena{#2}%
1826     \get@namepart@a
1827 }
```

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

```
1828 \def\get@namepart@a{%
1829     \futurelet\@let@token
1830     \get@namepart@b
1831 }
```

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

```
1832 \def\get@namepart@b{%
1833     \ifx\@let@token\@sptoken
1834         \xp\get@namepart@c
1835     \else
1836         \xp\get@namepart@d
1837     \fi
1838 }
```

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

```
1839 \def\get@namepart@c{%
1840     \after@deleting@token\get@namepart@a
1841 }
```

`\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.

```
1842 \def\get@namepart@d{%
1843     \futurelet\@let@token
1844     \get@namepart@e
1845 }
```

`\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.

```
1846 \def\get@namepart@e{%
1847     \ifx\@let@token,%
1848         \xp\let\the\toks@\@empty
1849         \edef\@tempa{%
1850             \@nx\after@deleting@token\the\@temptokena
1851         }%
```



```

1852      \@xp\@tempa
1853      \else
1854      \@xp\get@namepart@f
1855      \fi
1856 }

```

`\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.

```

1857 \def\get@namepart@f#1#2,{%
1858   \ifx\@let@token\bgroup
1859   \@xp\def\the\toks@{{#1}#2}%
1860   \else
1861   \@xp\def\the\toks@{{#1}#2}%
1862   \fi
1863   \the\@temptokena
1864 }

```

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

```

1865 \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.)

⁵More or less. If the second argument is brace-delimited, the braces will be lost. But as mentioned above (page 63), we don't really care.

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.

```

1866 \def\extract@initials#1{%
1867   \begingroup
1868     \auto@protect\ini@hyphen
1869     \auto@protect\nobreakspace
1870     \let~\relax
1871     \@apply\auto@protect\amsrefs@textsymbols
1872     \@apply\auto@protect\amsrefs@textaccents
1873     \normalize@edef\@tempa{#1}%
1874     \ifx\@tempa\@empty
1875     \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.-:”)

```

1876       \process@hyphens\@tempa
1877       \process@dots\@tempa

```

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

```

1878         \process@names\@tempa
1879         \@chomp\@tempa{~}%
1880     \fi
1881     \edef\@tempa{\def\@nx\bib'initials{\@tempa}}%
1882 \exp\endgroup
1883 \@tempa
1884 }

```

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

```
1885 \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.

```

1886 \def\process@hyphens#1{%
1887     \begingroup
1888     \toks@{\emptytoks
1889     \@xp\process@one@hyphen #1-\@nil
1890     \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.)

```

1891     \@chomp\@tempa{ \ini@hyphen\space}%
1892     \edef\@tempa{\def\@nx#1{\@tempa}}%
1893 \exp\endgroup
1894 \@tempa
1895 }

```

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

```

1896 \def\process@one@hyphen{%
1897     \futurelet\@let@token
1898     \process@one@hyphen@a
1899 }

```

`\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.

```

1900 \def\process@one@hyphen@a{%
1901     \ifx\@let@token\@nil
1902         \let\fsa@n\@gobble
1903     \else
1904         \ifx\@let@token -%
1905             \TrailingHyphenWarning
1906             \let\fsa@n\process@one@hyphen@b
1907         \else
1908             \ifx\@let@token\@sptoken

```

```

1909             \let\fsa@n\process@one@hyphen@b
1910         \else
1911             \let\fsa@n\process@one@hyphen@c
1912         \fi
1913     \fi
1914 \fi
1915 \fsa@n
1916 }

```

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

```

1917 \def\process@one@hyphen@b{%
1918     \after@deleting@token\process@one@hyphen
1919 }

```

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

```

1920 \def\process@one@hyphen@c#1#2-{%
1921     \ifx\bgroup\@let@token
1922         \add@toks@{#1}#2 \ini@hyphen\space}%
1923     \else
1924         \add@toks@{#1#2 \ini@hyphen\space}%
1925     \fi
1926     \futurelet\@let@token
1927     \process@one@hyphen@d
1928 }

```

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

```

1929 \def\process@one@hyphen@d{%
1930     \ifx\@let@token\@nil
1931         \@xp\@gobble
1932     \else
1933         \@xp\process@one@hyphen
1934     \fi
1935 }

```

\TrailingHyphenWarning Or translator or contributor or...

```

1936 \def\TrailingHyphenWarning{%
1937     \amsrefs@warning{Trailing hyphen deleted from name}%
1938 }

```

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

```

1939 \def\process@dots#1{%
1940     \begingroup
1941         \toks@\@emptytoks
1942         \@xp\process@one@dot #1.\@nil
1943         \edef\@tempa{\the\toks@}%
1944         \@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.

```

1945         \@chomp\@tempa{ }%
1946         \edef\@tempa{\def\@nx#1{\@tempa}}%
1947     \@xp\endgroup
1948     \@tempa
1949 }

```

\process@one@dot

```

1950 \def\process@one@dot{%

```

```

1951 \futurelet\@let@token
1952 \process@one@dot@a
1953 }

```

`\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`.

```

1954 \def\process@one@dot@a{%
1955 \ifx\@let@token .%
1956 \def\fsa@n{\after@deleting@token\process@bare@dot}%
1957 \else
1958 \ifx\@let@token\@sptoken
1959 \let\fsa@n\process@one@dot@b
1960 \else
1961 \let\fsa@n\process@one@dot@c
1962 \fi
1963 \fi
1964 \fsa@n
1965 }

```

`\process@bare@dot`

```

1966 \def\process@bare@dot{%
1967 \add@toks@{. }%
1968 \futurelet\@let@token
1969 \process@one@dot@d
1970 }

```

`\process@one@dot@b`

```

1971 \def\process@one@dot@b{%
1972 \after@deleting@token\process@one@dot
1973 }

```

`\process@one@dot@c`

```

1974 \def\process@one@dot@c#1#2.{%
1975 \ifx\bgroup\@let@token
1976 \add@toks@{#{1}#2. }%
1977 \else
1978 \add@toks@{#1#2. }%
1979 \fi
1980 \futurelet\@let@token
1981 \process@one@dot@d
1982 }

```

`\process@one@dot@d`

```

1983 \def\process@one@dot@d{%
1984 \ifx\@let@token\@nil
1985 \xp\@gobble
1986 \else
1987 \xp\process@one@dot
1988 \fi
1989 }

```

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

```

1990 \def\process@names#1{%
1991 \begingroup
1992 \toks@\@emptytoks
1993 \xp\extract@initial #1 \@nil

```

```

1994      \edef\@tempa{\def\@nx#1{\the\toks@}}%
1995      \@xp\endgroup
1996      \@tempa
1997 }

```

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

```

1998 \def\extract@initial{%
1999     \futurelet\@let@token
2000     \extract@initial@a
2001 }

```

`\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.

```

2002 \def\extract@initial@a{%
2003     \ifx\@let@token\@nil
2004         \let\fsa@n\@gobble
2005     \else
2006         \ifx\@let@token\@sptoken
2007             \let\fsa@n\extract@initial@b
2008         \else
2009             \let\fsa@n\extract@initial@c
2010         \fi
2011     \fi
2012     \fsa@n
2013 }

```

`\extract@initial@b`

```

2014 \def\extract@initial@b{%
2015     \after@deleting@token\extract@initial
2016 }

```

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

```

2017 \def\extract@initial@c#1#2 {%
2018     \ifx\@let@token\bgroup

```

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

```

2019         \@extract@initial {{#1}}#2\@nil
2020     \else
2021         \@extract@initial #1#2\@nil
2022     \fi
2023     \futurelet\@let@token
2024     \extract@initial@d
2025 }

```

`\extract@initial@d`

```

2026 \def\extract@initial@d{%
2027     \ifx\@let@token\@nil
2028         \@xp\@gobble
2029     \else
2030         \@xp\extract@initial
2031     \fi
2032 }

```

`\@extract@initial` This handles the four cases mentioned on page 66.

```

2033 \def\@extract@initial#1#2\@nil{%
2034     \ifx\ini@hyphen#1%
2035         \add@toks@\ini@hyphen}%
2036     \else
2037         \in@{.\@nil}{#1#2\@nil}% Look for a period at the end of the name
2038         \ifin@
2039             \add@toks@{#1#2~}%
2040         \else
2041             \count@chars\@tempcnta{#1#2}%
2042             \ifnum\@tempcnta > \@ne
2043                 \add@toks@{#1.~}%
2044             \else
2045                 \add@toks@{#1~}%
2046             \fi
2047         \fi
2048     \fi
2049 }

```

`\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.

```

2050 \def\count@chars#1#2{%
2051     \begingroup
2052         \@tempcnta\z@
2053         \@count@chars#2\@nil
2054         \edef\@tempb{#1=\the\@tempcnta\relax}%
2055     \exp\endgroup
2056     \@tempb
2057 }

```

`\@count@chars`

```

2058 \def\@count@chars#1{%
2059     \ifx #1\@nil
2060     \else
2061         \advance\@tempcnta\@ne
2062         \xp\@count@chars
2063     \fi
2064 }

```

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

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

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

```
2065 \let\previous@stem\@empty
2066 \let\current@stem\@empty

2067 \let\previous@year\@empty
2068 \let\current@year\@empty
```

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

```
2069 \def\append@to@stem{\global\@concat\current@stem}
```

```
\generate@alphalabel
```

```
2070 \def\generate@alphalabel{%
```

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

```
2071 \ifx\bib'label\@empty
2072 \begingroup
```

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

```
2073 \global\let\previous@stem\current@stem
2074 \global\let\current@stem\@empty
```

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

```
\name{Last1,First1} \name{Last2,First2} ... \name{Lastn,Firstn}
```

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

```
2075 \@apply\auto@protect\amsrefs@textsymbols
2076 \@apply\auto@protect\amsrefs@textaccents
2077 \auto@protect\name
2078 \auto@protect\etaltext
2079 \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.

```
2080 \get@numberof\@tempcnta\name\@tempa
2081 \calc@author@part
```


Next append the year part.

```
2082      \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.

```
2083      \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.)

```
2084      \edef\@tempa{%
2085          \def\nx\bib'label{%
2086              \current@stem
2087              \alpha@label@suffix
2088          }%
2089      }%
2090      \exp\endgroup
2091      \@tempa
2092  \fi
2093 }
```

`\calc@author@part`

```
2094 \def\calc@author@part{%
2095     \ifnum \@tempcnta = 1
2096         \exp\@oneauthorlabel\exp{\@tempa}%
2097     \else
2098         \exp\@multiauthorlabel\exp{\@tempa}%
2099     \fi
2100 }
```

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

```
2101 \def\@firstone#1{\@car#1\@empty\@nil}
```

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

```
2102 \def\@firstthree#1{\@carcube#1\@empty\@empty\@empty\@nil}
```

`\@nametoken`

```
2103 \let\@nametoken\@firstone
```

`\hyph@to@space`

```
2104 \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).

```
2105 \def\@marknames#1{%
2106     \@ifnotempty{#1}{\surround@names#1 ' }%
2107 }
```

`\surround@names`

```
2108 \def\surround@names#1 {%
2109     \ifx '#1%
2110     \else
2111         \nx\@nametoken{#1}%
2112         \exp\surround@names
2113     \fi
2114 }
```

\extract@surnames

```
2115 \def\extract@surnames#1#2{%
2116     \get@namepart\@tempb\@nilgobble #2,\@nil
2117     \edef\@tempb{\@nx\@marknames{\@xp\hyph@to@space\@tempb\@gobble-}}%
2118     \edef#1{\@tempb}%
2119 }
```

\@oneauthorlabel This is the easy case.

```
2120 \newcommand{\@oneauthorlabel}[1]{%
2121     \def\name##1{%
2122         \extract@surnames\@tempa{##1}%
2123         \get@numberof\@tempcnta\@nametoken\@tempa
2124         \ifnum \@tempcnta = 1
2125             \let\@nametoken\@firstthree
2126         \fi
2127         \append@to@stem{\@tempa}%
2128     }%
2129     #1%
2130 }
```

\@threeauthors

```
2131 \def\@threeauthors\name#1\name#2\name#3#4\@empty{%
2132     \name{#1}\name{#2}\name{#3}%
2133     \append@to@stem{\etalchar{+}}%
2134 }
```

\@multiauthorlabel

```
2135 \newcommand{\@multiauthorlabel}[1]{%
2136     \def\name##1{%
2137         \ifx\etaltext ##1%
2138             \def\@tempa{\@nx\etalchar{+}}%
2139             \let\name\@gobble
2140         \else
2141             \extract@surnames\@tempa{##1}%
2142         \fi
2143         \append@to@stem{\@tempa}%
2144     }%
2145     \ifnum \@tempcnta > 4 \@xp \@threeauthors \fi
2146     #1\@empty
2147 }
```

\etalchar

```
2148 \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.

```
2149 \def\year@short#1#2#3#4\@nil{#3#4}
```

\append@label@year

```
2150 \def\append@label@year{%
2151     \safe@set\@tempcnta\bib@year
2152     \edef\bib@citeyear{the\@tempcnta}%
2153     \append@to@stem{%
2154         \ifx\bib@year\@empty
2155         \else
2156             \@xp\year@short \bib@citeyear \@nil
2157         \fi
2158     }%
2159 }
```

```

2160 \let\alpha@label@suffix\@empty
2161
2162 \newcount\alpha@suffix
2163 \alpha@suffix\@ne
2164 \let\@suffix@format\@alph

```

\calc@alpha@suffix

```

2165 \def\calc@alpha@suffix{%
2166   \@tempswafalse
2167   \compare@stems\previous@stem\current@stem
2168   \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.

```

2169     \ifx\previous@year\current@year
2170       \@tempswatrue
2171     \fi
2172   \fi
2173   \if@tempswa
2174     \global\advance\alpha@suffix\@ne
2175     \edef\alpha@label@suffix{\@suffix@format\alpha@suffix}%
2176     \ifnum\alpha@suffix=\tw@
2177       \immediate\write\@auxout{%
2178         \string\ModifyBibLabel{\prev@citekey}%
2179       }%
2180     \fi
2181   \else
2182     \let\alpha@label@suffix\@empty
2183     \global\alpha@suffix\@ne
2184     \@xp\ifx \csname b@\current@citekey @suffix\endcsname \relax
2185     \else
2186       \edef\alpha@label@suffix{\@suffix@format\alpha@suffix}%
2187     \fi
2188   \fi
2189 }

```

\ifsame@stems

```

2190 \newif\ifsame@stems

```

\compare@stems

```

2191 \def\compare@stems#1#2{%
2192   \begingroup
2193     \purge@edef\@tempa{#1}%
2194     \purge@edef\@tempb{#2}%
2195     \lc@edef\@tempa{\@tempa}%
2196     \lc@edef\@tempb{\@tempb}%
2197     \ifx\@tempa\@tempb
2198       \def\@tempa{\same@stemstrue}%
2199     \else
2200       \def\@tempa{\same@stemsfalse}%
2201     \fi
2202   \@xp\endgroup
2203   \@tempa
2204 }

```

\ModifyBibLabel

```

2205 \def\ModifyBibLabel#1{%
2206   \global\@xp\let\csname b@#1@suffix\endcsname\@empty
2207 }

```

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 90 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

$$\text{\PrintSeries}\{S\}\{i_1\}\{i_2\}\{i_3\}\{E\}\{\text{\do}\{T_1\} \dots \text{\do}\{T_n\}\}$$

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

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

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\} \dots` is

$$\begin{aligned} &f_0\{A\} \\ &f_0\{A\} \text{ p1 } i_1 f_1\{B\} \\ &f_0\{A\} \text{ p2 } i_2 f_2\{B\} \text{ p3 } i_3 f_3\{B\} \end{aligned}$$

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}\... \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.

```
2208 \newcount\sindex
```

\PrintSeries

```
2209 \def\PrintSeries#1#2#3#4#5#6#7#8{%
2210     \begingroup
2211         \def\sindex@add@a{#2}%
2212         \def\sindex@add@b{\SwapBreak{#3}#4}%
2213         \def\sindex@add@c{\SwapBreak{#5}#6}%
2214         \def\sindex@add@d{\SwapBreak{#7}#8}%
2215         \def\sindex@add@e{\SwapBreak{#7}}%
2216         \PrintSeries@a{#1}%
2217 }
```

\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**.

```
2218 \def\PrintSeries@a#1#2#3#4{%
2219     \get@numberof\@tempcnta#1{#3}%
2220     \chardef\sindex@total=\@tempcnta
2221     \ifnum\sindex@total=\@ne
2222         \let\SingularPlural\@firstoftwo
2223     \else
2224         \let\SingularPlural\@secondoftwo
2225     \fi
2226     \sindex=index=\z@
2227     \let#1\sindex@add
2228     #2#3#4\relax
2229 \endgroup
2230 }
```

\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**.

```
2231 \def\sindex@add#1{%
2232     \advance\sindex@index\@ne
2233     \ifx\etaltext#1\relax
2234         \ifnum\sindex@index=\tw@
2235             \def\@tempa{\space\SubEtal}%
2236         \else
2237             \def\@tempa{\series@add@e\space\SubEtal}%
2238         \fi
```

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

```
2239     \series@index\@MM
2240     \else
2241         \ifcase\sindex@index
2242         \or
```

Material before name 1:

```
2243         \let\@tempa\sindex@add@a
2244         \or
```

Material before name 2:

```

2245      \ifnum\series@total<\thr@@
2246      \let\@tempa\series@add@b
2247      \else
2248      \let\@tempa\series@add@c
2249      \fi
2250      \else

```

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

```

2251      \ifnum\series@index=\series@total
2252      \let\@tempa\series@add@d
2253      \else
2254      \ifnum\series@index<\series@total
2255      \let\@tempa\series@add@c
2256      \else
2257      \let\@tempa\@gobble
2258      \fi
2259      \fi
2260      \fi
2261      \fi
2262      \@tempa{#1}%
2263 }

```

\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.

```

2264 \def\SwapBreak#1{%
2265     \relax\ifvmode\leavevmode\fi
2266     \@tempcnta\@MM
2267     \toks@{#1}%

```

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

```

2268     \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.

```

2269     \ifnum \lastkern>\@ne \unkern \fi

```

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

```

2270     \ifnum\lastpenalty=\z@
2271     \else
2272         \@tempcnta\lastpenalty

```

```

2273      \unpenalty
2274      \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`.

```

2275      \edef\@tempa{%
2276          \@nx\deferredquoteslogical
2277          \ifnum\lastkern=\@ne
2278          \else
2279              \ifnum\spacefactor=\sfcode\@xp\@xp\@xp'\@xp\@car\string#1)\@nil
2280              \else
2281                  \ifnum\spacefactor=\@nopunctsfcode
2282                  \else
2283                      \the\toks@
2284                  \fi
2285              \fi
2286          \fi
2287          \@nx\deferredquotes
2288          \ifnum\@tempcnta=\@MM \else \penalty\number\@tempcnta\space \fi
2289          \ifnum\lastkern=\@ne \ignorespaces \fi
2290      }%
2291      \@tempa
2292 }

```

`\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.

```

2293 \newcommand{\SingularPlural}[2]{#1}
2294 \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.)

```

2295 \BibSpec{nameLE}{
2296     +{}{}{given}
2297     +{}{\IfEmptyBibField{given}}{}{ }{}{surname}
2298     +{}{ }{}{jr}
2299 }

```

`\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.”

```

2300 \BibSpec{nameBE}{

```

```

2301    +{}{}{surname}
2302    +{}{ }{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.

```

2303 %    +{}{ }{jr}
2304 }

```

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

```

2305 \BibSpec{nameinverted}{
2306    +{} {} {surname}
2307    +{,}{ } {given}
2308    +{,}{ } {jr}
2309 }

```

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.

```

2310 \newcommand{\PrintNames}{%
2311    \@ifstar{\PrintNames@a\set@othername}{\PrintNames@a\set@firstname}%
2312 }

```

`\PrintNames@a`

```

2313 \newcommand{\PrintNames@a}[4]{%
2314    \PrintSeries{name}
2315    {#1}
2316    {}{ and \set@othername}
2317    {,}{ \set@othername}
2318    {,}{ and \set@othername}
2319    {#2}{#4}{#3}%
2320 }

```

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

```

2321 \def\set@firstname#1{%
2322    \set@name{#1}\setbib@nameLE
2323 }

```


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

```
2324 \def\set@othername#1{%
2325     \set@name{#1}\setbib@nameLE
2326 }
```

`\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.

```
2327 \def\set@name#1{%
2328     \name@split#1,,,\@nil
2329     \set@name@a
2330 }
```

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

```
2331 \def\set@name@a#1{%
2332     \begingroup
2333     \get@current@properties
2334     \select@auxlanguage
2335     \def\@tempa{yes}%
2336     \ifx\@tempa\prop'inverted
2337         \setbib@nameBE
2338     \else
2339         #1%
2340     \fi
2341 \endgroup
2342 }
```

`\PrintPrimary`

```
2343 \def\PrintPrimary{%
2344     \ifx\current@primary\@empty
2345         \EmptyPrimaryWarning
2346     \else
2347         \print@primary\current@primary
2348     \fi
2349 }
```

`\EmptyPrimaryWarning`

```
2350 \def\EmptyPrimaryWarning{%
2351     \amsrefs@warning{No authors, editors or translators}%
2352 }
```

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

```
2353 \newcommand{\PrintAuthors}[1]{%
2354     \ifx\previous@primary\current@primary
2355         \sameauthors\@empty
2356     \else
2357         \def\current@bibfield{\bib'author}%
2358         \PrintNames{}-{}-{}{#1}%
2359     \fi
2360 }
```

`\sameauthors`

```
2361 \newcommand{\sameauthors}[1]{\bysame#1}
```

`\bysame`

```
2362 \def\bysame{%
```

```

2363 \leavevmode\hbox to3em{\hrulefill}\thinspace
2364 \kern\z@
2365 }

```

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

```

2366 \newcommand{\PrintNameList}{\PrintNames{}}

```

`\PrintEditorsC`

```

2367 \newcommand{\PrintEditorsC}[1]{%
2368 \PrintNames{Edited by }{#1}%
2369 }

```

`\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.

```

2370 \newcommand{\PrintEditorsA}[1]{%
2371 \ifx\previous@primary\current@primary
2372 \sameauthors{(ed\Plural{s}.)}%
2373 \else
2374 \def\current@bibfield{\bib'editor}%
2375 \PrintNames{(ed\Plural{s}.)}{#1}%
2376 \fi
2377 \erase@field\bib'editor
2378 }

```

`\PrintEditorsB`

```

2379 \newcommand{\PrintEditorsB}{%
2380 \PrintNames*{(\SwapBreak{,}~ed\Plural{s}.)}%
2381 }

```

`\PrintContributions`

```

2382 \newcommand{\PrintContributions}[1]{%
2383 \PrintSeries
2384 {\fld@elt}
2385 {\print@contribution}
2386 {}{ and \print@contribution}
2387 {,}{ \print@contribution}
2388 {,}{ and \print@contribution}{#1}{%
2389 }

```

`\print@contribution`

```

2390 \newcommand{\print@contribution}[1]{%
2391 \in@=#1}%
2392 \ifin@
2393 \ifnum\series@index=\@ne with \fi
2394 \RestrictedSetKeys{\bib}{%
2395 \bib@print@inner\setbib@contribution{\the\rsk@toks}%
2396 }{#1}%
2397 \else
2398 #1%
2399 \fi
2400 }

```

`\resolve@inner`

```

2401 \def\resolve@inner#1#2{%
2402 \in@=#2}%
2403 \ifin@

```

```

2404     \RestrictedSetKeys{}\bib}\#1{\the\rsk@toks}\#2}%
2405     \else
2406     \@ifundefined{bi@#2}{%
2407     \XRefWarning{#2}%
2408     }{%
2409     #1{\csname bi@#2\endcsname}%
2410     }%
2411     \fi
2412 }

```

\PrintConference

```

2413 \def\PrintConference{%
2414     \resolve@inner{\bib@print@inner\setbib@conference}
2415 }

```

\PrintConferenceDetails

```

2416 \def\PrintConferenceDetails#1{%
2417     \ifx\@empty\bib'address
2418     \ifx\@empty\bib'date
2419     \else
2420     \PrintConferenceDetails@
2421     \fi
2422     \else
2423     \PrintConferenceDetails@
2424     \fi
2425 }

```

\PrintConferenceDetails@

```

2426 \def\PrintConferenceDetails@{%
2427     \ifnum\lastkern=\@ne\else\space\fi(\kern 1sp
2428     \ifx\@empty\bib'address
2429     \else
2430     \bib'address
2431     \fi
2432     \ifx\@empty\bib'date
2433     \else
2434     \SwapBreak{,}\space
2435     \print@date
2436     \fi
2437     )%\spacefactor\sfcode'\,%
2438 }

```

\PrintBook

```

2439 \def\PrintBook{%
2440     \resolve@inner{\bib@print@inner\setbib@innerbook}
2441 }

```

\PrintReprint

```

2442 \def\PrintReprint{%
2443     \resolve@inner{\bib@reprint}
2444 }

```

\bib@reprint

```

2445 \def\bib@reprint#1{%
2446     \begingroup
2447     #1\relax % execute definitions locally
2448     \bib@resolve@xrefs
2449     \bib@field@patches

```

```

2450      \bib'setup
2451      \IfEmptyBibField{copula}{reprinted in}{\bib'copula} \nopunct
2452      \let\bib'language\@empty
2453      \setbib@book
2454      \endgroup
2455 }

```

\PrintTranslation

```

2456 \def\PrintTranslation{%
2457     \resolve@inner{\bib@translation}
2458 }

```

\bib@translation

```

2459 \def\bib@translation#1{%
2460     \begingroup
2461     #1\relax           % execute definitions locally
2462     \bib@resolve@xrefs
2463     \bib@field@patches
2464     \bib'setup
2465     \let\PrintPrimary\@empty
2466     \bib@append{;}{ % keep this space!
2467         \IfEmptyBibField{language}{English}{\bib'language} transl.%
2468         \IfEmptyBibField{pages}{ in \kern\@one sp}{, }%
2469     }\bib'transition
2470     \let\bib'language\@empty
2471     \setbib@@
2472     \endgroup
2473 }

```

\PrintTranslatorsC

```

2474 \newcommand{\PrintTranslatorsC}[1]{%
2475     \PrintNames{translated by }{ }{#1}%
2476 }

```

\PrintTranslatorsA

```

2477 \newcommand{\PrintTranslatorsA}[1]{%
2478     \ifx\previous@primary\current@primary
2479         \sameauthors{(trans.)}%
2480     \else
2481         \def\current@bibfield{\bib'translator}%
2482         \PrintNames{}{ (trans.)}{#1}%
2483     \fi
2484     \erase@field\bib'translator
2485 }

```

\PrintTranslatorsB

```

2486 \newcommand{\PrintTranslatorsB}[1]{
2487     \PrintNames*{ }{\SwapBreak{,}~tran\Plural{s}.)}%
2488 }

```

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

```

2489 \DefineName{alii}{\etaltext}
2490 \DefineName{others}{\etaltext}

```

\etaltext The Chicago Manual of Style suggests that it is slightly better not to italicize ‘et al’ and some other extremely common abbreviations inherited from Latin. (Compare ‘etc’.)

\SubEtal

```

2491 \newcommand{\etaltext}{et al.}
2492 \newcommand{\SubEtal}[1]{\etaltext}

```

6.23 The partial field

`\print@partial`

```
2493 \newcommand{\print@partial}{%
2494   \resolve@inner{\bib@print@inner\setbib@partial}
2495 }
```

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.

```
2496 \newcommand{\parenthesize}[1]{%
2497   \leavevmode\push@bracket)\upn{(#1\pop@bracket
2498 }
```

`\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.)

```
2499 \providecommand{\upn}[1]{#1}
```

`\push@bracket`

```
\pop@bracket 2500 \let\bracket@stack\@empty
2501
2502 \def\push@bracket#1{%
2503   \xdef\bracket@stack{#1\bracket@stack}%
2504 }
2505
2506 \def\pop@bracket{%
2507   \iffalse\fi
2508   \@xp\pop@bracket@a\bracket@stack \@empty}%
2509 }
2510
2511 \def\pop@bracket@a#1{%
2512   \leavevmode\upn{#1}%
2513   \xdef\bracket@stack{\iffalse}\fi
2514 }
```

`\bibquotes`

```
2515 \newcommand{\bibquotes}[1]{%
2516   \textquotedblleft#1%
2517   \gdef\deferredquotes{%
2518     \global\let\deferredquotes\@empty
2519     \textquotedblright
2520   }%
2521 }
```

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

```
\ndash 2522 \providecommand{\mdash}{\textendash}
2523 \providecommand{\ndash}{\textendash}
```

`\@scanforMR`

```
2524 \def\strip@MRprefix#1#2#3#4\@nil{%
2525   \def\@tempa{#1#2#3#4}%
2526   \if#1M%
2527     \if#2R%
2528       \def\@tempa{#3#4}%
2529     \fi
2530   \fi
2531 }
```



```

2565      #1\IfEmptyBibField{eprint}{}\IfEmptyBibField{journal}{ pp.}{}}%
2566 }

```

\PrintThesisType

```

2567 \def\PrintThesisType#1{%
2568     \thesis@type#1?\@nil{#1}%
2569 }
2570
2571 \def\thesis@type#1#2\@nil#3{%
2572     \ifx p#1%
2573         Ph.D. Thesis%
2574     \else
2575         \ifx m#1%
2576             Master's Thesis%
2577         \else
2578             #3%
2579         \fi
2580     \fi
2581 }

```

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

```

2582 \newcommand{\PrintDOI}[1]{%
2583     DOI #1%
2584     \IfEmptyBibField{volume}{, (to appear in print)}{}%
2585 }

```

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

```

2586 \newcommand{\PrintDatePV}[1]{%
2587     \IfEmptyBibField{doi}{%
2588         \let\@tempa\PrintDate
2589     }{%
2590         \IfEmptyBibField{volume}{%
2591             \let\@tempa\PrintDatePosted
2592         }{%
2593             \let\@tempa\PrintDate
2594         }%
2595     }%
2596     \@tempa{#1}%
2597 }

```

\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.

```

2598 \newcommand{\PrintDate}[1]{(\print@date)}

```

\PrintDateB The same, but without the parentheses.

```

2599 \newcommand{\PrintDateB}[1]{\print@date}

```

\print@date

```

2600 \def\print@date{%
2601     \ifx\bib@month\@empty
2602     \else
2603         \print@month@day
2604     \fi
2605     \bib@year
2606 }

```

`\print@month@day`

```
2607 \def\print@month@day{%
2608     \bib@monthname
2609     \ifx\@empty\bib@day \else \nobreakspace\number 0\bib@day,\fi
2610     \space
2611 }
```

`\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.

```
2612 \newcommand{\bib@monthname}{%
2613     \ifcase 0\bib@month
2614     \or January\or February\or March\or April\or May\or June\or
2615     July\or August\or September\or October\or November\or December\or
2616     Winter\or Spring\or Summer\or Fall\else Unknown Month%
2617     \fi
2618 }
```

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

```
2619 \newcommand{\PrintYear}[1]{\bib@year}
```

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

```
2620 \newcommand{\PrintDatePosted}[1]{\unskip, posted on \print@date}
```

`\PrintEdition`

```
2621 \newcommand{\PrintEdition}[1]{%
2622     \afterassignment\print@edition
2623     \count@ 0#1\relax\@nil
2624 }
```

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

```
2625 \def\print@edition#1#2\@nil{%
2626     \ifx\relax#1\relax
2627         \ifnum\count@>\z@
2628             \CardinalNumeric\count@
2629         \else
2630             ??th%
2631         \fi
2632         \ \editiontext
2633     \else
2634         \ifnum \count@>\z@ \number\count@ \fi
2635         #1#2\relax
2636     \fi
2637 }
```

`\editiontext`

```
2638 \newcommand{\editiontext}{ed.}
```

`\CardinalNumber`

```
2639 \newcommand{\CardinalNumeric}[1]{%
2640     \number#1\relax
```



```

2641 \if
2642   \ifnum#1<14
2643     \ifnum#1>\thr@@ T\else F\fi
2644   \else
2645     F%
2646   \fi
2647   T%
2648   th%
2649 \else
2650   \@xp\keep@last@digit\@xp#1\number#1\relax
2651   \ifcase#1th\or st\or nd\or rd\else th\fi
2652 \fi
2653 }

```

\keep@last@digit

```

2654 \def\keep@last@digit#1#2{%
2655   \ifx\relax#2%
2656     \@xp\@gobbletwo
2657   \else
2658     #1=#2\relax
2659   \fi
2660   \keep@last@digit#1%
2661 }

```

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

```
2662 \newcommand{\SentenceSpace}{\relax\ifhmode\spacefactor'\. \fi}
```

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

```
2663 \newcommand{\eprint}[1]{\url{#1}}
```

The www.arXiv.org recommendations for citing their eprints are found at <http://xxx.lanl.gov/help/faq/references>, including these examples:

```

arXiv:hep-th/9910001
arXiv:math.AT/9910001
arXiv:physics.acc-ph/9911027

```

6.25 Bib_T_EX support

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

```
2664 \let\bibliographystyle\@gobble
```

\bibtex@style

```

2665 \def\bibtex@style{amsrn}

2666 \AtBeginDocument{
2667   \if@files
2668     \immediate\write\@auxout{\string\bibstyle{\bibtex@style}}%
2669   \fi
2670 }

```

6.26 Implementing package options

6.26.1 The alphabetic option

```

2671 \IfOption{alphabetic}{%
2672   \def\bibtex@style{amsra}%
2673   \def\alpha@label{%
2674     \ifx\@empty\bib'label
2675     \def\thebib{\CurrentBib}%
2676   \else

```

```

2677         \let\thebib\bib'label
2678     \fi
2679 }%
2680 \let\generate@label\generate@alphalabel
2681 \let\process@citelist\process@citelist@unsorted
2682 \def\numeric@refs{01}%
2683 }{}

```

6.26.2 The shortalphabetic option

```

2684 \IfOption{shortalphabetic}{%
2685     \def\bibtex@style{amsrs}%
2686     \def\alpha@label{%
2687         \ifx\@empty\bib'label
2688             \def\thebib{\CurrentBib}%
2689         \else
2690             \let\thebib\bib'label
2691         \fi
2692     }%
2693     \let\@suffix@format\@arabic
2694     \def\calc@author@part{%
2695         \@xp\@multiauthorlabel\@xp{\@tempa}%
2696     }%
2697     \let\append@label@year\@empty
2698     \let\generate@label\generate@alphalabel
2699     \let\process@citelist\process@citelist@unsorted
2700     \def\numeric@refs{01}%
2701 }{}

```

6.26.3 The backrefs option

Rather than checking for the `backrefs` option *per se*, we check to see if the `backref` package is loaded. This accomodates authors who load the `backref` package explicitly but do not pass the `backrefs` option to `amsrefs`.

```

2702 \AtBeginDocument{%
2703     \@ifpackageloaded{backref}{%
2704         \let\PrintBackRefs\print@backrefs
2705         \let\BackCite\back@cite

```

The `backref` package uses `\@starttoc` inside `\thebibliography` to open and read the `.brf` file. We could do something similar with `\biblist`, but it seems cleaner to use `\AtBeginDocument`. Unfortunately, `amsart` redefines `\@starttoc` in a way that interacts badly with this use. So, we inline the relevant parts of `\@starttoc` here. (The group and `\makeatletter` are unnecessary at present, but I'll leave them in as future-proofing.)

```

2706     \begingroup
2707         \makeatletter
2708         \@input{\jobname.brf}%
2709         \if@files
2710             \newwrite\tf@brf
2711             \immediate\openout\tf@brf \jobname.brf\relax
2712         \fi
2713     \endgroup
2714 }{}%
2715 }

```

6.26.4 The citation-order option

```

2716 \IfOption{citation-order}{%
2717     \IfOption{alphabetic}{%
2718         \amsrefs@warning@nl{%
2719             The citation-order and alphabetic options are
2720             incompatible%

```

```

2721     }%
2722   }{
2723     \def\bibtex@style{amsru}%
2724   }
2725 }{}

```

6.26.5 The initials option

```

2726 \IfOption{initials}{% TRUE:
2727   \BibSpec{nameLE}{
2728     +{}{}{initials}
2729     +{}{\IfEmptyBibField{initials}{}{ }}{surname}
2730     +{}{ }{jr}
2731   }
2732
2733   \BibSpec{nameBE}{
2734     +{}{}{surname}
2735     +{}{ }{initials}
2736     %   +{}{ }{jr}
2737   }
2738
2739   \BibSpec{nameinverted}{
2740     +{} {} {surname}
2741     +{,}{ } {initials}
2742     +{,}{ } {jr}
2743   }
2744 }{% initials? FALSE:
2745 %   \let\extract@initials@gobble
2746 } % end conditional code for initials option

```

6.26.6 The logical-quotes option

\deferredquotes

```

2747 \let\deferredquotes\@empty

```

\deferredquoteslogical

```

2748 \IfOption{logical-quotes}{%
2749   \def\deferredquoteslogical{\deferredquotes}%
2750 }{%
2751   \let\deferredquoteslogical\relax
2752 }

```

6.26.7 The non-compressed-cites option

```

2753 \IfOption{non-compressed-cites}{%
2754   \let\cite@compress\cite@print
2755 }{}

```

6.26.8 The non-sorted-cites option

```

2756 \IfOption{non-sorted-cites}{%
2757   \let\process@citelist\process@citelist@unsorted
2758 }{}

```

6.26.9 The short-journals option

```

2759 \IfOption{short-journals}{%
2760   \renewcommand{\DefineJournal}[4]{%
2761     \bib*{#1}{periodical}{
2762       issn={#2},
2763       journal={#3},
2764     }%
2765   }
2766 }{}

```

```

2767 \IfOption{short-publishers}{%
2768   \renewcommand{\DefinePublisher}[4]{%
2769     \bib*{#1}{publisher}{%
2770       publisher={#2},%

```

```

2771         address={#4},
2772     }%
2773 }%
2774 }-{}

```

```

2775 \IfOption{short-months}{%
2776     \renewcommand{\bib@monthname}{%
2777         \ifcase 0\bib@month
2778         \or Jan.\or Feb.\or Mar.\or Apr.\or May\or June\or
2779         July\or Aug.\or Sep.\or Oct.\or Nov.\or Dec.\or
2780         Winter\or Spring\or Summer\or Fall\else Unknown Month%
2781         \fi
2782     }%
2783 }{}

```

```

2784 \IfOption{y2k}{%
2785   \IfOption{alphabetic}{%
2786     \def\year@short#1\@nil{#1}%
2787     \def\bibtex@style{amsry}%
2788   }{%
2789     \amsrefs@warning@nl{%
2790       The y2k option can only be used with the^^J
2791       alphabetic option%
2792     }%
2793   }
2794 }{}

```

```

2795 \IfOption{bibtex-style}{%
2796     \RequirePackage{amsbst}
2797 }{}

```

Unless you're using pdf_TE_X, links cannot be broken across lines, which causes problems for long-form MR numbers such as “MR2149145 (2006d:01012)”. To mitigate the problem, we manually break such numbers into two separate links.

```

2799 \@ifundefined{href}{-}%
2800 \def\parse@MR#1 (#2)#3\@nil{%
2801 \def\MR@url{http://www.ams.org/mathscinet-getitem?mr=#1}%
2802 \def\@tempd{#1}%
2803 \def\@tempe{#2}%
2804 }%
2805 \def\MRhref#1#2{%
2806 \begingroup
2807 \parse@MR#1 ()\@empty\@nil%
2808 \href{\MR@url}{\@tempd\vphantom{()}}%
2809 \ifx\@tempe\@empty
2810 \else
2811 \ \href{\MR@url}{(\@tempe)}%

```

```

2812             \fi
2813         \endgroup
2814     }%
2815 }{}
2816 }{}

```

6.26.15 The author-year option

Here ends the `amsrefs` package, unless the `author-year` option is in effect; then we want to use some different bibspecs.

```

2817 \IfOption{author-year}{\PopCatcodes \endinput}

```

`\generate@label`

```

2818 \def\generate@label{%

```

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

```

2819     \ifx\bib'label\@empty
2820         \begingroup

```

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

```

2821         \global\let\previous@stem\current@stem
2822         \global\let\current@stem\@empty
2823         \global\let\previous@year\current@year
2824         \global\let\current@year\bib@year

```

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

```

\name{Last1,First1} \name{Last2,First2} ... \name{Lastn,Firstn}

```

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

```

2825         \@apply\auto@protect\amsrefs@textsymbols
2826         \@apply\auto@protect\amsrefs@textaccents
2827         \def\name##1{\@nx\name{\lscan@a##1,\@nil}}%
2828         \auto@protect\etaltext
2829         \normalize@edef\current@stem{\current@primary}%
2830         \xdef\current@stem{\current@stem}%

```

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

```

2831         \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.)

```

2832         \edef\@tempa{%
2833             \def\@nx\cite@label{\current@stem}%
2834             \def\@nx\bib@label@year{%
2835                 \current@year
2836                 \alpha@label@suffix
2837             }%
2838         }
2839     \@xp\endgroup
2840     \@tempa
2841 \fi
2842 }

```

```

\lnscan@a
2843 \def\lnscan@a#1,#2\@nil{#1}

\citesel@author
2844 \def\citesel@author#1#2#3#4#5{\PrintCiteNames{#3}}

\citesel@authoryear
2845 \def\citesel@authoryear#1#2#3#4#5{\PrintCNY{#3}{#4}}

\citesel@object
2846 \def\citesel@object#1#2#3#4#5{\PrintCiteNames{#3} \citeleft#4}

\citesel
2847 \let\citesel\citesel@authoryear

\numeric@refs
2848 \def\numeric@refs{01}%

\citeleft
2849 \def\citeleft{()%

\citeright
2850 \def\citeright{)%

\@citeleft
2851 \def\@citeleft{\ifx\citesel\citesel@object\else\citeleft\fi}%

\citepunct
2852 \def\citepunct{; }

\BibLabel
2853 \def\BibLabel{%
2854   \hyper@anchorstart{cite.\CurrentBib}\relax\hyper@anchorend
2855 }

\process@citelist
2856 \let\process@citelist\process@citelist@unsorted

\ycite
2857 \DeclareRobustCommand{\ycite}[1]{%
2858   \star@{\cite@a\citesel@year{#1}}{}}%
2859 }

\ycites
2860 \DeclareRobustCommand{\ycites}[1]{%
2861   \begingroup
2862     \def\citepunct{, }%
2863     \let\citesel\citesel@year
2864     \cites{#1}%
2865   \endgroup
2866 }

\ocite
2867 \DeclareRobustCommand{\ocite}[1]{%
2868   \star@{\cite@a\citesel@object{#1}}{}}%
2869 }

```

`\ocites`

```
2870 \DeclareRobustCommand{\ocites}[1]{%
2871   \begingroup
2872     \let\@citelist\@ocitelist
2873     \cites{#1}%
2874   \endgroup
2875 }
```

`\ocitelist`

```
2876 \def\@ocitelist#1{%
2877   \PrintSeries{\InnerCite}%
2878   {\ocite}%
2879   {}{ 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.

```
2880   {},{ \ocite}%
2881   {},{ and \ocite}%
2882   {}%
2883   {#1}%
2884   {}%
2885 }
```

`\citeauthor`

```
2886 \DeclareRobustCommand{\citeauthor}[1]{%
2887   \star@\cite@a\citesel@author{#1}}{}%
2888 }
```

`\citeauthory`

```
2889 \DeclareRobustCommand{\citeauthory}[1]{%
2890   \citeauthor{#1} \ycite{#1}%
2891 }
```

`\fullcite`

```
2892 \DeclareRobustCommand{\fullcite}[1]{%
2893   \begingroup
2894     \let\print@citenames\CiteNamesFull
2895     \star@\cite@a\citesel@authoryear{#1}}{}%
2896   \endgroup
2897 }
```

`\fullocite`

```
2898 \DeclareRobustCommand{\fullocite}[1]{%
2899   \begingroup
2900     \let\print@citenames\CiteNamesFull
2901     \star@\cite@a\citesel@object{#1}}{}%
2902   \endgroup
2903 }
```

Invert the first author’s name.

```
2904 \def\set@firstname#1{%
2905   \set@name{#1}\setbib@nameinverted
2906 }
```

`\PrintCNY`

```
2907 \def\PrintCNY#1#2{%
```

```

2908 \PrintCiteNames{#1}%
2909 \@ifnotempty{#2}{\@addpunct{,} #2}%
2910 }

```

\PrintCiteNames

```

2911 \def\PrintCiteNames#1{%
2912 \leavevmode
2913 \def\@tempa{#1}%
2914 \ifx\@tempa\prev@names
2915 \else
2916 \gdef\prev@names{#1}%
2917 \@xp\ifx\@car#1.\@nil\CitePrintUndefined
2918 #1\relax
2919 \else
2920 \print@citenames{#1}%
2921 \fi
2922 \fi
2923 }

```

\CiteNames

```

2924 \newcommand{\CiteNames}[1]{%
2925 \PrintSeries{name}%
2926 {}%
2927 {}{ 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.

```

2928 {}{\@gobble}%
2929 {}{ \etaltext\@gobble}%
2930 {}%
2931 {#1}%
2932 {}%
2933 }

```

\print@citenames

```

2934 \let\print@citenames\CiteNames

```

\CiteNamesFull

```

2935 \newcommand{\CiteNamesFull}[1]{%
2936 \PrintSeries{name}%
2937 {}%
2938 {}{ 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.

```

2939 {,}{ }%
2940 {,}{ and }%
2941 {}%
2942 {#1}%
2943 {}%
2944 }

```

\PrintDate No parentheses around the year.

```

2945 \renewcommand{\PrintDate}[1]{\bib@label@year}

```


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

```

2946 \def\print@date{%
2947   \IfEmptyBibField{date}{%
2948     \IfEmptyBibField{year}{\BibField{status}}{\bib@year}%
2949   }{%
2950     \bib@year
2951   }%
2952 }

2953 \BibSpec{article}{%
2954   +{} {\PrintAuthors}           {author}
2955   +{.} { \PrintDate}           {date}
2956   +{.} { \textit}             {title}
2957   +{.} { }                     {part}
2958   +{:} { \textit}             {subtitle}
2959   +{,} { \PrintContributions}  {contribution}
2960   +{.} { \PrintPartials}      {partial}
2961   +{,} { }                     {journal}
2962   +{} { \textbf}              {volume}
2963   +{,} { \issuetext}          {number}
2964   +{,} { \eprintpages}        {pages}
2965   +{,} { }                     {status}
2966   +{,} { \PrintDOI}           {doi}
2967   +{,} { available at \eprint} {eprint}
2968   +{} { \parenthesize}        {language}
2969   +{} { \PrintTranslation}     {translation}
2970   +{;} { \PrintReprint}        {reprint}
2971   +{.} { }                     {note}
2972   +{.} {}                      {transition}
2973   +{} {\SentenceSpace \PrintReviews} {review}
2974 }
2975
2976 \BibSpec{book}{%
2977   +{} {\PrintPrimary}          {transition}
2978   +{.} { \PrintDate}          {date}
2979   +{.} { \textit}             {title}
2980   +{.} { }                     {part}
2981   +{:} { \textit}             {subtitle}
2982   +{,} { \PrintEdition}        {edition}
2983   +{} { \PrintEditorsB}        {editor}
2984   +{,} { \PrintTranslatorsC}   {translator}
2985   +{,} { \PrintContributions}  {contribution}
2986   +{,} { }                     {series}
2987   +{,} { \voltext}            {volume}
2988   +{,} { }                     {publisher}
2989   +{,} { }                     {organization}
2990   +{,} { }                     {address}
2991   +{,} { }                     {status}
2992   +{} { \parenthesize}        {language}
2993   +{} { \PrintTranslation}     {translation}
2994   +{;} { \PrintReprint}        {reprint}
2995   +{.} { }                     {note}
2996   +{.} {}                      {transition}
2997   +{} {\SentenceSpace \PrintReviews} {review}
2998 }
2999
3000 \BibSpec{collection.article}{%
3001   +{} {\PrintAuthors}          {author}
3002   +{.} { \PrintDate}          {date}

```

```

3003      +{.} { \textit}           {title}
3004      +{.} { }                   {part}
3005      +{:} { \textit}           {subtitle}
3006      +{,} { \PrintContributions} {contribution}
3007      +{,} { \PrintConference}   {conference}
3008      +{ } { \PrintBook}         {book}
3009      +{,} { }                   {booktitle}
3010      +{,} { pp.~}               {pages}
3011      +{,} { }                   {status}
3012      +{,} { \PrintDOI}         {doi}
3013      +{,} { available at \eprint} {eprint}
3014      +{ } { \parenthesize}      {language}
3015      +{ } { \PrintTranslation}   {translation}
3016      +{;} { \PrintReprint}      {reprint}
3017      +{.} { }                   {note}
3018      +{.} {}                   {transition}
3019      +{ } { \SentenceSpace \PrintReviews} {review}
3020 }
3021
3022 \BibSpec{report}{%
3023      +{ } { \PrintPrimary}       {transition}
3024      +{.} { \PrintDate}         {date}
3025      +{.} { \textit}           {title}
3026      +{.} { }                   {part}
3027      +{:} { \textit}           {subtitle}
3028      +{,} { \PrintEdition}      {edition}
3029      +{,} { \PrintContributions} {contribution}
3030      +{,} { Technical Report }   {number}
3031      +{,} { }                   {series}
3032      +{,} { }                   {organization}
3033      +{,} { }                   {address}
3034      +{,} { \eprint}           {eprint}
3035      +{,} { }                   {status}
3036      +{ } { \parenthesize}      {language}
3037      +{ } { \PrintTranslation}   {translation}
3038      +{;} { \PrintReprint}      {reprint}
3039      +{.} { }                   {note}
3040      +{.} {}                   {transition}
3041      +{ } { \SentenceSpace \PrintReviews} {review}
3042 }
3043
3044 \BibSpec{thesis}{%
3045      +{ } { \PrintAuthors}       {author}
3046      +{.} { \PrintDate}         {date}
3047      +{.} { \textit}           {title}
3048      +{:} { \textit}           {subtitle}
3049      +{,} { \PrintThesisType}    {type}
3050      +{,} { }                   {organization}
3051      +{,} { }                   {address}
3052      +{,} { \eprint}           {eprint}
3053      +{,} { }                   {status}
3054      +{ } { \parenthesize}      {language}
3055      +{ } { \PrintTranslation}   {translation}
3056      +{;} { \PrintReprint}      {reprint}
3057      +{.} { }                   {note}
3058      +{.} {}                   {transition}
3059      +{ } { \SentenceSpace \PrintReviews} {review}
3060 }
3061 \PopCatcodes

```

3062 $\langle /pkg \rangle$

6.27 The amsbst package

3063 $\langle *bst \rangle$

3064 $\backslash NeedsTeXFormat\{LaTeX2e\}[1995/12/01]$

3065 $\backslash ProvidesPackage\{amsbst\}[2004/03/29\ v1.68]$

3066 $\% \backslash RequirePackage\{amsrefs\}[2004/03/29]$

```

3067 \BibSpec{article}{%
3068     +{ } {\PrintAuthors}           {author}
3069     +{.} { }                       {title}
3070     +{.} { }                       {part}
3071     +{:} { }                       {subtitle}
3072     +{.} { \PrintContributions}     {contribution}
3073     +{.} { \PrintPartials}          {partial}
3074     +{.} { \emph}                  {journal}
3075     +{ } { }                       {volume}
3076     +{ } { \parenthesize}          {number}
3077     +{:} { }                       {pages}
3078     +{,} { \PrintDateB}             {date}
3079     +{,} { }                       {status}
3080     +{.} { \PrintTranslation}       {translation}
3081     +{.} { Reprinted in \PrintReprint} {reprint}
3082     +{.} { }                       {note}
3083     +{.} { }                       {transition}
3084 }
3085
3086 \BibSpec{partial}{%
3087     +{ } { }                       {part}
3088     +{:} { }                       {subtitle}
3089     +{.} { \PrintContributions}     {contribution}
3090     +{.} { \emph}                  {journal}
3091     +{ } { }                       {volume}
3092     +{ } { \parenthesize}          {number}
3093     +{:} { }                       {pages}
3094     +{,} { \PrintDateB}            {date}
3095 }
3096
3097 \BibSpec{book}{%
3098     +{ } {\PrintPrimary}           {transition}
3099     +{.} { \emph}                  {title}
3100     +{.} { }                       {part}
3101     +{:} { \emph}                  {subtitle}
3102     +{.} { }                       {series}
3103     +{,} { \voltext}               {volume}
3104     +{.} { Edited by \PrintNameList} {editor}
3105     +{.} { Translated by \PrintNameList} {translator}
3106     +{.} { \PrintContributions}     {contribution}
3107     +{.} { }                       {publisher}
3108     +{.} { }                       {organization}
3109     +{,} { }                       {address}
3110     +{,} { \PrintEdition}          {edition}
3111     +{,} { \PrintDateB}            {date}
3112     +{.} { }                       {note}
3113     +{.} { }                       {transition}
3114     +{.} { \PrintTranslation}       {translation}
3115     +{.} { Reprinted in \PrintReprint} {reprint}
3116     +{.} { }                       {transition}
3117 }
3118
```

```

3119 \BibSpec{collection.article}{%
3120     +{} {} \PrintAuthors} {author}
3121     +{.} { } {title}
3122     +{.} { } {part}
3123     +{:} { } {subtitle}
3124     +{.} { } \PrintContributions} {contribution}
3125     +{.} { } \PrintConference} {conference}
3126     +{.} { } \PrintBook} {book}
3127     +{.} { } In } {booktitle}
3128     +{,} { } pages~} {pages}
3129     +{.} { } \PrintDateB} {date}
3130     +{.} { } \PrintTranslation} {translation}
3131     +{.} { } Reprinted in \PrintReprint} {reprint}
3132     +{.} { } {note}
3133     +{.} {} {transition}
3134 }
3135
3136 \BibSpec{conference}{%
3137     +{} {} {title}
3138     +{} {} \PrintConferenceDetails} {transition}
3139 }
3140
3141 \BibSpec{innerbook}{%
3142     +{.} { } \emph} {title}
3143     +{.} { } {part}
3144     +{:} { } \emph} {subtitle}
3145     +{.} { } {series}
3146     +{,} { } \voltext} {volume}
3147     +{.} { } Edited by \PrintNameList} {editor}
3148     +{.} { } Translated by \PrintNameList} {translator}
3149     +{.} { } \PrintContributions} {contribution}
3150     +{.} { } {publisher}
3151     +{.} { } {organization}
3152     +{,} { } {address}
3153     +{,} { } \PrintEdition} {edition}
3154     +{,} { } \PrintDateB} {date}
3155     +{.} { } {note}
3156     +{.} {} {transition}
3157 }
3158
3159 \BibSpec{report}{%
3160     +{} {} \PrintPrimary} {transition}
3161     +{.} { } \emph} {title}
3162     +{.} { } {part}
3163     +{:} { } \emph} {subtitle}
3164     +{.} { } \PrintContributions} {contribution}
3165     +{.} { } Technical Report } {number}
3166     +{,} { } {series}
3167     +{.} { } {organization}
3168     +{,} { } {address}
3169     +{,} { } \PrintDateB} {date}
3170     +{.} { } \PrintTranslation} {translation}
3171     +{.} { } Reprinted in \PrintReprint} {reprint}
3172     +{.} { } {note}
3173     +{.} {} {transition}
3174 }
3175
3176 \BibSpec{thesis}{%
3177     +{} {} \PrintAuthors} {author}

```

```

3178 +{,} { \emph} {title}
3179 +{:} { \emph} {subtitle}
3180 +{.} { \PrintThesisType} {type}
3181 +{.} { } {organization}
3182 +{,} { } {address}
3183 +{,} { \PrintDateB} {date}
3184 +{.} { \PrintTranslation} {translation}
3185 +{.} { Reprinted in \PrintReprint} {reprint}
3186 +{.} { } {note}
3187 +{.} {} {transition}
3188 }

```

`\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.

```

3189 \renewcommand{\PrintEditorsA}[1]{%
3190   \def\current@bibfield{\bib'editor}%
3191   \PrintNames{},{, editor\Plural{s)}}{#1}%
3192   \erase@field\bib'editor
3193 }

```

`\PrintTranslatorsA`

```

3194 \renewcommand{\PrintTranslatorsA}[1]{%
3195   \def\current@bibfield{\bib'translator}%
3196   \PrintNames{},{, translator\Plural{s)}}{#1}%
3197   \erase@field\bib'translator
3198 }

3199 </bst>

```

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

```
3200 \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> 57, 1684	10, 10, 43, 53, 54, 54, 54, 55, 55	<code>\@MM</code> .. 2239, 2266, 2288
<code>\&</code> 25	<code>.bbl</code> .. 8, 8, 9, 25, 27, 41	<code>\@addpunct</code> .. 6, 14, 14, 107, 1347, 2909
<code>\'</code> 40, 57, 1685	<code>.bib</code> .. 8	<code>\@alph</code> .. 2164
<code>\+</code> 24	<code>.brf</code> .. 90	<code>\@apply</code> .. 68
<code>\,</code> 24, 119, 2437	<code>.dtx</code> .. 8	<code>\@arabic</code> .. 2693
<code>\-</code> 24	<code>.ltb</code> .. 25, 27, 41	<code>\@bibdef</code> 26, 26, 573,
<code>\.</code>	.. 24, 114, 1686, 2662	<code>\:</code> .. 24, 117, 622, 638	583, 588, 600,
<code>.aux</code>	. 8, 8, 8, 8, 9, 9, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10,	<code>\;</code> .. 24, 118	1160, 1163, 1179
		<code>\=</code> .. 24, 1687	<code>\@biblist</code> 486, 491, 495
		? option .. 16	<code>\@car</code> . 2101, 2279, 2917
		<code>\@undefined</code> 1384	<code>\@carcube</code> .. 2102

- \@chomp .. 85, 1879, 1891, 1944, 1945
- \@chomper 14, 14, 88, 102
- \@citeleft 1411, 1419, 2851
- \@citelist 46, 1334, 1335, 2872
- \@citestyle . 1338, 1355
- \@citex 48
- \@close@bbl@file . 1176, 1248, 1257, 1267
- \@concat 59, 2069
- \@count@chars 2053, 2058
- \@currentlabel ... 1005, 1047
- \@current 135
- \@currname 137
- \@empty 13, 18, 18, 19, 42, 43
- \@endbiblist 488, 493, 515
- \@extract@initial 70, 2019, 2021, 2033
- \@file@und .. 1193, 1196, 1199, 1213
- \@firstofone 15, 36, 981, 1311, 1333, 1376
- \@firstoftwo 48, 995, 2222
- \@firstone . 2101, 2103
- \@firstthree 2102, 2125
- \@for 33
- \@gobblefour 690
- \@gobblethree ... 1429
- \@ifclassloaded .. 1104, 1105, 1106
- \@ifempty 45, 267, 1063, 1280, 1305, 1391, 1564, 1570, 1586, 1638
- \@ifnextchar 18
- \@ifnotempty 45, 238, 655, 679, 1292, 1413, 2106, 2909
- \@ifpackageloaded 1662, 1665, 1670, 2703
- \@ifpackagewith .. 5, 8, 1671
- \@ifsame@patterns 984, 990
- \@ifsamepat .. 991, 993
- \@input@ 1180
- \@lappend 61, 1682, 1711, 1712, 1714, 1715
- \@latex@warning .. 517, 1399
- \@let@token 64, 70
- \@listctr 504
- \@m 110, 509, 548
- \@marknames . 2105, 2117
- \@maxlabelwidth 502, 520, 523, 524, 532
- \@mkboth 39, 1108
- \@multiauthorlabel . 2098, 2135, 2695
- \@nametoken .. 2103, 2111, 2123, 2125
- \@nameuse 60
- \@nil 13, 36, 61, 63, 67, 67, 68
- \@nilgobble 63, 82, 1821, 2116
- \@nmbrlisttrue ... 503
- \@noitemerr 517
- \@nopunctsfcodes .. 122, 123, 2281
- \@ocitelist . 2872, 2876
- \@oneauthorlabel . 2096, 2120
- \@open@bbl@file 613, 1173, 1247, 1263
- \@pass@options 134
- \@prev@cite 1541
- \@scanforMR 2524
- \@secondoftwo 48, 997, 2224
- \@sptoken ... 1833, 1908, 1958, 2006
- \@starttoc . 90, 90, 90
- \@suffix@format .. 2164, 2175, 2186, 2693
- \@tabacckludge . 60, 1741, 1753, 1772
- \@tempa 64
- \@tempcnta 13, 53
- \@tempcntb 53
- \@tempd 2802, 2808
- \@tempdima 522, 523, 524
- \@tempe 2803, 2809, 2811
- \@tempswafalse ... 576, 1038, 2166
- \@tempswatrue . 572, 1056, 1059, 2170
- \@temptokena 64
- \@testdef 39
- \@threeauthors ... 2131, 2145
- \@uclclist 58, 58
- \@urlbreak 25, 27
- \@urlfix 24, 32
- \@urlsetup 20, 22
- \@xifempty 45, 47
- \^ 106, 1688
- \' 1689
- \~ 29, 1690
- _ ... 1332, 2632, 2811
- A**
- \A 546
- abbrv BibTeX style .. 15
- \add@toks@ 60
- address field 3, 3, 3, 40, 40, 92
- \addto@defer@list 701, 705
- \addto@hook ... 60, 819
- \AE 1720
- \ae 58, 58, 1720
- \after@deleting@token 41, 1840, 1850, 1918, 1956, 1972, 2015
- \afterassignment 12, 42, 82, 712, 2622
- \aftergroup ... 19, 290
- alpha BibTeX style .. 14
- \alpha@label . 1006, 1009, 2673, 2686
- \alpha@label@suffix 2087, 2160, 2175, 2182, 2186, 2836
- \alpha@suffix 2162, 2163, 2174, 2175, 2176, 2183, 2186
- alphabetic option . 14, 72, 75, 89
- amsart class 90
- amsbook class 46
- amsbst package 99
- amsgen package 14
- amsproc class 46
- amsrdoc.tex 2
- AMSREFS package . 5, 6, 7
- amsrefs package ... 1, 2, 2, 2, 2, 3, 3, 4, 4, 4, 5, 5, 5, 6, 7, 9, 9, 16, 41, 54, 78, 90, 93, 101, 101
- \amsrefs@bibcite . 1609, 1614, 1618
- \amsrefs@error ... 105, 257, 307, 593
- \amsrefs@textaccents 58,

- 849, 1037, 1680,
 1682, 1740,
 1754, 1771,
 1872, 2076, 2826
 \amsrefs@textsymbols
 60, 848, 1036,
 1709, 1711,
 1712, 1714,
 1715, 1739,
 1755, 1769,
 1871, 2075, 2825
 \amsrefs@warning .
 104, 892,
 893, 1070, 1084,
 1184, 1210,
 1288, 1293,
 1298, 1481,
 1865, 1937, 2351
 \amsrefs@warning@nl
 .. 4, 6, 9, 157,
 167, 2718, 2789
 amsxport BibTeX style
 72, 72
 \append@label@year
 . 2082, 2150, 2697
 \append@to@stem ..
 .. 2069, 2127,
 2133, 2143, 2153
 \ar@hyperlink
 . 1412, 1657, 1666
 \ar@SK@cite .. 1410,
 1660, 1672, 1675
 article entry type . 4, 7
 \AtBeginDocument 90,
 588, 589, 1610,
 1661, 2666, 2702
 \AtEndDocument .. 1100
 author field 5, 5
 author-year option
 46,
 75, 80, 80, 93, 93
 \AuthorList 7
 \auto@protect 12, 12,
 12, 57, 60, 848,
 849, 1018, 1035,
 1036, 1037,
 1739, 1754,
 1762, 1769,
 1868, 1869,
 1871, 1872,
 2075, 2076,
 2077, 2078,
 2825, 2826, 2828
B
 \b 1691
 \b@ 37, 47, 48,
 48, 51, 55, 55, 55
 \b@* 1471
 \b@aaa 52
 \b@bbb 52
 \b@moo 8, 9, 43
 \b@whatever .. 9, 10, 48
 \b@xyz 10
 \b@zzz 52
 babel package 36
 \back@cite . 1646, 2705
 \BackCite ... 1364,
 1443, 1645, 2705
 backref package ...
 90, 90, 90
 backrefs option ...
 9, 90, 90, 90
 \bbl@copy ... 618, 672
 \bbl@copy@a .. 676, 678
 \bbl@out 1245,
 1255, 1256,
 1258, 1266, 1269
 \bbl@write
 615, 619, 682,
 1246, 1258, 1268
 \bblname 1167, 1189, 1256
 \begin 9, 9, 9
 \bgroup
 89, 1790, 1858,
 1921, 1975, 2018
 \bi@ 29
 \bib 3, 3, 3, 5, 7,
 7, 7, 10, 10, 10,
 10, 25, 36, 40,
 41, 42, 42, 42,
 42, 43, 47, 60, 569
 \bib' 33
 \bib'bar 19
 \bib'date 35
 \bib'given 62, 63
 \bib'initials 63
 \bib'jr 63
 \bib'language 35
 \bib'pages 18
 \bib'surname 63
 \bib* 25, 27
 \bib@append
 ... 18, 19, 19,
 19, 245, 258, 2466
 \bib@author 37
 \bib@bar 19
 \bib@cite 37, 725, 1007
 \bib@cite@a . 1012, 1015
 \bib@cite@check ..
 1027, 1030
 \bib@citeyear 2152, 2156
 \bib@day
 952, 959, 960, 2609
 \bib@dbfile
 .. 1190, 1193,
 1196, 1199,
 1218, 1221, 1224
 \bib@div@mark
 . 1103, 1118, 1127
 \bib@end .. 31, 742, 759
 \bib@exec .. 11, 26,
 29, 589, 597, 602
 \bib@field@patches
 721, 748,
 903, 2449, 2463
 \bib@label 73, 73, 93, 93
 \bib@label@year ..
 .. 1021, 1054,
 1068, 1075,
 1082, 2834, 2945
 \bib@language
 . 35, 973, 975,
 978, 980, 981,
 984, 985, 986, 1000
 \bib@language@fixup
 948, 970
 \bib@mklab .. 505, 521
 \bib@month
 951, 958, 961,
 2601, 2613, 2777
 \bib@monthname ...
 . 2608, 2612, 2776
 \bib@parsedate ...
 930, 933, 953
 \bib@parsedate@a .
 954, 956
 \bib@print
 .. 26, 30, 589, 716
 \bib@print@inner .
 ... 744, 2395,
 2414, 2440, 2494
 \bib@reprint 2443, 2445
 \bib@reset 853
 \bib@resolve@xrefs
 720, 747,
 767, 2448, 2462
 \bib@selectlanguage
 722, 983
 \bib@start
 .. 73, 93, 717, 755
 \bib@store 26, 602, 711
 \bib@translation .
 2457, 2459
 \bib@year . 37, 928,
 937, 950, 957,
 962, 2151, 2154,
 2605, 2619,
 2824, 2948, 2950
 \BibAbbrevWarning
 801, 836, 892
 bibchapter environ-
 ment 1113
 \bibchapter 1134
 \biblecite . 9, 9, 9, 9,
 9, 9, 9, 10, 10,
 10, 10, 39, 54,

- 54, 54, 1098,
 1100, 1614, 1618
 \bibcite@a . 1609, 1620
 \bibcite@b . 1625, 1629
 \bibcite@write 1013, 1088
 \bibdata 8, 9
 \bibdefer@list 704,
 708, 1177, 1178
 bibdiv environment 1131
 \bibdiv 1137
 \BibField 7, 7, 184, 2948
 \BibItem . 574, 577, 580
 \bibitem 9, 9, 9
 \BibLabel 500, 556, 560, 2853
 \biblanguage@default 973, 1000
 \biblanguage@English 1000
 \bibliography . . 8, 8, 9
 \bibliographystyle 8, 89, 2664
 biblist environment . . . 1, 23, 23, 484
 \biblist 90, 1138
 biblist* environment 490
 \bibname 39, 1101, 1113
 \bibquotes 3, 6, 6, 6, 2515
 bibsection environ-
 ment 1122
 \bibsection 1132
 \BibSelect 1161, 1164, 1167
 \bibselect 25, 28, 29, 50,
 1158, 1185, 1186
 \bibselect* 27, 50
 \bibselect@msg . . 1170
 \BibSpec . . . 3, 4, 4,
 5, 5, 5, 6, 7, 7,
 7, 8, 18, 18, 18, 236
 \bibspeg@scan 18, 18, 18, 239, 244
 \bibspeg@scan@error 253, 257
 \BibSpecAlias . . 20,
 304, 460, 461,
 462, 463, 464,
 465, 466, 467, 468
 \bibstyle . . 8, 9, 2668
 bibtex-style option . 92
 \bibtex@style 2665,
 2668, 2672,
 2685, 2723, 2787
 book class 40
 \bracket@stack 2500,
 2503, 2508, 2513
 bst 3
- \bud 1734
 \bysame 7, 7, 7, 2361, 2362
- C**
- \c 1692
 \c@bib@env 532, 537, 540
 \calc@alpha@suffix 2083, 2165, 2831
 \calc@author@part 2081, 2094, 2694
 \CardinalNumber . 2639
 \CardinalNumeric 8, 8, 2628, 2639
 \catcode 25
 \cdprime 1733
 \ch@ck 1260
 \chapter* 39
 \check@mathfonts . . 23
 \chomp@ 95, 100
 \citation 8, 8, 9, 9,
 9, 10, 10, 10, 1422
 citation-order op-
 tion 72, 90
 cite package 45
 \cite 8, 8, 8, 9, 9, 9,
 10, 10, 10, 37,
 43, 44, 44, 46,
 49, 49, 49, 49, 1278
 \cite@a . . 47, 1362,
 1363, 1469,
 2858, 2868,
 2887, 2895, 2901
 \cite@b . 47, 1373, 1380
 \cite@bc 47, 1381, 1383
 \cite@begingroup 1340, 1356,
 1365, 1442, 1449
 \cite@cb 1409,
 1420, 1458, 1545
 \cite@cj 47, 52, 53, 1406
 \cite@cjs . . 1493, 1554
 \cite@compress 1494, 1562,
 1582, 1606, 2754
 \cite@compress@a 1566, 1569, 1599
 \cite@compress@b 1575, 1585
 \cite@dash 53, 1503,
 1557, 1560,
 1590, 1594, 1602
 \cite@dest . 1630, 1631
 \cite@dest@b 1635, 1637
 \cite@endgroup 1345,
 1357, 1417,
 1444, 1450, 1464
 \cite@label 1005, 1021,
 1047, 1048,
- 1050, 1065,
 1068, 1075, 2833
 \cite@nobib@test 1386, 1390
 \cite@print . . 1544,
 1551, 1563,
 1574, 1577,
 1581, 1592,
 1597, 1605, 2754
 \cite@single 1279, 1303
 \cite@sort . 1497, 1507
 \cite@sort@a 1511, 1514
 \cite@sort@b 1520, 1530
 \cite@sorted@e 52, 1474, 1500
 \cite@sorted@s 52, 1474, 1490
 \CiteAltPunct . . 45, 46
 \citeAltPunct 1332, 1343
 \citeauthor 1438, 2886, 2890
 \citeauthority 1439, 2889
 \citedest 9,
 10, 10, 55, 55, 1630
 \citeform 1333
 \citeleft 1328, 1339,
 1347, 1369,
 1419, 1452,
 2846, 2849, 2851
 \citelist 49, 1293, 1314,
 1334, 1416, 1445
 \citemid 45, 46, 1328, 1413
 \citen 1431
 \CiteNames . 2924, 2934
 \CiteNamesFull 2894, 2900, 2935
 \CitePrintUndefined 1039, 1395,
 1402, 1405,
 1492, 1553, 2917
 \citepunct 45, 1328,
 1339, 1343,
 1347, 2852, 2862
 \citeright 1328, 1348,
 1352, 1370,
 1414, 1453, 2850
 \cites . . 1289, 1430,
 1433, 1435,
 1446, 2864, 2873
 \cites@a 1284,
 1305, 1306, 1430
 \cites@b . . . 1308, 1317
 \cites@c . . . 1321, 1323
 \cites@init . . 1346,
 1358, 1366, 1451
 \cites@o . . . 1281, 1304

- `\cites@oo` .. 1304, [1305](#)
`\CITSEL` 47
`\citesel` 37,
 37, 43, 47, 48,
 55, 55, [1276](#), [2847](#)
`\citesel@author` ..
 1368, [2844](#), [2887](#)
`\citesel@authoryear`
 2845, [2847](#), [2895](#)
`\citesel@number` ..
 1274, [1276](#), [1473](#)
`\citesel@object` ..
 2846,
 2851, [2868](#), [2901](#)
`\citesel@other` ...
 1469, [1470](#)
`\citesel@update` ..
 1271, [1425](#)
`\citesel@write` ...
 1091, [1096](#)
`\citesel@year`
 1275, [2858](#), [2863](#)
`\CloseBBLFile` [1257](#), [1265](#)
`\closein` [1218](#)
`\clubpenalty` . 510, 511
`cmr font` 46
`cmsy font` 6
collection entry type 7
`\compare@stems` ...
 2167, [2191](#)
`\copy@bibdef`
 26, 50, [605](#),
 695, 1160, [1461](#)
`\copy@bibdef@a` ...
 609, [612](#), 658
`\count` 13, [1252](#)
`\count@` 88
`\count@chars` 2041, [2050](#)
`\cprime` 1732
`\CPU@normal` . [1405](#), [1553](#)
`\CPU@sort` .. [1489](#), [1492](#)
`\cs` 59, 60, 60
`\CurLine` 42,
 42, 43, [1224](#),
 1225, [1231](#), [1242](#)
`\current@bibfield`
 20, 266,
 294, [2357](#), [2374](#),
 2481, [3190](#), [3195](#)
`\current@citekey` .
 753, [757](#), 764, [2184](#)
`\current@primary` 72,
 81, 93, [894](#),
 900, 906, 909,
 913, [2079](#), [2344](#),
 2347, 2354,
 2371, [2478](#), [2829](#)
`\current@stem` .. 73,
 93, 2066, 2069,
 2073, 2074,
 2086, 2167,
 2821, 2822,
 2829, [2830](#), [2833](#)
`\current@year`
 75, 2068, 2169,
 2823, [2824](#), [2835](#)
`\CurrentBib` ... [555](#),
 557, 761, 1008,
 1086, 1098,
 2675, [2688](#), [2854](#)
`\CurrentBibType` 26, 591
`\cydot` 1735
- ## D
- `\d` [1693](#)
`\DashPages` . 6, 6, [2557](#)
`\dateLANGUAGE` 36
`\Dbar` [1729](#)
`\dbar` [1729](#)
`\DeclareBooleanOption`
 127, [128](#),
 129, 130, 131,
 132, [142](#), [143](#),
 144, [145](#), [146](#), [147](#)
`\DeclareExclusiveOptions`
 125,
 126, [139](#), [140](#), [141](#)
`\DeclareNameAccent`
 1681,
 1684, 1685,
 1686, 1687,
 1688, 1689,
 1690, 1691,
 1692, 1693,
 1694, 1695,
 1696, 1697,
 1698, 1699,
 1700, 1701,
 1702, 1703,
 1704, 1705,
 1706, [1707](#), [1708](#)
`\DeclareNameSymbol`
 1710,
 1718, 1719,
 1720, 1721,
 1722, 1723,
 1724, 1725,
 1726, 1727,
 1728, 1729,
 1730, 1731,
 1732, 1733,
 1734, [1735](#), [1736](#)
`\DeclareOption` ... 133
`\DeclareRobustCommand`
 18, [1334](#), [1402](#),
 1430, 2857,
 2860, 2867,
 2870, 2886,
 2889, [2892](#), [2898](#)
`\def` 20, 48, 49
`\defer@bibdef`
 607, [697](#), [1215](#)
`\deferredquotes` ..
 2287, [2517](#),
 2518, [2747](#), [2749](#)
`\deferredquoteslogical`
 2276, [2748](#)
`\DefineAdditiveKey`
 188,
 189, 190, 191,
 192, 193, 194, 195
`\DefineDummyKey` .. 233
`\DefineJournal` 3, 15,
 32, 40, [1146](#), [2760](#)
`\DefineName`
 3, 31, 40, 42,
 1143, [2489](#), [2490](#)
`\DefinePublisher` .
 15, 32,
 40, 40, [1152](#), [2768](#)
`\DefineSimpleKey` .
 196, 197,
 198, 199, 200,
 201, 202, 203,
 204, 205, 206,
 207, 208, 209,
 210, 211, 212,
 213, 214, 215,
 216, 217, 218,
 219, 220, 221,
 222, 223, 224,
 225, 226, 227,
 228, 229, 230,
 231, 232, 234, 235
`\DH` 1723
`\dh` 1723
`\DJ` 1724
`\dj` 1724
`\do` 12, 13, 16, 57
`docstrip` 101
`\dospecials` 13
`\DSK@append` 851
`\DSK@def` 850
`\dudot` 1702
`\DuplicateBibKeyWarning`
 1032, [1083](#)
- ## E
- `\edef` 59, 59
`\editiontext`
 6, 6, 7, [2632](#), [2638](#)
editor field 19
`\emph` 3074,
 3090, 3099,
 3101, 3142,

- 3144, 3161, 3163, 3178, 3179
- \empty@cite . 1385, 1389
- \EmptyNameWarning 1815, 1865
- \EmptyPrimaryWarning 2345, 2350
- \endbibchapter . . 1134
- \endbibdiv 1141
- \endbiblist 1140
- \endbibsection . . 1132
- \endgroup 19
- \endinput 42, 101
- \endlist 518
- environments:
 - bibchapter . . . 1113
 - bibdiv 1131
 - biblist 1, 23, 23, 484
 - biblist* 490
 - bibsection . . . 1122
 - tabbing 60
 - thebibliography 1136
- \eprint 328, 390, 433, 451, 2663, 2967, 3013, 3034, 3052
- \eprintpages . . 325, 345, 2564, 2964
- \erase@field 19, 19, 19, 19, 289, 2377, 2484, 3192, 3197
- \errmessage 262
- \etalchar 46, 58, 59, 1707, 2133, 2138, 2148
- \etaltext 58, 1018, 1035, 1708, 2078, 2137, 2233, 2489, 2490, 2491, 2828, 2929
- \ExecuteOptions . . 148
- export-bibtex package 26
- \extr@cite . . 1070, 1395, 1400, 1422, 1429, 1648, 1658, 1660
- \extract@initial 1993, 1998, 2015, 2030
- \extract@initial@a . . . 67, 2000, 2002
- \extract@initial@b 2007, 2014
- \extract@initial@c 2009, 2017
- \extract@initial@d 2024, 2026
- \extract@initials . . . 1819, 1866, 2745
- \extract@surnames . . . 2115, 2122, 2141
- F**
- \fld@elt 186, 191, 192, 193, 194, 195, 2384, 2542, 2546, 2553
- \font 23
- \fontseries 1403
- \fontshape 1403
- \foo 59
- \frenchspacing 79, 89, 113, 512
- \fsa@n 67
- \fullcite . . . 1436, 2892
- \fullocite . . . 1437, 2898
- \futurelet 63
- G**
- \G@refundefinedtrue 1392
- \g@undef . . 58, 662, 1177
- \generate@alphalabel . . . 2070, 2680, 2698
- \generate@label 723, 1004, 2680, 2698, 2818
- \get@current@properties . . . 275, 292, 2333
- \get@namepart 63, 63, 64, 67, 1806, 1809, 1812, 1823, 2116
- \get@namepart@a 64, 67, 1826, 1828, 1840
- \get@namepart@b 64, 67, 1830, 1832
- \get@namepart@c 68, 1834, 1839
- \get@namepart@d 63, 1836, 1842
- \get@namepart@e 1844, 1846
- \get@namepart@f 64, 64, 68, 1854, 1857
- \get@nth@property 294
- \get@numberof . . 72, 2080, 2123, 2219
- \get@property 817
- H**
- \H 1694
- har.bst 8
- \hbox 2363
- \hfil 526
- hij.bib 8
- \href 2808, 2811
- \hrulefill 2363
- \Hy@backout 1650
- \hyper@link 1658
- \hyper@anchorend 557, 1668, 2854
- \hyper@anchorstart 557, 1667, 2854
- hyperref package 54, 54, 56, 56
- \hyph@to@space 2104, 2117
- \hyphenpenalty 30
- I**
- \i 59, 61, 1718, 1773
- \if@tempwa . . . 606, 616, 1062, 2173
- \ifBR@verbose 1644, 1647
- \IfEmptyBibField 7, 7, 17, 185, 2297, 2451, 2467, 2468, 2555, 2565, 2584, 2587, 2590, 2729, 2947, 2948
- \IfFileExists 11, 154, 164, 1192, 1195, 1198
- \ifin@ 668, 873, 1040, 1342, 2038, 2392, 2403
- ifoption package . . . 14
- \ifsame@stems 2168, 2190
- \ifx 12
- \in@ 667, 872, 1039, 1341, 2037, 2391, 2402
- \ini@hyphen 1868, 1885, 1891, 1922, 1924, 2034, 2035
- initials option . . 63, 91
- \InnerCite . . 1319, 1324, 1349, 1362, 1454, 2877
- inputenc package . . . 60
- \interlinepenalty 509
- inverted attribute . . 63
- isbn field 7
- \issuetext 6, 6, 324, 344, 2556, 2963
- \itemindent 564
- J**
- \j 59, 61, 1719, 1774

- journal field 32
- K**
- \k 1695
- \keep@last@digit .
..... 2650, 2654
- \kern 726,
2364, 2427, 2468
- klm.bib 8
- L**
- \L 1725
- \l 1725
- label field 72, 93
- \labelsep 499, 542, 566
- \labelwidth ... 540,
541, 560, 561, 562
- \language name 984, 1000
- \lasp 1730
- \lastkern 268, 2269,
2277, 2289, 2427
- \lastpenalty 2270, 2272
- \lc@do 1755, 1760
- \lc@edef 1751, 2195, 2196
- \lccode 29
- \lcsymbol 58
- \leftmargin 541, 542,
561, 563, 564, 566
- \let 20, 20, 48, 49
- \lfhook 1703
- \linebreak 5
- \list 500
- lite option 16
- \lncan@a .. 2827, 2843
- logical-quotes op-
tion 6, 91
- \lowercase 29, 648,
795, 832, 980, 1756
- ltoutenc.dtx 60
- M**
- \m@ne 265, 994
- \m@th 1355
- \macrotext . 52, 55, 584
- \makeatletter 90
- \makelabel 505
- \MakeUppercase .. 1108
- \mathchar 30
- \mathcode 28, 33
- mathscinet package
15, 57, 58, 59, 59
- \mathsurround 46
- \mdash 8, 8, 2522
- \mddefault 1403
- \mdseries 1355
- \meaning 20, 52
- \MessageBreak
.. 106, 1290, 1299
- \MessageBreakNS ..
..... 106, 1086
- \modify@xref@fields
..... 624, 642
- \ModifyBibLabel ..
..... 2178, 2205
- \month@ 88
- \moo 18, 19, 19
- \MR 2532
- \MR@url 2801, 2808, 2811
- \MRhref 2536, 2540, 2805
- msc-links option ... 92
- \MultipleBibSelectWarning
..... 1181, 1183
- \MultipleCiteKeyWarning
..... 1279, 1287
- N**
- name entry type 3
- \name ... 72, 72, 72,
72, 93, 93, 93,
93, 186, 188,
189, 190, 782,
798, 802, 808,
825, 2077, 2080,
2121, 2131,
2132, 2136,
2139, 2314,
2827, 2925, 2936
- \name@split 63,
63, 63, 1805, 2328
- \name@split@finish
..... 1812, 1814
- \name@split@given
... 63, 1806, 1808
- \name@split@jr ...
..... 1809, 1811
- \NameCheck 1817
- \ndash 8, 8, 86,
2522, 2558, 2563
- \NeedsTeXFormat 2, 3064
- \newcount .. 2162, 2208
- \newcounter .. 483, 528
- \newdimen 520
- \newif 2190
- \newlabel 532
- \newread 1190
- \NG 1726
- \ng 1726
- \NoBibDBFile 1202, 1209
- \nobreak 67
- \nobreakspace 1869, 2609
- nobysame option ... 128
- \nocite 49, 1440
- non-compressed-cites
option 91
- non-sorted-cites op-
tion 15, 91
- \NonNumericCiteWarning
..... 1476, 1480
- \nopunct . 6, 6, 122,
1339, 1559, 2451
- \normal@bibdef ...
..... 26, 26,
588, 590, 603, 1179
- \NormalCatcodes ... 37
- \normalfont 497
- \normalize@edef ..
..... 1737,
1873, 2079, 2829
- \number
30, 2288, 2609,
2634, 2640, 2650
- \numeric@refs . 715,
2682, 2700, 2848
- O**
- \O 1722
- \o 1722
- \ObsoleteCiteOptionWarning
..... 1283, 1297
- \ocite 1431,
1434, 1437,
2867, 2878,
2879, 2880, 2881
- \ocitelist 2876
- \ocites . 46, 1435, 2870
- \OE 58, 1721
- \oe 58, 1721
- \on@line 14
- \OpenBBLFile 1173, 1249
- \openin 1193, 1196, 1199
- \or 2242, 2244,
2614, 2615,
2616, 2651,
2778, 2779, 2780
- organization field ... 3
- \OtherCite . 1454, 1469
- \othercitelist ...
..... 1445, 1448
- \othercites . 1440, 1441
- \output@inner@xref@
..... 631, 666
- \output@xref@
630, 650, 654, 669
- \output@xref@a ...
..... 625, 626,
627, 628, 629, 646
- \overshadebox ... 1656
- P**
- \PackageError 105
- \PackageInfo 1648
- \PackageWarning .. 104
- \PackageWarningNoLine
..... 4
- \parenthesize .. 5,
5, 5, 6, 6, 85,
329, 369, 391,

- 435, 453, [2496](#),
 2968, 2992,
 3014, 3036,
 3054, 3076, 3092
 \parse@MR .. 2800, 2807
 partial field 85
 pcatcode package ... 16
 place field 3, 3, 3
 \Plural ... 7, 7, 77,
 79, [2293](#), 2372,
 2375, 2380,
 2487, 3191, 3196
 \polhk 1705
 \pop@bracket [2497](#), [2500](#)
 \pop@bracket@a ...
 2508, 2511
 \PopCatCodes 2817, 3061
 \pp@scan ... 2558, 2561
 \pp@scan@a . 2558, 2563
 \prev@cite 52,
 52, 52, 52, 52,
 53, 1504, [1539](#),
 1547, 1574,
 1592, 1595,
 1596, 1603, 1604
 \prev@cite@cb
 ... 53, [1540](#), 1591
 \prev@citekey
 ... [754](#), 764, 2178
 \prev@names
 . 1359, 2914, 2916
 \previous@primary
 . 81, [895](#), 900,
 2354, 2371, 2478
 \previous@stem 2065,
 2073, 2167, 2821
 \previous@year . 75,
 2067, 2169, 2823
 \prime 6
 \print@backrefs ..
 [1652](#), 2704
 \print@citenames .
 2894,
 2900, 2920, [2934](#)
 \print@contribution
 .. 2385, 2386,
 2387, 2388, [2390](#)
 \print@date .. 2435,
 2598, 2599,
 2600, 2620, [2946](#)
 \print@edition ...
 2622, [2625](#)
 \print@month@day .
 2603, [2607](#)
 \print@one@dash ..
 [1558](#), 1590
 \print@partial ...
 .. 2493, 2547,
 2548, 2549, 2550
 \print@primary ...
 907, 910, 914, 2347
 \PrintAuthors .. 5,
 5, 6, 6, 6, 315,
 378, 444, 476,
 914, [2353](#), 2954,
 3001, 3045,
 3068, 3120, 3177
 \PrintBackRefs ...
 .. 761, [1655](#), 2704
 \PrintBook ... 384,
 2439, 3008, 3126
 \PrintCiteNames ..
 2844,
 2846, 2908, [2911](#)
 \PrintCNY .. 2845, [2907](#)
 \PrintConference .
 383,
 2413, 3007, 3125
 \PrintConferenceDetails
 .. 401, [2416](#), 3138
 \PrintConferenceDetails@
 . 2420, 2423, [2426](#)
 \PrintContributions
 319, 340,
 361, 382, 411,
 427, [2382](#), 2959,
 2985, 3006,
 3029, 3072,
 3089, 3106,
 3124, 3149, 3164
 \PrintDate ... 8, 8,
 8, 2588, 2593,
 2598, 2945,
 2955, 2978,
 3002, 3024, 3046
 \PrintDateB ... 367,
 386, 417, 432,
 450, 2599,
 3078, 3094,
 3111, 3129,
 3154, 3169, 3183
 \PrintDatePosted .
 2591, [2620](#)
 \PrintDatePV
 ... 323, 343, [2586](#)
 \PrintDOI 327, 389,
 2582, 2966, 3012
 \PrintEdition 6, 7, 7,
 7, 8, 358, 408,
 426, [2621](#), 2982,
 3028, 3110, 3153
 \PrintEditorsA . 6,
 6, 7, 7, 7, 80,
 81, 910, [2370](#), [3189](#)
 \PrintEditorsB ...
 .. 6, 7, 7, 359,
 409, [2379](#), 2983
 \PrintEditorsC ...
 7, 7, 7, [2367](#)
 \PrinteSeries@a . [2218](#)
 \PrintISBNs [2552](#)
 \PrintNameList 350,
 2366, 3104,
 3105, 3147, 3148
 \PrintNames 80,
 2310, 2358,
 2366, 2368,
 2375, 2380,
 2475, 2482,
 2487, 3191, 3196
 \PrintNames@a [2311](#), [2313](#)
 \PrintPartials 320,
 2544, 2960, 3073
 \PrintPrimary . 6, 6,
 354, 422, [2343](#),
 2465, 2977,
 3023, 3098, 3160
 \PrintReprint . 331,
 371, 393, 437,
 455, [2442](#), 2970,
 2994, 3016,
 3038, 3056,
 3081, 3115,
 3131, 3171, 3185
 \PrintReviews . 7, 7,
 334, 374, 396,
 440, 458, [2541](#),
 2973, 2997,
 3019, 3041, 3059
 \PrintSeries ... 76,
 76, 77, 77, 80,
 2209, 2314,
 2383, 2542,
 2545, 2553,
 2877, 2925, 2936
 \PrintSeries@a ...
 ... 77, 2216, 2218
 \PrintThesisType .
 447,
 2567, 3049, 3180
 \PrintTranslation
 330,
 370, 392, 436,
 454, [2456](#), 2969,
 2993, 3015,
 3037, 3055,
 3080, 3114,
 3130, 3170, 3184
 \PrintTranslatorsA
 7, 7,
 81, 907, [2477](#), [3194](#)
 \PrintTranslatorsB
 7, 7, [2486](#)
 \PrintTranslatorsC
 7, 7, 360,
 410, [2474](#), 2984

- \PrintYear
 . 8, 8, 8, 88, 2619
 proceedings entry
 type 7
 \process@bare@dot
 1956, 1966
 \process@citelist
 1350,
 1488, 2681,
 2699, 2757, 2856
 \process@citelist@sorted
 1472, 1488
 \process@citelist@unsorted
 1477,
 1485, 2681,
 2699, 2757, 2856
 \process@dots .. 66,
 66, 69, 1877, 1939
 \process@hyphens 66,
 68, 69, 1876, 1886
 \process@names ...
 1878, 1990
 \process@one@dot .
 69, 1942,
 1950, 1972, 1987
 \process@one@dot@a
 1952, 1954
 \process@one@dot@b
 1959, 1971
 \process@one@dot@c
 1961, 1974
 \process@one@dot@d
 . 1969, 1981, 1983
 \process@one@hyphen
 67, 1889,
 1896, 1918, 1933
 \process@one@hyphen@a
 69, 70, 1898, 1900
 \process@one@hyphen@b
 . 1906, 1909, 1917
 \process@one@hyphen@c
 1911, 1920
 \process@one@hyphen@d
 ... 67, 1927, 1929
 \process@xrefs 614, 640
 \ProcessExclusiveOptions
 152
 \ProcessOptions .. 150
 \prop ... 284, 286, 2336
 \prop@reset 297
 \protected@edef ..
 1019,
 1073, 1115, 1124
 \ProvidesFile ... 1213
 \ProvidesPackage 3, 3065
 publisher field 32
 \purge@accent 1771, 1802
 \purge@edef
 . 1767, 2193, 2194
 \purge@edef@
 .. 1777, 1782,
 1792, 1793, 1797
 \purge@edef@a
 ... 61, 1784, 1786
 \push@bracket 2497, 2500
 \PushCatcodes 37
- R**
- \r 1696
 \rasp 1731
 \read 1224
 \ReadBibData 1174, 1191
 \ReadBibData@a ...
 . 1203, 1207, 1212
 \ReadBibLoop
 . 1216, 1220, 1234
 \ReadBibLoop@a ...
 1225, 1229
 \ReadBibLoop@b ...
 1236, 1240
 \ReadBibLoop@e .. 1229
 \refname 39, 1102, 1122
 \RequirePackage 4,
 12, 36, 124, 155,
 165, 178, 181,
 182, 2796, 3066
 \reset@nth@property 820
 \resetbiblist
 ... 538, 559, 1138
 \ResetCapSFCodes .
 513, 545
 \resolve@inner 2401,
 2414, 2440,
 2443, 2457, 2494
 \restore@labelwidth
 501, 536
 \RestrictedSetKeys
 11, 19,
 296, 596, 618,
 642, 2394, 2404
 review field 5, 7
 rkeyval package ...
 5, 10, 19, 67
 \rkv@DSAK 19
 \rkv@setter .. 868, 888
 \rkvIfAdditive ... 274
 \rkvIfEmpty ... 16, 185
 \romannumeral 106
 \rsk@resume 684
 \rsk@set 27, 618
 \rsk@set@a 19
 \rsk@set@b 19
 \rsk@toks
 10, 11, 26, 30,
 30, 297, 597,
 642, 2395, 2404
- S**
- \safe@set 81,
 1508, 1515,
 1531, 1565,
 1572, 1588, 2151
 \same@stemfalse 2200
 \same@stemstrue . 2198
 \sameauthors .. 6, 7,
 7, 7, 7, 7, 2355,
 2361, 2372, 2479
 \save@labelwidth .
 516, 529
 \save@primary 763, 896
 \section* 39
 \selbibdef@a . 687, 689
 \selbibdef@b . 691, 694
 \select@auxlanguage
 ... 276, 283, 2334
 \selective@bibdef
 26,
 50, 686, 1163, 1461
 \selectlanguage ..
 ... 35, 36, 286, 986
 \SentenceSpace 334,
 374, 396, 440,
 458, 1653, 2662,
 2973, 2997,
 3019, 3041, 3059
 \series@add . 2227, 2231
 \series@add@... .. 77
 \series@add@a 2211, 2243
 \series@add@b 2212, 2246
 \series@add@c
 . 2213, 2248, 2255
 \series@add@d 2214, 2252
 \series@add@e 2215, 2237
 \series@index
 20, 265,
 294, 781, 793,
 820, 2208, 2226,
 2232, 2234,
 2239, 2241,
 2251, 2254, 2393
 \series@total
 .. 2220, 2221,
 2245, 2251, 2254
 \set 625,
 626, 627, 628,
 629, 630, 631,
 632, 633, 634, 635
 \set@firstname ...
 . 2311, 2321, 2904
 \set@name ... 2322,
 2325, 2327, 2905
 \set@name@a
 ... 81, 2329, 2331
 \set@namea 2331

- `\set@othername` ...
 ... 2311, 2316,
 2317, 2318, 2324
`\setbib@` 26, 26, 26,
 27, 28, 29, 30, 30
`\setbib@@` 718,
 727, 732, 735,
 738, 741, 940, 2471
`\setbib@article` 727, 940
`\setbib@book` 2453
`\setbib@conference` 2414
`\setbib@contribution`
 2395
`\setbib@incollection`
 469, 732, 735, 738
`\setbib@innerbook` 2440
`\setbib@inproceedings`
 472
`\setbib@misc` 594
`\setbib@nameBE` ...
 2300, 2337
`\setbib@nameinverted`
 2305, 2905
`\setbib@nameLE` ...
 . 2295, 2322, 2325
`\setbib@partial` . 2494
`\setbib@TYPE` 18
`\setcounter` 485
`\settowidth` .. 522, 560
`\sfcode` 79,
 114, 115, 116,
 117, 118, 119,
 548, 2279, 2437
`\shade@cite` .. 1656,
 1658, 1663, 1666
`shaderef` package ... 56
`short-journals` op-
 tion 15, 91
`short-months` option . 92
`short-publishers` op-
 tion .. 15, 92, 92
`shortalphabetic` op-
 tion 90
`showkeys` package ..
 44, 56, 56
`\SingularPlural` ..
 .. 7, 7, 77, 79,
 2222, 2224, 2293
`\sixt@on` ... 1254, 1260
`\SK@` .. 1011, 1660, 1677
`\SK@label` . 1011, 1676
`\SK@ref` 1660
`\soft` 1706
`sorted` option 14
`\space` 67
`\spacefactor` .. 110,
 122, 2279, 2281,
 2437, 2533, 2662
`\SS` 1727
`\ss` 1727
`\star@` 676,
 1304, 1362,
 2858, 2868,
 2887, 2895, 2901
`\star@bibdef` ... 26,
 573, 601, 658, 1215
`\stepcounter` 496
`\string` 60
`\strip@MRprefix` ..
 2524, 2535
`\strip@prefix` .. 20, 52
`\SubEtal` 2235, 2237, 2491
`\surround@names` ..
 2106, 2108
`\SwapBreak` 78,
 78, 270, 2212,
 2213, 2214,
 2215, 2264,
 2380, 2434, 2487
`\symbol` 58, 58

T

`\t` 1697
`tabbing` environment 60
`\textbf` 5, 322, 342, 2962
`textcmds` package ..
 8, 15, 85
`\textemdash` ... 8, 2522
`\textendash` 8, 1559, 2523
`\textfont` 23
`\textit` 78,
 316, 318, 339,
 355, 357, 379,
 381, 423, 425,
 445, 446, 2956,
 2958, 2979,
 2981, 3003,
 3005, 3025,
 3027, 3047, 3048
`\textquotedblleft` 2516
`\textquotedblright` 2519
`\textsc` 76
`\textsubscript` 6
`\textsuperscript` ... 6
`\texttt` 20
`\tf@brf` 2710, 2711
`\TH` 1728
`\th` 1728
`\the` 26, 30, 30
`\thebib` 557,
 560, 1009, 2675,
 2677, 2688, 2690
`thebibliography` envi-
 ronment ... 1136
`\thebibliography` .. 90
`\thesis@type` 2568, 2571
`\thinspace` 2363
`\thr@@` 2245, 2643
`\toks@` 18, 64
`\tprime` 6, 6, 6, 6
`\TrailingHyphenWarning`
 1905, 1936
`transition` field ...
 ... 5, 5, 5, 17, 31
`\tsub` 6, 6
`\tsup` 6, 6
`\ttfamily` 1403
`\tw@` 2176, 2234
`\typeout` 172, 1169, 1171

U

`\u` 1698
`\uarc` 1701
`\udot` 1704
`\UndefinedCiteWarning`
 1393, 1398
`\unkern` 2269
`\unpenalty` 2273
`\unskip` .. 67, 1885,
 2268, 2533, 2620
`\uparrow` 1653
`\upn` 85, 85,
 2497, 2499, 2512
`upref` package 85
`\upshape` 1355
`\url` 18, 2663
`\use@accent`
 . 1741, 1747, 1753
`\utilde` 1700

V

`\v` 1699
`\vdef` ... 53, 1309, 1374
`\voltext` ... 6, 6, 6,
 363, 413, 2555,
 2987, 3103, 3146
`\vphantom` 2808

W

`\widowpenalty` 511
`\wrap@accent`
 ... 61, 1740, 1748
`wrapper` macros, defi-
 nition 12

X

`xref` field 32, 33
`\xref@add@toks` 850, 865
`\xref@append` . 851, 886
`\xref@check@a`
 769, 770, 771, 775
`\xref@check@aa` 782, 792
`\xref@check@ab` 804, 811
`\xref@check@b`
 ... 772, 773, 828
`\xref@check@c`
 . 32, 768, 839, 844

<code>\xref@ID</code>	<code>xyz.tex</code> 8	<code>\ycites</code> 1433 , 2860
854, 855, 856, 858		<code>\year@short</code>
<code>\XRefWarning</code>	Y	. 2149 , 2156, 2786
. . . 856, 893, 2407	y2k option . . . 71, 92, 127	
<code>\XrefWarning</code> 893	<code>\ycite</code> . . . 49, 1432 ,	Z
<code>xyz.aux</code> 8	1434, 1438,	<code>\Z</code> 550
<code>xyz.bbl</code> 9, 9	1439, 2857 , 2890	<code>\zap@space</code> . 1311, 1376