

The `amsrefs` package

Michael Downes and David M. Jones
American Mathematical Society

Version 2.10, 2012/05/08

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	35
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	40
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	48
6.15.4	The <code>\nocite</code> command	49
6.15.5	Citation sorting	50

6.15.6	Range compression	51
6.15.7	Munging the <code>.aux</code> file	54
6.15.8	Back references	55
6.15.9	<code>hyperref</code> , <code>showkeys</code> and <code>shaderef</code> support	56
6.16	Lexical structure of names	56
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	65
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	75
6.21	Formatting series	75
6.22	Formatting names and series of names	79
6.23	The <code>partial</code> field	84
6.24	Special formatting for other fields	84
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	89
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	90
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	91
6.26.11	The <code>short-months</code> option	91
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	92
6.27	The <code>amsbst</code> package	98

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 re-instated `address` as the primary field and `place` is now an undocumented alias.

The following fields are reserved for future use:

doi Digital Object Identifier

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

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

url Universal Resource Locator.

3.2 Bibliography entry types

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

`collection.article`
`proceedings.article`
`partial`
`conference`
`innerbook`
`name`
`nameLE`
`nameBE`
`nameinverted`
`publisher`

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

`miscellaneous`
`periodical`

4 Customizing the bibliography style

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

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

\RequirePackage{amsrefs}\relax

\BibSpec{article}{
  ...
}

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

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

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

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

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

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

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

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

This means that if two names are given, as in

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

then the final value of the `author` field seen when \LaTeX processes the style specification will be

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

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

5 Miscellaneous commands provided by the `amsrefs` package

Most of the following commands are helper commands for use in `\BibSpec` statements. The others are intended for use in bibliography data.

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

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

`\voltext` This is used to format volume numbers. By default, it precedes the volume number by “vol.”

`\issuetext` This is used to format issue numbers. By default, it precedes the volume number by “no.”

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

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

`\tsup`, `\tsub`, `\tprime` These are for text subscripts and superscripts, with `\tprime` producing a superscript prime symbol. Unlike the standard `\textsuperscript` and `\textsubscript` functions provided by 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 $\text{Bib}\text{\TeX}$.

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

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

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

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

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

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

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

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

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

6.2 How cites are processed by `amsrefs`

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

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

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

We therefore have

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

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

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

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

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

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

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

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

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

6.3 Data structures

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

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

The scanned result is to assign

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

```

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

```

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

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

6.4 Preliminaries

```
1 \*pkg
```

Standard declaration of package name and date.

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

```
3 \ProvidesPackage{amsrefs}[2012/05/08 v2.10]
```

```
\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 (#2) 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}}% 65 \@xp\endgroup 66 \@tempa 67 }</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.

`\@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
`\name` processing is done. (It's not clear to me there's any real benefit to using these instead of just using `\do.—dmj`)


```

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

```

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

```

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

196 \DefineSimpleKey{bib}{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 `\@emptys` 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     \xp\get@nth@property\@xp\@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 \@xp\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   \xp\def\@xp\@tempa\@xp{\csname setbib@#1\@xp\endcsname}%
306   \xp\ifx\csname setbib@#2\endcsname\@tempa
307     \amsrefs@error{%
308       Mirror alias #1->#2 not allowed (infinite loop)}\@ehc
309   \else
310     \xp\def\csname setbib@#1\@xp\endcsname
311       \@xp{\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 }

\biblistfont

495 \newcommand{\biblistfont}{%

```

```

496     \normalfont
497     \footnotesize
498 }

```

\@biblist

```

499 \newcommand\@biblist[1] [] {%
500     \stepcounter{bib@env}
501     \biblistfont
502     \labelsep .5em\relax
503     \list{\BibLabel}{%
504         \restore@labelwidth
505         \@maxlabelwidth\z@
506         \@nmbrlisttrue
507         \def\@listctr{bib}%
508         \let\makelabel\bib@mklab
509         #1\relax
510     }%
511     \sloppy

```

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

```

512     \interlinepenalty\@m
513     \clubpenalty\@M
514     \widowpenalty\clubpenalty
515     \frenchspacing
516     \ResetCapSFCodes
517 }

```

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

```

518 \def\@endbiblist{%
519     \save@labelwidth
520     \def\@noitemerr{\@latex@warning{Empty bibliography list}}%
521     \global\let\previous@primary\@empty
522     \endlist
523 }

```

\@maxlabelwidth

```

524 \newdimen\@maxlabelwidth

```

\bib@mklab

```

525 \def\bib@mklab#1{%
526     \settowidth\@tempdima{#1}%
527     \ifdim \@tempdima > \@maxlabelwidth
528         \global\@maxlabelwidth\@tempdima
529     \fi
530     #1\hfil
531 }

```

```

532 \newcounter{bib@env}

```

\save@labelwidth

```

533 \def\save@labelwidth{%
534     \if@filesw
535         \immediate\write\@auxout{%
536             \string\newlabel{[bibenv:\the\c@bib@env]}{\the\@maxlabelwidth}%
537         }%
538     \fi
539 }

```


`\restore@labelwidth`

```

540 \def\restore@labelwidth{%
541   \@xp\ifx \csname r@[bibenv:\the\c@bib@env]\endcsname \relax
542     \resetbiblist{00}%
543   \else
544     \@xp\labelwidth\csname r@[bibenv:\the\c@bib@env]\endcsname
545     \leftmargin\labelwidth
546     \advance\leftmargin\labelsep
547   \fi
548 }
```

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

```

549 \providecommand{\ResetCapSFCodes}{%
550   \count@=\A
551   \def\@tempa{%
552     \sfcode\count@=\@m
553     \advance\count@\@ne
554     \ifnum\count@>\Z\relax \expandafter\@gobble \fi
555   \@tempa
556 }%
557 \@tempa
558 }
```

`\CurrentBib` In case this is undefined sometimes.

```
559 \def\CurrentBib{??}
```

`\BibLabel`

```

560 \newcommand{\BibLabel}{%
561   \hfill
562   \Hy@raisedlink{\hyper@anchorstart{cite.\CurrentBib}\hyper@anchorend}%
563   [\thebib]%
564 }
```

`\resetbiblist`

```

565 \newcommand{\resetbiblist}[1]{%
566   \settowidth\labelwidth{\def\thebib{#1}\BibLabel}%
567   \leftmargin\labelwidth
568   \ifdim\labelwidth=\z@
569     \leftmargin=1em
570     \itemindent=-\leftmargin
571   \else
572     \advance\leftmargin\labelsep
573   \fi
574 }
```

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.

```

575 \newcommand{\bib}{%
576   \begingroup
```

```

577      \@ifstar{%
578          \@tempwattrue
579          \let\@bibdef\star@bibdef
580          \BibItem
581      }{%
582          \@tempwafalse
583          \BibItem
584      }%
585 }

```

`\BibItem` *Arguments:*

```

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

```

```

586 \newcommand{\BibItem}[2]{%
587     \def\@tempa{#1}%
588     \edef\@tempb{%
589         \@nx\@bibdef\@xp\@nx\csname setbib@#2\endcsname{#2}%
590         {\macrotext\@tempa}%
591     }%
592     \@tempb
593 }

```

`\@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.

```

```

594 \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.

```

```

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

```

6.12.1 `\@bibdef` Implementations

`\normal@bibdef` *Arguments:*

```

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

```

```

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

```

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

```

597 \def\CurrentBibType{#2}%
598 \ifx\relax#1%
599 \amsrefs@error{Undefined entry type: #2}\@ehc
600 \let#1\setbib@misc
601 \fi
602 \RestrictedSetKeys{bib}%
603 {\bib@exec{#3}{\the\rsk@toks}{#1}\endgroup}%
604 }
605
606 \let\@bibdef\normal@bibdef

```

`\star@bibdef` *Arguments:*

```

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

```

```

607 \def\star@bibdef{%
608 \let\bib@exec\bib@store
609 \normal@bibdef
610 }

```

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

```

611 \def\copy@bibdef{%
612 \if@tempswa
613 \@xp\defer@bibdef
614 \else
615 \@xp\copy@bibdef@a
616 \fi
617 }

```

`\copy@bibdef@a`

```

618 \def\copy@bibdef@a#1#2#3#4{%
619 \open@bbl@file
620 \process@xrefs{#4}%
621 \bbl@write{%
622 \string\bib\if@tempswa\fi{#3}{#2}\string{\iffalse}\fi
623 }%

```

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.

```

624 \RestrictedSetKeys{\global\let\rsk@set\bbl@copy}\@empty
625 {\bbl@write{\iffalse\fi\string{^^J}}%
626 \endgroup}{#4}%
627 }

```

```

628 \catcode`\:=11
629

```

```

630 \def\modify@xref@fields{%
631 \let\set:bib'author\output@xref@a
632 \let\set:bib'editor\output@xref@a
633 \let\set:bib'translator\output@xref@a
634 \let\set:bib'journal\output@xref@a
635 \let\set:bib'publisher\output@xref@a
636 \def\set:bib'xref##1##2{\output@xref@{##1}\@empty}%
637 \def\set:bib'book##1##2{\output@inner@xref@{##1}\@empty}%
638 \let\set:bib'conference\set:bib'book

```

```

639 \let\set:bib'partial\set:bib'book
640 \let\set:bib'reprint\set:bib'book
641 \let\set:bib'translation\set:bib'book
642 }
643
644 \catcode`\:=12
645
646 \def\process@xrefs#1{%
647 \begingroup
648 \RestrictedSetKeys{\modify@xref@fields}{bib}{\the\rsk@toks}{#1}%
649 \endgroup
650 }
651
652 \def\output@xref@a#1#2{%
653 \def\@tempa{#1}%
654 \lowercase{\def\@tempb{#1}}%
655 \ifx\@tempa\@tempb
656 \output@xref@{#1}%
657 \fi
658 }
659
660 \def\output@xref@#1{%
661 \@ifnotempty{#1}{%
662 \ifundefined{bi@#1}{%
663 \begingroup
664 \let\star@bibdef\copy@bibdef@a
665 \csname bi@#1\endcsname
666 \endgroup
667 }%
668 \@xp\g@undef\csname bi@#1\endcsname
669 }%
670 }
671
672 \def\output@inner@xref@#1{%
673 \in@={#1}%
674 \ifin@
675 \output@xref@{#1}%
676 \fi
677 }

```

\bbl@copy

```

678 \def\bbl@copy#1\endcsname#2{%
679 \begingroup
680 \def\@tempa{#1}%
681 \toks@{{#2}}%
682 \star@{\bbl@copy@a}{#1}%
683 }

```

\bbl@copy@a

```

684 \def\bbl@copy@a#1{%
685 \@ifnotempty{#1}{%
686 \add@toks@{*}{#1}%
687 }%
688 \bbl@write{ \space\@tempa=\the\toks@,}%
689 \endgroup
690 \rsk@resume
691 }

```

\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.
```

```
692 \def\selective@bibdef#1#2#3{%
693   \exp\selbibdef@a\csname b@#3\endcsname{#1}{#2}{#3}%
694 }
```

\selbibdef@a

```
695 \def\selbibdef@a#1{%
696   \def\@tempa{\endgroup\@gobblefour}%
697   \ifx\relax#1\else \exp\selbibdef@b#1\@nil \fi
698   \@tempa
699 }
```

\selbibdef@b

```
700 \def\selbibdef@b#1#2#3\@nil{%
701   \ifx 1#2\let\@tempa\copy@bibdef\fi
702 }
```

\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.
```

```
703 \def\defer@bibdef#1#2#3#4{%
704   \exp\gdef\csname bi@#3\endcsname{%
705     \bib*{#3}{#2}{#4}%
706   }%
707   \exp\addto@defer@list \csname bi@#3\endcsname
708   \endgroup
709 }
```

\bibdefer@list

```
710 \let\bibdefer@list\@empty
```

\addto@defer@list

```
711 \def\addto@defer@list#1{%
712   \begingroup
713   \def\do{\@nx\do\@nx}%
714   \xdef\bibdefer@list{\bibdefer@list\do#1}%
715   \endgroup
716 }
```

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.

```
717 \def\bib@store#1{%
718   \afterassignment\@gobble
719   \exp\xdef\csname bi@#1\endcsname
720 }
```

\numeric@refs

```
721 \def\numeric@refs{00}
```

```

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

722 \def\bib@print#1#2#3{%
723     \bib@start{#1}%
724     \let\setbib@@#3%
725     #2\relax      % execute definitions locally
726     \bib@resolve@xrefs
727     \bib@field@patches
728     \bib@selectlanguage
729     \generate@label
730     \bib'setup
731     \bib@cite{#1}%
732     \kern\@ne sp
733     \ifx\setbib@@\setbib@article
734         \ifx\bib'booktitle\@empty
735             \ifx\bib'book\@empty
736                 \ifx\bib'conference\@empty
737                     \else
738                         \let\setbib@@\setbib@incollection
739                     \fi
740                 \else
741                     \let\setbib@@\setbib@incollection
742                 \fi
743             \else
744                 \let\setbib@@\setbib@incollection
745             \fi
746         \fi
747     \setbib@@
748     \bib@end
749 }

```

`\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.

750 \def\bib@print@inner#1#2{%
751     \begingroup
752     #2\relax      % execute definitions locally
753     \bib@resolve@xrefs
754     \bib@field@patches
755     \bib'setup
756     #1%
757     \endgroup
758 }

```

```

\current@citekey

759 \let\current@citekey\@empty

```

```

\prev@citekey

760 \let\prev@citekey\@empty

```

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

```

761 \def\bib@start#1{%
762     \begingroup

```

```

763      \def\current@citekey{#1}%
764 }

```

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

```

765 \def\bib@end{%
766     \relax
767     \@xp\PrintBackRefs\@xp{\CurrentBib}%
768     \par
769     \save@primary
770     \global\let\prev@citekey\current@citekey
771     \endgroup
772 }

```

6.12.3 Resolving cross-references

`\bib@resolve@xrefs`

```

773 \def\bib@resolve@xrefs{%
774     \xref@check@c\bib'xref
775     \xref@check@a\bib'author
776     \xref@check@a\bib'editor
777     \xref@check@a\bib'translator
778     \xref@check@b\bib'journal
779     \xref@check@b\bib'publisher
780 }

```

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

```

781 \def\xref@check@a#1{%
782     \ifx\@empty#1\relax
783     \else
784         \begingroup
785             \toks@{\@emptytoks
786                 \@temptokenb\@emptytoks
787                 \series@index\z@
788                 \def\name{\xref@check@aa#1}%
789                 #1\relax
790                 \edef\@tempa{%
791                     \def\@nx#1{\the\toks@}%
792                     \the\@temptokenb
793                 }%
794             \@xp\endgroup
795             \@tempa
796     \fi
797 }

```

`\xref@check@aa`

```

798 \def\xref@check@aa#1#2{%
799     \advance\series@index\@ne
800     \def\@tempa{#2}%
801     \lowercase{\def\@tempb{#2}}%
802     \ifx\@tempa\@tempb
803         \ifx\@tempa\@empty
804             \add@toks@{\name{}}%
805         \else
806             \@ifundefined{bi@#2}{%
807                 \BibAbbrevWarning{#2}%
808                 \add@toks@{\name{#2}}%
809             }{%

```

```

810          \xref@check@ab#1{#2}%
811      }%
812  \fi
813  \else
814      \add@toks@{\name{#2}}%
815  \fi
816 }

```

\xref@check@ab

```

817 \def\xref@check@ab#1#2{%
818     \csname bi@#2\endcsname
819     \ifx\@empty\bib'name
820         \@temptokena{#2}%
821     \else
822         \@temptokena\exp{\bib'name}%
823         \get@property\@tempa\bib'name
824         \edef\@tempa{%
825             \@nx\addto@hook\@temptokenb{%
826                 \@nx\reset@nth@property\@nx#1\the\series@index{\@tempa}%
827             }%
828         }%
829         \@tempa
830     \fi
831     \edef\@tempa{\@nx\add@toks@{\@nx\name{\the\@temptokena}}}%
832     \@tempa
833 }

```

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

```

834 \def\xref@check@b#1{%
835     \ifx\@empty#1%
836     \else
837         \toks@{\exp{#1}%
838         \edef\@tempb{\lowercase{\def\@nx\@tempa{\the\toks@}}}%
839         \@tempb
840         \ifx\@tempa#1\relax % all lowercase
841             \@ifundefined{bi@#1}{%
842                 \BibAbbrevWarning{#1}%
843             }{%

```

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.

```

844             \let#1\@empty
845             \xref@check@c\@tempa
846         }%
847     \fi
848 \fi
849 }

```

\xref@check@c Resolve an xref field.

```

850 \def\xref@check@c#1{%
851     \ifx#1\@empty
852     \else
853         \begingroup
854             \@apply\auto@protect\amsrefs@textsymbols
855             \@apply\auto@protect\amsrefs@textaccents
856             \let\DSK@def\xref@add@toks
857             \let\DSK@append\xref@append

```



```

858      \toks@\@emptytoks
859      \let\bib@reset\@empty
The \@for here is just a fancy way of expanding #1. (Or is it?)
860      \@for\xref@ID:=#1\do{%
861          \ifundefined{bi@\xref@ID}{%
862              \XrefWarning{\xref@ID}%
863          }{%
864              \csname bi@\xref@ID\endcsname
865          }%
866      }%
867      \edef\@tempa{\endgroup\the\toks@}%
868      \@tempa
869      \fi
870 }

```

`\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.

```

```

871 \def\xref@add@toks#1#2#3{%
872     \ifx#1\@empty
873         \edef\@tempa{%
874             \@nx\add@toks@{\@xp\@nx\csname\rkv@setter#1\endcsname{#2}{#3}}%
875         }%
876         \@tempa
877     \else
878         \in@\bib'title{#1}%
879         \ifin@
880             \ifx\bib'booktitle\@empty
881                 \edef\@tempa{%
882                     \@nx\add@toks@{%
883                         \@xp\@nx\csname set:bib'booktitle\endcsname
884                     }%
885                 }%
886                 \@tempa
887             \add@toks@{{#2}{#3}}%
888         \fi
889     \fi
890 \fi
891 }

892 \def\xref@append#1#2#3#4{%
893     \edef\@tempa{%
894         \@nx\add@toks@{\@xp\@nx\csname\rkv@setter#2\endcsname{#3}{#4}}%
895     }%
896     \@tempa
897 }

```

`\BibAbbrevWarning`

```

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

```

`\XrefWarning`

```

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

```

6.12.4 Bib field preprocessing

```

\current@primary
900 \let\current@primary\@empty

\previous@primary
901 \let\previous@primary\@empty

\save@primary
902 \IfOption{nobysame}{%
903   \let\save@primary\@empty
904 }{%
905   \def\save@primary{%
906     \global\let\previous@primary\current@primary
907   }%
908 }

\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.

```

909 \def\bib@field@patches{%
910   \ifx\bib'author\@empty
911     \ifx\bib'editor\@empty
912       \let\current@primary\bib'translator
913       \let\print@primary\PrintTranslatorsA
914     \else
915       \let\current@primary\bib'editor
916       \let\print@primary\PrintEditorsA
917     \fi
918   \else
919     \let\current@primary\bib'author
920     \let\print@primary\PrintAuthors
921   \fi
922   \ifx\bib'address\@empty
923     \let\bib'address\bib'place
924   \fi
925   \ifx\bib'organization\@empty
926     \ifx\bib'institution\@empty
927       \let\bib'organization\bib'school
928     \else
929       \let\bib'organization\bib'institution
930     \fi
931   \fi
932   \ifx\bib'date\@empty
933     \ifx\bib'year\@empty
934       \let\bib@year\bib'status
935     \else
936       \bib@parsedate\bib'year
937     \fi
938   \else
939     \bib@parsedate\bib'date
940   \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.

```

941   \def\@tempa{year}%
942   \ifx\bib'volume\@tempa
943     \let\bib'volume\bib@year

```

```

944     \let\bib'date\@empty
945     \fi

```

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

```

946     \ifx\setbib@@\setbib@article
947         \ifx\bib'volume\@empty
948             \ifx\bib'number\@empty\else
949                 \let\bib'volume\bib'number
950                 \let\bib'number\@empty
951             \fi
952         \fi
953     \fi

```

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

```

954     \bib@language@fixup
955 }

```

6.12.5 Date setup

`\bib@year`

```

956 \let\bib@year\@empty

```

`\bib@month`

```

957 \let\bib@month\@empty

```

`\bib@day`

```

958 \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.

```

959 \def\bib@parsedate#1{%
960     \@xp\bib@parsedate@a#1---\@nil
961 }

```

`\bib@parsedate@a`

```

962 \def\bib@parsedate@a#1-#2-#3-#4\@nil{%
963     \def\bib@year{#1}%
964     \def\bib@month{#2}%
965     \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.

```

966     \ifx\@empty\bib@day
967         \ifx\@empty\bib@month
968             \let\bib'date\bib@year
969         \else
970             \def\bib'date{#1-#2}%
971         \fi
972     \else
973         \def\bib'date{#1-#2-#3}%
974     \fi
975 }

```

6.12.6 Language setup

`\bib@language@fixup`

```

976 \def\bib@language@fixup{%
977     \ifx\bib'hyphenation\@empty
978         \ifx\bib'language\@empty

```

```

979         \let\bib@language\biblanguagedefault
980     \else
981         \let\bib@language\bib'language
982     \fi
983 \else
984     \let\bib@language\bib'hyphenation
985 \fi
986 \def\@tempa##1 ##2\@nil{\lowercase{\def\bib@language{##1}}}%

```

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

```

987     \@firstofone{\@xp\@tempa\bib@language} \@nil
988 }

```

`\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.

```

989 \def\bib@selectlanguage{%
990     \@ifsame@patterns{\language}{\bib@language}{}%
991     \@ifundefined{date\bib@language}{}%
992     \@xp\selectlanguage\@xp{\bib@language}%
993     }%
994 }%
995 }

```

`\@ifsame@patterns`

```

996 \def\@ifsame@patterns#1#2{%
997     \@xp\@ifsamepat\csname l@#1\@xp\endcsname\csname l@#2\endcsname
998 }

```

`\@ifsamepat`

```

999 \def\@ifsamepat#1#2{%
1000     \ifnum \ifx\relax#1\m@ne\else#1\fi = \ifx\relax#2\m@ne\else#2\fi
1001     \@xp\@firstoftwo
1002     \else
1003     \@xp\@secondoftwo
1004     \fi
1005 }

```

`\language`

```

\biblanguageEnglish 1006 \providecommand{\language}{english}
\biblanguagedefault 1007 \def\biblanguageEnglish{english}
\bib@language       1008 \let\biblanguagedefault\biblanguageEnglish
                   1009 \let\bib@language\@empty

```

6.12.7 Citation label setup

`\generate@label`

```

1010 \let\generate@label\relax

```

`\cite@label`

```

1011 \def\cite@label{\@currentlabel}

```

`\alpha@label`

```

1012 \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.`

```

1013 \def\bib@cite#1{%
1014     \def\CurrentBib{#1}%
1015     \alpha@label           % modify \thebib if necessary
1016     \item\leavevmode
1017     \SK@\SK@@label{#1}%
1018     \@xp\bib@cite@a\csname b@#1\endcsname
1019     \bibcite@write{#1}%
1020 }

1021 \def\bib@cite@a#1{%
1022     \ifx\relax#1%
1023         \begingroup
1024             \auto@protect\etaltext
1025             \protected@edef\@tempa{%
1026                 \gdef\@nx#1{%
1027                     \@nx\citesel 01{\cite@label}{\bib@label@year}{}%
1028                 }%
1029             }%
1030         \@xp\endgroup
1031         \@tempa
1032     \else
1033         \@xp\bib@cite@check\@xp#1#1\@empty\@empty\@empty\@empty\@empty
1034     \fi
1035 }
```

`\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.`

```

1036 \def\bib@cite@check#1#2#3#4#5#6#7{%
1037     \ifx 1#4\relax
1038         \DuplicateBibKeyWarning
1039     \else
```

This has gotten *way* out of hand.

```

1040     \begingroup
1041         \auto@protect\etaltext
1042         \@apply\auto@protect\amsrefs@textsymbols
1043         \@apply\auto@protect\amsrefs@textaccents
1044         \@tempswafalse
1045         \in@CitePrintUndefined{#5}%
```

```

1046         \ifin@
1047         \let\@tempa\@empty
1048     \else
1049         \def\@tempa{#5}%
1050     \fi
1051     \ifx\@tempa\@empty
1052     \else
1053         \@xp\ifx\@xp\@currentlabel\cite@label
1054         \edef\@tempb{\cite@label}%
1055     \else
1056         \let\@tempb\cite@label
1057     \fi
1058     \ifx\@tempa\@tempb
1059         \def\@tempa{#6}%
1060         \ifx\@tempa\bib@label@year
1061         \else
1062             \@tempswatrue
1063         \fi
1064     \else
1065         \@tempswatrue
1066     \fi
1067 \fi
1068 \if@tempswa
1069     \@ifempty{#6}{%
1070         \def\@tempa{#5}%
1071         \let\@tempb\cite@label
1072     }{%
1073         \def\@tempa{#5, #6}%
1074         \def\@tempb{\cite@label, \bib@label@year}%
1075     }%
1076     \amsrefs@warning{Citation label for \extr@cite#1 is
1077         changing from \@tempa ' to \@tempb '%}
1078 \fi
1079 \protected@edef\@tempa{%
1080     \gdef\@nx#1{%
1081         \@nx\citesel #31{\cite@label}{\bib@label@year}{#7}%
1082     }%
1083 }%
1084 \@xp\endgroup
1085 \@tempa
1086 \fi
1087 }

```

\bib@label@year

```
1088 \let\bib@label@year\@empty
```

\DuplicateBibKeyWarning

```

1089 \def\DuplicateBibKeyWarning{%
1090     \amsrefs@warning{%
1091         Duplicate \protect\bib\space key
1092         `\'CurrentBib ' detected\MessageBreakNS}%
1093 }

```

\bibcite@write

```

1094 \def\bibcite@write#1{%
1095     \if@filesw
1096     \begingroup
1097         \let\citesel\citesel@write
1098         \csname b@#1\endcsname

```

```

1099     \endgroup
1100   \fi
1101 }

```

`\citesel@write`

```

1102 \def\citesel@write#1#2#3#4#5{%
1103   \toks@{{#3}{#4}}%
1104   \immediate\write\@auxout{\string\bibcite{\CurrentBib}{\the\toks@}}%
1105 }

```

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

```
1106 \AtEndDocument{\let\bibcite\@gobbletwo}
```

6.12.8 Printing the bibliography

`\bibname`

```
1107 \providecommand{\bibname}{Bibliography}
```

`\refname`

```
1108 \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.

```
1109 \let\bib@div@mark\@gobble
```

This is verbose, but probably safer than any alternative.

```

1110 \@ifclassloaded{amsbook}{-}{%
1111   \@ifclassloaded{amsart}{-}{%
1112     \@ifclassloaded{amsproc}{-}{%
1113       \def\bib@div@mark#1{%
1114         \@mkboth{\MakeUppercase{#1}}{\MakeUppercase{#1}}%
1115       }%
1116     }%
1117   }%
1118 }

```

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

```

1119 \newenvironment{bibchapter}[1][\bibname]{%
1120   \begin{group}
1121     \protected@edef\@{%
1122       \endgroup
1123       \protect\chapter*{#1}%
1124       \protect\bib@div@mark{#1}%
1125     }%
1126     \@
1127 }{\par}

```

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

```

1128 \newenvironment{bibsection}[1][\refname]{%
1129   \begin{group}
1130     \protected@edef\@{%
1131       \endgroup
1132       \protect\section*{#1}%
1133       \protect\bib@div@mark{#1}%
1134     }%
1135     \@
1136 }{\par}

```

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

```

1137 \@ifundefined{chapter}{%
1138   \newenvironment{bibdiv}{\bibsection}{\endbibsection}
1139 }{%
1140   \newenvironment{bibdiv}{\bibchapter}{\endbibchapter}
1141 }
```

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

```

1142 \renewenvironment{thebibliography}[1]{%
1143   \bibdiv
1144   \biblist[\resetbiblist{#1}]%
1145 }{%
1146   \endbiblist
1147   \endbibdiv
1148 }
```

6.13 Name, journal and publisher abbreviations

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

\DefineName

```

1149 \newcommand{\DefineName}[2]{%
1150   \bib*{#1}{name}{name={#2}}%
1151 }
```

\DefineJournal

```

1152 \newcommand{\DefineJournal}[4]{%
1153   \bib*{#1}{periodical}{
1154     issn={#2},
1155     journal={#4}
1156   }%
1157 }
```

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

```

1158 \newcommand{\DefinePublisher}[4]{%
1159   \bib*{#1}{publisher}{%
1160     publisher={#3},
1161     address={#4}
1162   }%
1163 }
```

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

```

1164 \newcommand{\bibselect}{%
1165   \@ifstar{%
1166     \let\@bibdef\copy@bibdef
1167     \BibSelect
1168   }{%
1169     \let\@bibdef\selective@bibdef
1170     \BibSelect
1171   }%
1172 }
```

\BibSelect

```

1173 \newcommand{\BibSelect}[2][\bblname]{%
1174   \if@files
1175     \typeout{Trying to create bbl file `#1.bbl' ...}%
1176     \def\bibselect@msg{%
1177       \typeout{ ... rats. Unable to create bbl file.}%
1178     }%
1179     \let\@open@bbl@file\OpenBBLFile
1180     \for\@tempa:=#2\do{\ReadBibData{\@tempa}}%
1181   \fi
1182   \@close@bbl@file
1183   \@apply\g@undef\bibdefer@list
1184   \global\let\bibdefer@list\@empty
  Now read the .bbl file we just created.
1185   \let\@bibdef\normal@bibdef
1186   \@input{#1.bbl}%
1187   \let\BibSelect\MultipleBibSelectWarning
1188 }
```

\MultipleBibSelectWarning

```

1189 \newcommand\MultipleBibSelectWarning[2][{%
1190   \amsrefs@warning{%
1191     Multiple \string\bibselect 's found (only one
1192     \string\bibselect\space per biblist environment is allowed)%
1193   }%
1194 }
```

\bblname

```

1195 \def\bblname{\jobname}
```

\bib@dbfile

```

1196 \newread\bib@dbfile
```

\ReadBibData

```

1197 \newcommand{\ReadBibData}[1]{%
1198   \IfFileExists{#1.ltb}{%
1199     \openin\bib@dbfile=\@filef@und \relax
1200   }{%
1201     \IfFileExists{#1.ltx}{%
1202       \openin\bib@dbfile=\@filef@und \relax
1203     }{%
1204       \IfFileExists{#1.tex}{%
1205         \openin\bib@dbfile=\@filef@und \relax
1206       }{%
1207         \begingroup
1208           \NoBibDBFile{#1}%
1209           \let\ReadBibData@a\endgroup

```

```

1210      }%
1211    }%
1212  }%
1213  \ReadBibData@a
1214 }

```

\NoBibDBFile

```

1215 \def\NoBibDBFile#1{%
1216   \amsrefs@warning{No data file #1.ltb (.ltx, .tex) found}%
1217 }

```

\ReadBibData@a

```

1218 \def\ReadBibData@a{%
1219   \ProvidesFile{\@filef@und}\relax
1220   \begingroup
1221     \let\star@bibdef\defer@bibdef
1222     \ReadBibLoop
1223   \endgroup
1224   \closein\bib@dbfile
1225 }

```

\ReadBibLoop

```

1226 \def\ReadBibLoop{%
1227   \ifeof\bib@dbfile
1228     \xp\@gobble
1229   \else
1230     \read\bib@dbfile to\CurLine
1231     The \@empty is in case \CurLine is empty.
1232     \xp\ReadBibLoop@a\CurLine\@empty\@nil
1233   \fi
1234 }

```

\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.

```

1235 \long\def\ReadBibLoop@a#1#2\@nil{%
1236   \ifx\bib#1%
1237     \CurLine % just exec it
1238   \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.

```

1239     \ifx\endinput#1%
1240       \let\ReadBibLoop\@empty
1241     \else

```

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

```

1242       \xp\ReadBibLoop@b#1#2\@empty\@nil
1243     \fi
1244   \fi
1245 }

```

`\ReadBibLoop@b`

```

1246 \long\def\ReadBibLoop@b#1#2\@nil{%
1247     \ifx\bib#1%
1248         \CurLine % just exec it
1249     \fi
1250 }

1251 \let\bbl@out=\relax
1252 \let\bbl@write\@gobble
1253 \let\@open@bbl@file\relax
1254 \let\@close@bbl@file\relax

```

`\OpenBBLFile`

```

1255 \def\OpenBBLFile{%
1256     \if@filesw
1257         % Just use the next unused output stream
1258         \count@\count17
1259         \advance\count@\@ne
1260         \ifnum\count@<\sist@\@n
1261             \global\chardef\bbl@out=\count@
1262             \immediate\openout\bbl@out=\bblname.bbl\relax
1263             \global\let\@close@bbl@file\CloseBBLFile
1264             \gdef\bbl@write{\immediate\write\bbl@out}%
1265         \else
1266             \ch@ck\count@\sist@\@n\write
1267         \fi
1268     \fi
1269     \global\let\@open@bbl@file\relax
1270 }

```

`\CloseBBLFile`

```

1271 \def\CloseBBLFile{%
1272     \immediate\closeout\bbl@out\relax
1273     \global\let\@close@bbl@file\relax
1274     \global\let\bbl@write\@gobble
1275     \global\let\bbl@out\relax
1276 }

```

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
1277 \def\citesel@update#1#2#3#4#5#6{%
1278   \gdef#6{\citesel 1#2{#3}{#4}{#5}}%
1279 }
```

```
\citesel@number
1280 \def\citesel@number#1#2#3#4#5{#3}
```

```
\citesel@year
1281 \def\citesel@year#1#2#3#4#5{#4}
```

```
\citesel
1282 \let\citesel\citesel@number
```

6.15.2 The basic \cite command

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

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

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

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

Canceling the old L^AT_EX definition of \cite_l prevents certain problems that could arise with the showkeys package.

```
1283 \expandafter\let\csname cite \endcsname\relax
```

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

```
1284 \renewcommand{\cite}[2] [] {%
1285   \if\cite@single#2,\@gobble \else\MultipleCiteKeyWarning{#2}{#1}\fi
1286   \@ifempty{#1}{%
1287     \cites@o{#2}%
1288   }{%
1289     \ObsoleteCiteOptionWarning
1290     \cites@a{*{#1}}{#2}%
1291   }%
1292 }
```

\MultipleCiteKeyWarning

```
1293 \def\MultipleCiteKeyWarning#1#2{%
1294   \amsrefs@warning{%
1295     Use of \string\cites\space is recommended instead of %
1296     \string\cite\space\MessageBreak
1297     for multiple cites '#1'%
1298   \@ifnotempty{#2}{%
1299     \amsrefs@warning{Star option requires \string\citelist\space here}%
1300   }%
1301   \global\let\MultipleCiteKeyWarning\@gobbletwo
1302 }
```

\ObsoleteCiteOptionWarning

```
1303 \def\ObsoleteCiteOptionWarning{%
1304   \amsrefs@warning{%
1305     The form \string\cite{...}*{...} is recommended\MessageBreak
1306     instead of \string\cite[...]{...}%
1307   \global\let\ObsoleteCiteOptionWarning\@empty
1308 }
```

```

\cite@single
1309 \edef\cite@single#1,#2{\iffalse{\fi\iffalse{\fi\string}#2.\string}}

\cites@o
1310 \def\cites@o#1{\star@\{ \cites@oo{#1} \} \{ \}}

\cites@oo
1311 \def\cites@oo#1#2{\@ifempty{#2}{\cites@a{ \{ #1 \} } \{ \cites@a{ *{ #2 } } \{ #1 \} }}

\cites@a
1312 \def\cites@a#1#2{%
1313     \begingroup
1314     \toks@{\endgroup \cites@b{#1}}%
1315     \vdef\@tempa{#2}%
1316     \edef\@tempa{%
1317         \the\toks@ \@firstofone{\@xp\zap@space\@tempa} \@empty
1318     }%
1319     \@tempa,\@empty
1320     \edef\@tempa{\endgroup\@nx\citelist{\the\toks@}}%
1321     \@tempa
1322 }

\cites@b
1323 \def\cites@b#1#2,#3{%
1324     \begingroup
1325     \toks@{\InnerCite{#2}#1}%
1326     \ifx\@empty#3\@xp\@gobble\fi
1327     \cites@c#3%
1328 }

\cites@c
1329 \def\cites@c#1,#2{%
1330     \add@toks@{\InnerCite{#1}}%
1331     \ifx\@empty#2\@xp\@gobble\fi
1332     \cites@c#2%
1333 }

\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.
1334 \def\citeleft{[]}
1335 \def\citeright{[]}
1336 \def\citimid{\penalty9999 \space}
1337 \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.
1338 \def\citeAltPunct{; \ }

\citeform This is used for formatting the citation label. It can be used, for example, to
bolden the labels (as in amsbook and amsproc) or to do more elaborate things
such as convert the numbers to roman numerals. By default, it’s just a no-op.
Note that currently there is no corresponding macro for changing the for-
matting of \cite’s optional argument. This is probably a bug.
1339 \providecommand\citeform{\@firstofone}

```

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

```
1340 \DeclareRobustCommand{\citelist}{\@citelist}
```

`\@citelist`

```
1341 \def\@citelist#1{%
1342   \leavevmode
1343   \begingroup
1344     \@citestyle
1345     \citeleft\nopunct    % suppress first \citepunct
1346     \cite@begingroup
1347       \in@*{#1}%
1348       \ifin@
1349         \let\citepunct\citeAltPunct
1350       \fi
1351       \let\cite@endgroup\@empty
1352       \cites@init
1353       \def\citeleft{\@addpunct{\citepunct}}%
1354       \let\citeright\ignorespaces
1355       \def\cite{\InnerCite}%
1356       \process@citelist{#1}%
1357     \endgroup
1358     \citeright
1359   \endgroup
1360 }
```

`\@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).

```
1361 \providecommand{\@citestyle}{\m@th\upshape\mdseries}
```

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

```
1362 \def\cite@begingroup{\begingroup\let\cite@begingroup\relax}
```

`\cite@endgroup`

```
1363 \let\cite@endgroup\endgroup
```

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

```
1364 \def\cites@init{%
1365   \gdef\prev@names{???}%
1366   \let\cites@init\@empty
1367 }
```

`\InnerCite`

```
1368 \newcommand{\InnerCite}[1]{\star@\cite@a\citesel{#1}}{}}
```

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

Arguments:

`#1` <- `\CITESEL`.

`#2` <- `citekey`.

```
1369 \def\cite@a#1#2{%
1370   \BackCite{#2}%
1371   \cite@begingroup
1372     \cites@init
1373     \let\citesel#1\relax
1374   \ifx\citesel\citesel@author
```

```

1375         \let\citeleft\@empty
1376         \let\citeright\@empty
1377     \fi
1378     \begingroup
1379         \toks@{\endgroup \cite@b}%
1380         \vdef\@tempa{#2}%
1381         \edef\@tempa{%
1382             \the\toks@{\@firstofone{\@xp\zap@space\@tempa} \@empty}}%
1383         }%
1384     \@tempa
1385 }

```

`\cite@b` *Arguments:*

```

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

```

```

1386 \def\cite@b#1#2{%
1387     \@xp\cite@bc\csname b@#1\@xp\endcsname {#1}{#2}%
1388 }

```

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

```

1389 \def\cite@bc#1#2{%
1390     \ifx#1\@undefined \global\let#1\relax \fi
1391     \ifx#1\relax \global\let#1\empty@cite \fi
1392     \@xp\cite@nobib@test#1{}{}{}{}\@nil#1%
1393     \cite@cj#1%
1394 }

```

`\empty@cite`

```

1395 \def\empty@cite{\citesel 00{}{}{}}

```

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

Arguments:

```

#1 <- \citesel.
#2 <- cited?.
#3 <- used?.
#4 <- label.
#5 <- backrefs.
#6 <- \b@citekey.

```

```

1396 \def\cite@nobib@test#1#2#3#4#5\@nil#6{%
1397     \@ifempty{#4}{%
1398         \G@refundefinedtrue
1399         \UndefinedCiteWarning#6%
1400         \xdef#6{\@nx\citesel #2#3{%
1401             \@nx\CitePrintUndefined{\extr@cite#6}}{}{}}%
1402     }{}%
1403 }

```

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

```

1404 \def\UndefinedCiteWarning#1{%
1405     \@latex@warning{%
1406         Citation `~\extr@cite#1' on page \thepage\space undefined}%
1407 }

```

```

\CitePrintUndefined
1408 \DeclareRobustCommand{\CitePrintUndefined}[1]{%
1409     \begingroup\fontshape{n}\fontseries\mddefault \ttfamily ?#1\endgroup
1410 }

\CPU@normal This has to be a \let, not a \def.
1411 \let\CPU@normal\CitePrintUndefined

\cite@cj Arguments:
        #1 <- \b@citekey.
        #2 <- star-optional-arg.
1412 \def\cite@cj#1#2{%
1413     \leavevmode
1414     \begingroup
1415         \cite@cb#1% write info to aux file
1416         \ar@SK@cite#1%
1417         \@citeleft
1418         \ar@hyperlink{#1}%
1419         \@ifnotempty{#2}{\citelink{#2}}%
1420         \citeright
1421     \endgroup
1422     \ignorespaces % ignore spaces inside \citelist
1423     \cite@endgroup
1424 }

\@citeleft The following definition provides some indirection that helps to deal with
author-year object cites.
1425 \def\@citeleft{\citeleft}

\cite@cb
1426 \def\cite@cb#1{%
1427     \if@filesw
1428         \immediate\write\@auxout{\string\citation{\extr@cite#1}}%
1429     \fi
    Define \citesel to make \b@whatever update itself.
1430     \begingroup
1431         \let\citesel\citesel@update
1432         #1#1%
1433     \endgroup
1434 }

\extr@cite Extract citekey from \b@citekey.
1435 \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.)

```
1436 \DeclareRobustCommand{\cites}{\cites@a{}}
```

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

```
1437 \providecommand{\citen}{\ocite}
```

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

```
1438 \def\ycite{\cite}
```


\ycites

1439 \let\ycites\cites

\ocite

1440 \let\ocite\ycite

\ocites

1441 \let\ocites\cites

\fullcite

1442 \let\fullcite\cite

\fullocite

1443 \let\fullocite\ocite

\citeauthor

1444 \let\citeauthor\ycite

\citeauthory

1445 \let\citeauthory\ycite

6.15.4 The \nocite command

\nocite

1446 \renewcommand{\nocite}[1]{\othercites{#1}}

\othercites

```
1447 \newcommand{\othercites}[1]{%
1448     \cite@begingroup
1449     \let\BackCite\@gobble
1450     \let\cite@endgroup\@empty
1451     \def\citelist{\othercitelist}%
1452     \cites{#1}%
1453 }
```

\othercitelist

```
1454 \newcommand{\othercitelist}[1]{%
1455     \cite@begingroup
1456     \let\cite@endgroup\@empty
1457     \cites@init
1458     \let\citeleft\relax
1459     \let\citeright\ignorespaces
1460     \def\InnerCite{\OtherCite}%
1461     \def\cite@cj ##1##2{%
1462         \begingroup
1463         \exp\citesel##1%
1464         \cite@cb ##1%
1465         \endgroup
```

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

```
1466     \exp\ifx\csname b@*\endcsname ##1%
1467     \global\let\selective@bibdef\copy@bibdef
1468     \fi
1469     \ignorespaces
1470     \cite@endgroup
1471     }%
1472     #1\relax
1473     \endgroup
1474 }
```

`\OtherCite`

```
1475 \def\OtherCite#1{\cite@a\citesel@other{#1}{}}{}}
```

`\citesel@other`

```
1476 \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.

```
1477 \@namedef{b@*}{\citesel 11{*}{*}{*}}
```

6.15.5 Citation sorting

`\process@citelist@sorted`

```
1478 \def\process@citelist@sorted#1{%
1479   \ifx\citesel\citesel@number
1480     \cite@sorted@s #1\cite@sorted@e
1481   \else
1482     \NonNumericCiteWarning
1483     \process@citelist@unsorted{#1}%
1484   \fi
1485 }
```

`\NonNumericCiteWarning`

```
1486 \def\NonNumericCiteWarning{%
1487   \amsrefs@warning{%
1488     Unable to confirm that cite keys are numeric: not sorting%
1489   }%
1490 }
```

`\process@citelist@unsorted`

```
1491 \def\process@citelist@unsorted#1{%
1492   \ignorespaces#1\relax
1493 }
```

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

```
1494 \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.

```
1495 \def\CPU@sort#1{2147483647}
```

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

```
1496 \def\cite@sorted@s{%
1497   \begingroup
1498     \let\CitePrintUndefined\CPU@sort
1499     \let\cite@cjs\cite@cj
1500     \let\cite@cj\cite@compress
1501   \begingroup
1502     \toks@{@emptytoks
1503     \let\cite@cj\cite@sort
1504     \ignorespaces
1505 }
```

`\cite@sorted@e`

```
1506 \def\cite@sorted@e{%
1507   \@xp\endgroup
1508   \the\toks@
1509   \cite@dash
```

```

1510      \prev@cite
1511  \endgroup
1512 }

```

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

Arguments:

```

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

```

```

1513 \def\cite@sort#1#2{%
1514   \safe@set\@tempcnta#1% highest number so far
1515   \toks@{\cite@cj#1{#2}}%
1516   \@temptokena\toks@
1517   \let\cite@cj\cite@sort@a
1518   \ignorespaces
1519 }

```

\cite@sort@a

```

1520 \def\cite@sort@a#1#2{%
1521   \safe@set\@tempcntb#1%
1522   \ifnum\@tempcntb > \@tempcnta
1523     \add@toks@{\cite@cj#1{#2}}%
1524     \@tempcnta\@tempcntb
1525   \else
1526     \let\cite@cj\cite@sort@b
1527     \toks@\@emptytoks
1528     \def\@tempb{\add@toks@{\cite@cj#1{#2}}}%
1529     \the\@temptokena
1530     \@tempb
1531     \let\cite@cj\cite@sort@a
1532   \fi
1533   \@temptokena\toks@
1534   \ignorespaces
1535 }

```

\cite@sort@b

```

1536 \def\cite@sort@b#1#2{%
1537   \safe@set\count@#1%
1538   \ifnum\@tempcntb < \count@
1539     \@tempb
1540     \let\@tempb\@empty
1541   \fi
1542   \add@toks@{\cite@cj#1{#2}}%
1543   \ignorespaces
1544 }

```

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`

1545 `\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).

```

1546 \def\prev@cite@cb{%
1547   \ifx\@prev@cite\@empty
1548   \else
1549     \begingroup
1550       \def\cite@print##1##2{%
1551         \cite@cb##1%
1552       }%
1553       \prev@cite
1554     \endgroup
1555   \fi
1556 }
```

`\cite@print`

```
1557 \def\cite@print#1#2{%
1558     \begingroup
1559     \let\CitePrintUndefined\CPU@normal
1560     \cite@cjs#1{#2}%
1561     \endgroup
1562 }
```

`\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.

```
1563 \let\cite@dash\@empty
```

`\print@one@dash`

```
1564 \def\print@one@dash{%
1565     \textendash \nopunct
1566     \let\cite@dash\@empty
1567 }
```

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:

```
1568 \def\cite@compress#1#2{%
1569     \cite@print#1{#2}%
1570     \@ifempty{#2}{%
1571         \safe@set\@tempcnta#1%
1572         \let\cite@cj\cite@compress@a
1573     }{}%
1574 }
```

`\cite@compress@a` State 1:

```
1575 \def\cite@compress@a#1#2{%
1576     \@ifempty{#2}{%
1577         \advance\@tempcnta\@ne
1578         \safe@set\@tempcntb#1%
1579         \ifnum\@tempcnta=\@tempcntb
1580             \def\prevcite{\cite@print#1{}}%
1581             \let\cite@cj\cite@compress@b
1582         \else
1583             \cite@print#1{}%
1584             \@tempcnta\@tempcntb
1585         \fi
1586     }{%
1587         \cite@print#1{#2}%
1588         \let\cite@cj\cite@compress
1589     }%
1590 }
```

`\cite@compress@b` State 2:

```
1591 \def\cite@compress@b#1#2{%
1592     \@ifempty{#2}{%
1593         \advance\@tempcnta\@ne
1594         \safe@set\@tempcntb#1%
```

```

1595         \ifnum\@tempcnta=\@tempcntb
1596             \let\cite@dash\print@one@dash
1597             \prev@cite@cb
1598             \def\prev@cite{\cite@print#1{}}%
1599         \else
1600             \cite@dash
1601             \prev@cite
1602             \let\prev@cite\@empty
1603             \cite@print#1{}%
1604             \@tempcnta\@tempcntb
1605             \let\cite@cj\cite@compress@a
1606         \fi
1607     }{%
1608         \cite@dash
1609         \prev@cite
1610         \let\prev@cite\@empty
1611         \cite@print#1{#2}%
1612         \let\cite@cj\cite@compress
1613     }%
1614 }

```

6.15.7 Munging the .aux file

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

```

1615 \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.

```

1616 \AtBeginDocument{%
1617     \if@filesw
1618         \immediate\write\@auxout{%
1619             \string\@ifundefined{amsrefs@bibcite}{}{}%
1620             \string\let\string\bibcite\string\amsrefs@bibcite
1621         }%
1622     }%
1623 \fi

```

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

```

1624     \let\bibcite\amsrefs@bibcite
1625 }

```

`\bibcite@a` *Arguments:*

```

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

```

```

1626 \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.

```

1627     \ifx\relax#1%
1628         \gdef#1{\citesel 00#2{}}%
1629     \else
1630         \begingroup
1631             \@xp\bibcite@b\@xp#1#1{#2}%
1632         \endgroup
1633     \fi
1634 }

```

```

\cite@b Arguments:
    #1 <- \b@citekey.
    #2 <- \citesel.
    #3 <- cited?.
    #4 <- used?.
    #5 <- label.
    #6 <- year.
    #7 <- backrefs.
    #8 <- {newlabel}{newyear}.
1635 \def\cite@b#1#2#3#4#5#6#7#8{\gdef#1{\citesel#3#4#8{#7}}}

\citedest The \citedest command goes into the .aux file to provide back-reference sup-
port.
1636 \newcommand{\citedest}[1]{\@xp\cite@dest\csname b@#1\endcsname}

\cite@dest
1637 \def\cite@dest#1{%
1638     \ifx\relax#1%
1639         \gdef#1{\citesel 00{}{}{}}%
1640     \fi
1641     \@xp\cite@dest@b\@xp#1#1%
1642 }

\cite@dest@b Arguments:
    #1 <- \b@citekey.
    #2 <- \citesel.
    #3 <- cited?.
    #4 <- used?.
    #5 <- label.
    #6 <- year.
    #7 <- backrefs.
    #8 <- {more backrefs}.
1643 \def\cite@dest@b#1#2#3#4#5#6#7#8{%
1644     \@ifempty{#7}{%
1645         \def#1{\citesel #3#4{#5}{#6}{#8}}%
1646     }{%
1647         \gdef#1{\citesel #3#4{#5}{#6}{#7,#8}}%
1648     }%
1649 }

```

6.15.8 Back references

```

\ifBR@verbose
1650 \@ifundefined{ifBR@verbose}{\let\ifBR@verbose\iffalse \let\fi\fi}{%

\BackCite
1651 \let\BackCite\@gobble

\back@cite
1652 \def\back@cite#1{%
1653     \ifBR@verbose
1654         \PackageInfo{backref}{back cite \string `extr@cite#1'}%
1655     \fi
1656     \Hy@backout{#1}%
1657 }

```

`\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.

```
1658 \def\print@backrefs#1{%
1659     \space\SentenceSpace$\uparrow$\csname br@#1\endcsname
1660 }
```

`\PrintBackRefs`

```
1661 \let\PrintBackRefs\@gobble
```

6.15.9 hyperref, showkeys and shaderef support

`\shade@cite`

```
1662 \newcommand{\shade@cite}{\printref}
```

`\format@cite`

```
1663 \def\format@cite#1{\shade@cite{\citeform{#1}}}
```

`\ar@hyperlink`

```
1664 \def\ar@hyperlink#1{%
1665     \hyper@link[cite]{\cite.\extr@cite#1}{\format@cite{#1}}%
1666 }
```

`\ar@SK@cite`

```
1667 \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.

```
1668 \AtBeginDocument{%
1669     \@ifpackageloaded{shaderef}{\@firstofone
1670         \let\shade@cite\@firstofone
1671     }%
1672     \@ifpackageloaded{hyperref}{\@firstofone
1673         \def\ar@hyperlink{\format@cite}%
1674         \let\hyper@anchorstart\@gobble
1675         \let\hyper@anchorend\relax
1676         \let\Hy@raisedlink\@firstofone
1677     }%
1678     \@ifpackageloaded{showkeys}{\@firstofone
1679         \@ifpackagewith{showkeys}{notcite}{\@firstofone
1680             \let\ar@SK@cite\@gobble
1681         }{}
1682     }{}%
1683     \let\ar@SK@cite\@gobble
1684     \let\SK@@label\@gobble
1685     \let\SK@\@gobbletwo
1686 }%
1687 }
```

6.16 Lexical structure of names

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

- letters and punctuation (i.e., characters with catcode 11 or 12),
- ties (the token \sim_{13}),
- accent commands, such as `\` or `\k`,

- text symbol macros, such as `\i`, `\ae` or `\cprime`,
- grouping characters (braces).

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

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

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

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

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

6.16.1 Text accents

Syntactically, a text accent is a macro that takes a single, undelimited argument, i.e., it has a “prototype” of `macro:#1->`. Semantically, the implication is that it takes a letter (the *base*) as an argument and produces a glyph that for certain purposes can be considered equivalent to the base (see the discussion of stem comparison on page 71).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 \'
```

```
1688 \let\amsrefs@textaccents\@empty
```

`\DeclareNameAccent` *Arguments:*

`#1 <- accent.`

```
1689 \def\DeclareNameAccent{%
1690     \@lappend\amsrefs@textaccents
1691 }
```

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

```
1692 \DeclareNameAccent\"
1693 \DeclareNameAccent\'
1694 \DeclareNameAccent\.
1695 \DeclareNameAccent\=
1696 \DeclareNameAccent\^
1697 \DeclareNameAccent\`
1698 \DeclareNameAccent\~%
1699 \DeclareNameAccent\b
1700 \DeclareNameAccent\c
```

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

```

1701 \DeclareNameAccent\d
1702 \DeclareNameAccent\H
1703 \DeclareNameAccent\k
1704 \DeclareNameAccent\r
1705 \DeclareNameAccent\t
1706 \DeclareNameAccent\u
1707 \DeclareNameAccent\v

```

From mathscinet:

```

1708 \DeclareNameAccent\utilde
1709 \DeclareNameAccent\uarc
1710 \DeclareNameAccent\dudot
1711 \DeclareNameAccent\lfhook
1712 \DeclareNameAccent\udot
1713 \DeclareNameAccent\polhk
1714 \DeclareNameAccent\soft

```

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

```

1715 \DeclareNameAccent\etalchar
1716 %\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`.

```
1717 \let\amsrefs@textsymbols\@empty
```

`\DeclareNameSymbol` *Arguments:*

```

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

```

```

1718 \def\DeclareNameSymbol#1#2{%
1719   \@lappend\amsrefs@textsymbols#1%
1720   \@lappend\amsrefs@textsymbols#2%
1721   \ifx#1#2\else
1722     \@lappend\amsrefs@textsymbols#2%
1723     \@lappend\amsrefs@textsymbols#2%
1724   \fi
1725 }

```

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

Here are the standard \LaTeX and mathscinet text symbols.

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

```

1726 \DeclareNameSymbol\i\i
1727 \DeclareNameSymbol\j\j
1728 \DeclareNameSymbol\AE\ae
1729 \DeclareNameSymbol\OE\oe
1730 \DeclareNameSymbol\O\o
1731 \DeclareNameSymbol\DH\dh
1732 \DeclareNameSymbol\DJ\dj
1733 \DeclareNameSymbol\L\l
1734 \DeclareNameSymbol\NG\ng
1735 \DeclareNameSymbol\SS\ss
1736 \DeclareNameSymbol\TH\th

```

From mathscinet :

```

1737 \DeclareNameSymbol\Dbar\dbar
1738 \DeclareNameSymbol\lasp\lasp
1739 \DeclareNameSymbol\rasp\rasp
1740 \DeclareNameSymbol\cprime\cprime
1741 \DeclareNameSymbol\cdprime\cdprime
1742 \DeclareNameSymbol\bud\bud
1743 \DeclareNameSymbol\cydot\cydot

```

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

```

1744 \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 $\text{P{'o}lya}$. (This might result in a redundant layer of braces if the original text contained, say, “ $\text{P{'o}lya}$ ”, but that’s ok.) This lets us extract the first n characters from a name by using \TeX ’s macro argument-gobbling mechanism without worrying that an accent will be separated from its base letter. As a bonus, it also replaces ties (\sim) by spaces.

```

1745 \def\normalize@edef#1#2{%
1746     \begingroup
1747         \@apply\auto@protect\amsrefs@textsymbols
1748         \@apply\wrap@accent\amsrefs@textaccents

```

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

```

1749         \let\@tabacckludge\use@accent
1750         \let~\space
1751         \edef\@tempa{\def\@nx#1{#2}}%
1752     \@xp\endgroup
1753     \@tempa
1754 }

```

`\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

```
1755 \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}`.

```
1756 \def\wrap@accent#1{%
1757   \def#1##1{{\@nx#1##1}}%
1758 }
```

`\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.

```
1759 \def\lc@edef#1#2{%
1760   \begingroup
1761     \let\@tabacckludge\use@accent %%%
1762     \@apply\auto@protect\amsrefs@textaccents
1763     \@apply\lc@do\amsrefs@textsymbols
1764     \edef\@tempa{\lowercase{\def\@nx#1{#2}}}%
1765     \@xp\endgroup
1766     \@tempa
1767 }
```

`\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.

```
1768 \def\lc@do#1\do#2{%
1769   \ifx#1#2%
1770     \auto@protect#1%
1771   \else
1772     \def#1{#2}%
1773   \fi
1774 }
```

`\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`.

```
1775 \def\purge@edef#1#2{%
1776   \begingroup
1777     \@apply\auto@protect\amsrefs@textsymbols
1778     \let-\space
1779     \@apply\purge@accent\amsrefs@textaccents
1780     \let\@tabacckludge\@gobble
```

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

```
1781   \def\i{i}%
1782   \def\j{j}%
```

```

1783      \edef\@tempa{#2}%
1784      \toks@\@emptytoks
1785      \@xp\purge@edef@ \@tempa \@nil
1786      \edef\@tempa{\def\@nx#1{\the\toks@}}%
1787      \@xp\endgroup
1788      \@tempa
1789 }

```

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

```

1790 \def\purge@edef@{%
1791     \futurelet\@let@token
1792     \purge@edef@a
1793 }

```

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

```

1794 \def\purge@edef@a#1{%

```

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

```

1795     \ifx\@let@token\@nil
1796         \let\@tempa\@empty
1797     \else

```

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

```

1798         \ifx\@let@token\bgroup
1799             \def\@tempa{%
1800                 \purge@edef@ #1\@nil
1801                 \purge@edef@
1802             }%
1803         \else

```

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

```

1804             \add@toks@{#1}%
1805             \let\@tempa\purge@edef@
1806         \fi
1807     \fi
1808     \@tempa
1809 }

```

`\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).

```

1810 \def\purge@accent#1{%
1811     \def##1##1{##1}%
1812 }

```

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.)

```
1813 \def\name@split{%
1814     \get@namepart\bib'surname\name@split@given
1815 }
```

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

```
1816 \def\name@split@given{%
1817     \get@namepart\bib'given\name@split@jr
1818 }
```

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

```
1819 \def\name@split@jr{%
1820     \get@namepart\bib'jr\name@split@finish
1821 }
```

`\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.

```
1822 \def\name@split@finish{%
1823     \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.

```
1824 %%     \ifx\@empty\bib'given
1825 %%         \NameCheck \bib'surname ??\@nil
1826 %%     \else
1827         \extract@initials\bib'given
1828 %%     \fi
1829     \@nilgobble
1830 }
```

`\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.

```
1831 \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.

```
1832     \toks@{#1}%
1833     \@temptokena{#2}%
1834     \get@namepart@a
1835 }
```

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

```
1836 \def\get@namepart@a{%
1837     \futurelet\@let@token
1838     \get@namepart@b
1839 }
```

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

```
1840 \def\get@namepart@b{%
1841     \ifx\@let@token\@sptoken
1842         \xp\get@namepart@c
1843     \else
1844         \xp\get@namepart@d
1845     \fi
1846 }
```

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

```
1847 \def\get@namepart@c{%
1848     \after@deleting@token\get@namepart@a
1849 }
```

`\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.

```
1850 \def\get@namepart@d{%
1851     \futurelet\@let@token
1852     \get@namepart@e
1853 }
```

`\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.

```
1854 \def\get@namepart@e{%
1855     \ifx\@let@token,%
1856         \xp\let\the\toks@\@empty
1857         \edef\@tempa{%
1858             \@nx\after@deleting@token\the\@temptokena
1859         }%
1860         \xp\@tempa
1861     \else
1862         \xp\get@namepart@f
1863     \fi
1864 }
```

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

Note that the arguments are not expanded.

⁵More or less. If the second argument is brace-delimited, the braces will be lost. But as mentioned above (page 62), we don't really care.


```

1865 \def\get@namepart@f#1#2,{%
1866     \ifx\@let@token\bgroup
1867         \@xp\def\the\toks@{{#1}#2}%
1868     \else
1869         \@xp\def\the\toks@{{#1#2}%
1870     \fi
1871     \the\@temptokena
1872 }

```

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

```

1873 \def\EmptyNameWarning{\amsrefs@warning{Empty contributor name}}

```

6.18 Extracting initials

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

```

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

```

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

```

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

```

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

6.18.1 The algorithm

As a running example, consider the following contrived input:

```
\'E.-P\^{\i }erre J.K. M
```

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

We precede by stages.

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

```
{\'E}. -P{\^{\i }erre J.K. M
```

We also replace ~s by spaces at this stage.

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


```
{\'E}. \ini@hyphen P{\'i }erre J.K. M
```
3. Add a space after each period:


```
{\'E}. \ini@hyphen P{\'i }erre J. K. M
```
4. Now we have the name as a list of space-separated components. (In our example, the components are “`{\'E}.`”, “`\ini@hyphen`”, “`P{\'i }erre`”, “`J.`”, “`K.`”, and “`M`”.) We loop through the components and replace each one by its “initialized” form. There are four cases:
 - (a) The component ends in a period. Copy it and add the token `~`. (In our example, these are the components “`{\'E}.`”, “`J.`” and “`K.`”.)
 - (b) The component consists of a single (possibly compound) character without a period. Again, copy it and add `~`. (In our example, this is the component “`M`”.)
 - (c) The component is the token `\ini@hyphen`. Copy it.
 - (d) The component consists of two or more (possibly compound) characters without a period (e.g., “`P{\'i }erre`”). Copy the first character and add the tokens `~.`.
5. The token list generated above will end with an unwanted `~`. Delete it.

The end result is

```
{\'E}~\ini@hyphen P~J~K~M
```

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

6.18.2 The implementation

`\extract@initials` This is pretty straightforward.

```
1874 \def\extract@initials#1{%
1875     \begingroup
1876     \auto@protect\ini@hyphen
1877     \auto@protect\nobreakspace
1878     \let~\relax
1879     \@apply\auto@protect\amsrefs@textsymbols
1880     \@apply\auto@protect\amsrefs@textaccents
1881     \normalize@edef\@tempa{#1}%
1882     \ifx\@tempa\empty
1883     \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.-”)

```
1884         \process@hyphens\@tempa
1885         \process@dots\@tempa
1886         \process@names\@tempa
1887         \chomp\@tempa{~}%
1888     \fi
1889     \edef\@tempa{\def\nx\bib'initials{\@tempa}}%
1890 \exp\endgroup
1891 \@tempa
1892 }
```

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

```
1893 \def\ini@hyphen{\unskip-\nobreak}
```

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

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

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

```
1894 \def\process@hyphens#1{%
1895     \begingroup
1896     \toks@{\emptytoks
1897     \@xp\process@one@hyphen #1-\@nil
1898     \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.)

```
1899     \@chomp\@tempa{ \ini@hyphen\space}%
1900     \edef\@tempa{\def\@nx#1{\@tempa}}%
1901     \@xp\endgroup
1902     \@tempa
1903 }
```

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

```
1904 \def\process@one@hyphen{%
1905     \futurelet\@let@token
1906     \process@one@hyphen@a
1907 }
```

`\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.

```
1908 \def\process@one@hyphen@a{%
1909     \ifx\@let@token\@nil
1910         \let\fsa@n\@gobble
1911     \else
1912         \ifx\@let@token -%
1913             \TrailingHyphenWarning
1914             \let\fsa@n\process@one@hyphen@b
1915         \else
1916             \ifx\@let@token\@sptoken
1917                 \let\fsa@n\process@one@hyphen@b
1918             \else
1919                 \let\fsa@n\process@one@hyphen@c
1920             \fi
1921         \fi
1922     \fi
1923     \fsa@n
1924 }
```

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

```
1925 \def\process@one@hyphen@b{%
1926     \after\deleting@token\process@one@hyphen
1927 }
```

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

```

1928 \def\process@one@hyphen@c#1#2--{%
1929     \ifx\bgroup\@let@token
1930         \add@toks@{#1}#2 \ini@hyphen\space}%
1931     \else
1932         \add@toks@{#1#2 \ini@hyphen\space}%
1933     \fi
1934     \futurelet\@let@token
1935     \process@one@hyphen@d
1936 }

```

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

```

1937 \def\process@one@hyphen@d{%
1938     \ifx\@let@token\@nil
1939         \xp@gobble
1940     \else
1941         \xp\process@one@hyphen
1942     \fi
1943 }

```

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

```

1944 \def\TrailingHyphenWarning{%
1945     \amsrefs@warning{Trailing hyphen deleted from name}%
1946 }

```

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

```

1947 \def\process@dots#1{%
1948     \begingroup
1949         \toks@{\emptytoks
1950             \xp\process@one@dot #1.\@nil
1951             \edef\@tempa{\the\toks@}%
1952             \@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.

```

1953         \@chomp\@tempa{ }%
1954         \edef\@tempa{\def\@nx#1{\@tempa}}%
1955     \xp@endgroup
1956     \@tempa
1957 }

```

`\process@one@dot`

```

1958 \def\process@one@dot{%
1959     \futurelet\@let@token
1960     \process@one@dot@a
1961 }

```

`\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`.

```

1962 \def\process@one@dot@a{%
1963     \ifx\@let@token.%
1964         \def\fsa@n{\after@deleting@token\process@bare@dot}%
1965     \else
1966         \ifx\@let@token\@sptoken
1967             \let\fsa@n\process@one@dot@b
1968         \else

```

```

1969         \let\fsa@n\process@one@dot@c
1970     \fi
1971 \fi
1972 \fsa@n
1973 }

```

\process@bare@dot

```

1974 \def\process@bare@dot{%
1975     \add@toks@{. }%
1976     \futurelet\@let@token
1977     \process@one@dot@d
1978 }

```

\process@one@dot@b

```

1979 \def\process@one@dot@b{%
1980     \after@deleting@token\process@one@dot
1981 }

```

\process@one@dot@c

```

1982 \def\process@one@dot@c#1#2.{%
1983     \ifx\bgroup\@let@token
1984         \add@toks@{{#1}#2. }%
1985     \else
1986         \add@toks@{#1#2. }%
1987     \fi
1988     \futurelet\@let@token
1989     \process@one@dot@d
1990 }

```

\process@one@dot@d

```

1991 \def\process@one@dot@d{%
1992     \ifx\@let@token\@nil
1993         \@xp\@gobble
1994     \else
1995         \@xp\process@one@dot
1996     \fi
1997 }

```

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

```

1998 \def\process@names#1{%
1999     \begingroup
2000         \toks@\@emptytoks
2001         \@xp\extract@initial #1 \@nil
2002         \edef\@tempa{\def\@nx#1{\the\toks@}}%
2003     \@xp\endgroup
2004     \@tempa
2005 }

```

\extract@initial Scan through the token stream replacing words by their initials until we hit the terminating '₁

```

2006 \def\extract@initial{%
2007     \futurelet\@let@token
2008     \extract@initial@a
2009 }

```

\extract@initial@a As with \process@one@hyphen@a, the test for '₁ 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.

```

2010 \def\extract@initial@a{%
2011     \ifx\@let@token\@nil
2012         \let\fsa@n\@gobble
2013     \else
2014         \ifx\@let@token\@sptoken
2015             \let\fsa@n\extract@initial@b
2016         \else
2017             \let\fsa@n\extract@initial@c
2018         \fi
2019     \fi
2020     \fsa@n
2021 }
```

`\extract@initial@b`

```

2022 \def\extract@initial@b{%
2023     \after@deleting@token\extract@initial
2024 }
```

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

```

2025 \def\extract@initial@c#1#2 {%
2026     \ifx\@let@token\bgroup
```

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

```

2027         \@extract@initial {{#1}}#2\@nil
2028     \else
2029         \@extract@initial #1#2\@nil
2030     \fi
2031     \futurelet\@let@token
2032     \extract@initial@d
2033 }
```

`\extract@initial@d`

```

2034 \def\extract@initial@d{%
2035     \ifx\@let@token\@nil
2036         \@xp\@gobble
2037     \else
2038         \@xp\extract@initial
2039     \fi
2040 }
```

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

```

2041 \def\@extract@initial#1#2\@nil{%
2042     \ifx\ini@hyphen#1%
2043         \add@toks@{\ini@hyphen}%
2044     \else
2045         \in@{.\@nil}{#1#2\@nil}% Look for a period at the end of the name
2046         \ifin@
2047             \add@toks@{#1#2~}%
2048         \else
2049             \count@chars\@tempcnta{#1#2}%
2050             \ifnum\@tempcnta > \@ne
2051                 \add@toks@{#1.~}%
2052             \else
2053                 \add@toks@{#1~}%
2054             \fi
2055         \fi
```

```

2056   \fi
2057 }

```

`\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.

```

2058 \def\count@chars#1#2{%
2059   \begingroup
2060     \@tempcnta\z@
2061     \@count@chars#2\@nil
2062     \edef\@tempb{#1=\the\@tempcnta\relax}%
2063   \xp@endgroup
2064   \@tempb
2065 }

```

`\@count@chars`

```

2066 \def\@count@chars#1{%
2067   \ifx #1\@nil
2068   \else
2069     \advance\@tempcnta\@ne
2070     \@xp\@count@chars
2071   \fi
2072 }

```

6.19 Generating alphabetic labels

6.19.1 The algorithm

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

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

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

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

2. The year part. If the `y2k` option is in force, or if the year is less than 1901, the entire year is used. Otherwise the last two digits of the year are used.⁷ The combination of author part and year part will be referred to as the *stem*.

3. The suffix. If two or more items have the same stems, a suffix consisting of a lowercase latin letter will be appended to each label to make it unique.

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

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

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

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

6.19.2 The implementation

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

2075 \let\previous@year\@empty
2076 \let\current@year\@empty
```

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

```
2077 \def\append@to@stem{\global\@concat\current@stem}
```

```
\generate@alphalabel
```

```
2078 \def\generate@alphalabel{%
```

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

```
2079   \ifx\bib'label\@empty
2080     \begingroup
```

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

```
2081       \global\let\previous@stem\current@stem
2082       \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.

```
2083       \@apply\auto@protect\amsrefs@textsymbols
2084       \@apply\auto@protect\amsrefs@textaccents
2085       \auto@protect\name
2086       \auto@protect\etaltext
2087       \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.

```
2088       \get@numberof\@tempcnta\name\@tempa
2089       \calc@author@part
```

Next append the year part.

```
2090       \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.

```
2091       \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.)

```
2092       \edef\@tempa{%
2093         \def\@nx\bib'label{%
2094           \current@stem
2095           \alpha@label@suffix
2096         }%
2097       }%
```



```

2098      \xp\endgroup
2099      \@tempa
2100  \fi
2101 }

\calc@author@part

2102 \def\calc@author@part{%
2103   \ifnum \@tempcnta = 1
2104     \xp\@oneauthorlabel\exp{\@tempa}%
2105   \else
2106     \xp\@multiauthorlabel\exp{\@tempa}%
2107   \fi
2108 }

\@firstone This extracts the first character from a properly prepared author name (i.e.,
            one in which accents are properly wrapped).
2109 \def\@firstone#1{\@car#1\@empty\@nil}

\@firstthree And this extracts the first three characters.
2110 \def\@firstthree#1{\@carcube#1\@empty\@empty\@empty\@nil}

\@nametoken

2111 \let\@nametoken\@firstone

\hyph@to@space

2112 \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).
2113 \def\@marknames#1{%
2114   \@ifnotempty{#1}{\surround@names#1 ' }%
2115 }

\surround@names

2116 \def\surround@names#1 {%
2117   \ifx '#1%
2118   \else
2119     \nx\@nametoken{#1}%
2120     \xp\surround@names
2121   \fi
2122 }

\extract@surnames

2123 \def\extract@surnames#1#2{%
2124   \get@namepart\@tempb\@nilgobble #2,\@nil
2125   \edef\@tempb{\@nx\@marknames{\xp\hyph@to@space\@tempb\@gobble-}}%
2126   \edef#1{\@tempb}%
2127 }

\@oneauthorlabel This is the easy case.
2128 \newcommand{\@oneauthorlabel}[1]{%
2129   \def\name##1{%
2130     \extract@surnames\@tempa{##1}%
2131     \get@numberof\@tempcnta\@nametoken\@tempa
2132     \ifnum \@tempcnta = 1
2133       \let\@nametoken\@firstthree
2134     \fi
2135     \append@to@stem{\@tempa}%

```

```

2136    }%
2137    #1%
2138 }

```

`\@threeauthors`

```

2139 \def\@threeauthors\name#1\name#2\name#3#4\@empty{%
2140     \name{#1}\name{#2}\name{#3}%
2141     \append@to@stem{\etalchar{+}}%
2142 }

```

`\@multiauthorlabel`

```

2143 \newcommand{\@multiauthorlabel}[1]{%
2144     \def\name##1{%
2145         \ifx\etaltext ##1%
2146             \def\@tempa{\@nx\etalchar{+}}%
2147             \let\name\@gobble
2148         \else
2149             \extract@surnames\@tempa{##1}%
2150         \fi
2151         \append@to@stem{\@tempa}%
2152     }%
2153     \ifnum \@tempcnta > 4 \xp \@threeauthors \fi
2154     #1\@empty
2155 }

```

`\etalchar`

```

2156 \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.

```

2157 \def\year@short#1#2#3#4\@nil{#3#4}

```

`\append@label@year`

```

2158 \def\append@label@year{%
2159     \safe@set\@tempcnta\bib@year
2160     \edef\bib@citeyear{the\@tempcnta}%
2161     \append@to@stem{%
2162         \ifx\bib@year\@empty
2163             \else
2164                 \xp\year@short \bib@citeyear \@nil
2165             \fi
2166     }%
2167 }

```

```

2168 \let\alpha@label@suffix\@empty
2169
2170 \newcount\alpha@suffix
2171 \alpha@suffix\@ne
2172 \let\@suffix@format\@alph

```

`\calc@alpha@suffix`

```

2173 \def\calc@alpha@suffix{%
2174     \@tempwafalse
2175     \compare@stems\previous@stem\current@stem
2176     \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.

```

2177     \ifx\previous@year\current@year
2178     \@tempswatrue
2179     \fi
2180   \fi
2181   \if@tempswa
2182     \global\advance\alpha@suffix\@ne
2183     \edef\alpha@label@suffix{\@suffix@format\alpha@suffix}%
2184     \ifnum\alpha@suffix=\tw@
2185       \immediate\write\@auxout{%
2186         \string\ModifyBibLabel{\prev@citekey}%
2187       }%
2188     \fi
2189   \else
2190     \let\alpha@label@suffix\@empty
2191     \global\alpha@suffix\@ne
2192     \@xp\ifx \csname b@\current@citekey @suffix\endcsname \relax
2193     \else
2194       \edef\alpha@label@suffix{\@suffix@format\alpha@suffix}%
2195     \fi
2196   \fi
2197 }

\ifsame@stems

2198 \newif\ifsame@stems

\compare@stems

2199 \def\compare@stems#1#2{%
2200   \begingroup
2201     \purge@edef\@tempa{#1}%
2202     \purge@edef\@tempb{#2}%
2203     \lc@edef\@tempa{\@tempa}%
2204     \lc@edef\@tempb{\@tempb}%
2205     \ifx\@tempa\@tempb
2206       \def\@tempa{\same@stemstrue}%
2207     \else
2208       \def\@tempa{\same@stemsfalse}%
2209     \fi
2210   \@xp\endgroup
2211   \@tempa
2212 }

\ModifyBibLabel

2213 \def\ModifyBibLabel#1{%
2214   \global\@xp\let\csname b@#1@suffix\endcsname\@empty
2215 }

```

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 89 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

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

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

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

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

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

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:

$$\backslash\mathrm{PrintSeries}\{\backslash\mathrm{m}\} \{f_0\} \{p_1\}\{i_1 f_1\} \{p_2\}\{i_2 f_2\} \{p_3\}\{i_3 f_3\} \{S\} \{\backslash\mathrm{m}\{T_1\} \dots \backslash\mathrm{m}\{T_n\}\} \{E\}$$

and our comma-separated list example becomes

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

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

2216 `\newcount\series@index`

`\PrintSeries`

2217 `\def\PrintSeries#1#2#3#4#5#6#7#8{%`
 2218 `\begingroup`

```

2219      \def\series@add@a{#2}%
2220      \def\series@add@b{\SwapBreak{#3}#4}%
2221      \def\series@add@c{\SwapBreak{#5}#6}%
2222      \def\series@add@d{\SwapBreak{#7}#8}%
2223      \def\series@add@e{\SwapBreak{#7}}%
2224      \PrintSeries@a{#1}%
2225 }

```

`\PrinteSeries@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`.

```

2226 \def\PrintSeries@a#1#2#3#4{%
2227     \get@numberof\@tempcnta#1{#3}%
2228     \chardef\series@total=\@tempcnta
2229     \ifnum\series@total=\@ne
2230         \let\SingularPlural\@firstoftwo
2231     \else
2232         \let\SingularPlural\@secondoftwo
2233     \fi
2234     \series@index=\z@
2235     \let#1\series@add
2236     #2#3#4\relax
2237     \endgroup
2238 }

```

`\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`.

```

2239 \def\series@add#1{%
2240     \advance\series@index\@ne
2241     \ifx\etaltext#1\relax
2242         \ifnum\series@index=\tw@
2243             \def\@tempa{\space\SubEtal}%
2244         \else
2245             \def\@tempa{\series@add@e\space\SubEtal}%
2246         \fi

```

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

```

2247     \series@index\@MM
2248     \else
2249         \ifcase\series@index
2250         \or

```

Material before name 1:

```

2251         \let\@tempa\series@add@a
2252         \or

```

Material before name 2:

```

2253         \ifnum\series@total<\thr@@
2254             \let\@tempa\series@add@b
2255         \else
2256             \let\@tempa\series@add@c
2257         \fi
2258         \else

```

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

```

2259         \ifnum\series@index=\series@total
2260             \let\@tempa\series@add@d
2261         \else

```

```

2262             \ifnum\series@index<\series@total
2263             \let\@tempa\series@add@c
2264             \else
2265             \let\@tempa\@gobble
2266             \fi
2267         \fi
2268     \fi
2269 \fi
2270 \@tempa{#1}%
2271 }

```

\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.

```

2272 \def\SwapBreak#1{%
2273     \relax\ifvmode\leavevmode\fi
2274     \@tempcnta\@MM
2275     \toks@{#1}%

```

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

```
2276     \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.

```
2277     \ifnum \lastkern>\@ne \unkern \fi
```

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

```

2278     \ifnum\lastpenalty=\z@
2279     \else
2280         \@tempcnta\lastpenalty
2281         \unpenalty
2282     \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`.

```

2283 \edef\@tempa{%
2284 \nx\deferredquoteslogical
2285 \ifnum\lastkern=\@ne
2286 \else
2287 \ifnum\spacefactor=\sfcode\@xp\@xp\@xp\@xp\@car\string#1)\@nil
2288 \else
2289 \ifnum\spacefactor=\@nopunctsfcode
2290 \else
2291 \the\toks@
2292 \fi
2293 \fi
2294 \fi
2295 \nx\deferredquotes
2296 \ifnum\@tempcnta=\@MM \else \penalty\number\@tempcnta\space \fi
2297 \ifnum\lastkern=\@ne \ignorespaces \fi
2298 }%
2299 \@tempa
2300 }
```

`\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.

```

2301 \newcommand{\SingularPlural}[2]{#1}
2302 \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.)

```

2303 \BibSpec{nameLE}{
2304 +{}{}{given}
2305 +{}{\IfEmptyBibField{given}{ } }{surname}
2306 +{}{ }{jr}
2307 }
```

`\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.”

```

2308 \BibSpec{nameBE}{
2309 +{}{}{surname}
2310 +{}{ }{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.

```

2311 % +{}{ }{jr}
2312 }
```

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

```

2313 \BibSpec{nameinverted}{
2314     +{} {} {surname}
2315     +{,}{ } {given}
2316     +{,}{ } {jr}
2317 }

```

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.

```

2318 \newcommand{\PrintNames}{%
2319     \@ifstar{\PrintNames@a\set@othername}{\PrintNames@a\set@firstname}%
2320 }

```

`\PrintNames@a`

```

2321 \newcommand{\PrintNames@a}[4]{%
2322     \PrintSeries{name}
2323     {#1}
2324     {}{ and \set@othername}
2325     {,}{ \set@othername}
2326     {,}{ and \set@othername}
2327     {#2}{#4}{#3}%
2328 }

```

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

```

2329 \def\set@firstname#1{%
2330     \set@name{#1}\setbib@nameLE
2331 }

```

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

```

2332 \def\set@othername#1{%
2333     \set@name{#1}\setbib@nameLE
2334 }

```

`\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.

```

2335 \def\set@name#1{%
2336     \name@split#1,,,\@nil
2337     \set@name@a
2338 }

```


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

```

2339 \def\set@name@a#1{%
2340     \begingroup
2341         \get@current@properties
2342         \select@auxlanguage
2343         \def\@tempa{yes}%
2344         \ifx\@tempa\prop'inverted
2345             \setbib@nameBE
2346         \else
2347             #1%
2348         \fi
2349     \endgroup
2350 }
```

`\PrintPrimary`

```

2351 \def\PrintPrimary{%
2352     \ifx\current@primary\@empty
2353         \EmptyPrimaryWarning
2354     \else
2355         \print@primary\current@primary
2356     \fi
2357 }
```

`\EmptyPrimaryWarning`

```

2358 \def\EmptyPrimaryWarning{%
2359     \amsrefs@warning{No authors, editors or translators}%
2360 }
```

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

```

2361 \newcommand{\PrintAuthors}[1]{%
2362     \ifx\previous@primary\current@primary
2363         \sameauthors\@empty
2364     \else
2365         \def\current@bibfield{\bib'author}%
2366         \PrintNames{}{}{#1}%
2367     \fi
2368 }
```

`\sameauthors`

```

2369 \newcommand{\sameauthors}[1]{\bysame#1}
```

`\bysame`

```

2370 \def\bysame{%
2371     \leavevmode\hbox to3em{\hrulefill}\thinspace
2372     \kern\z@
2373 }
```

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

```

2374 \newcommand{\PrintNameList}{\PrintNames{}{}}
```

`\PrintEditorsC`

```

2375 \newcommand{\PrintEditorsC}[1]{%
2376     \PrintNames{Edited by }{}{#1}%
2377 }
```

`\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.

```

2378 \newcommand{\PrintEditorsA}[1]{%
2379     \ifx\previous@primary\current@primary
2380         \sameauthors{(ed\Plural{s}.)}%
2381     \else
2382         \def\current@bibfield{\bib'editor}%
2383         \PrintNames#{(ed\Plural{s}.)}{#1}%
2384     \fi
2385     \erase@field\bib'editor
2386 }
```

`\PrintEditorsB`

```

2387 \newcommand{\PrintEditorsB}{%
2388     \PrintNames*{(\SwapBreak{,}~ed\Plural{s}.)}%
2389 }
```

`\PrintContributions`

```

2390 \newcommand{\PrintContributions}[1]{%
2391     \PrintSeries
2392     {\fld@elt}
2393     {\print@contribution}
2394     {}{ and \print@contribution}
2395     {,}{ \print@contribution}
2396     {,}{ and \print@contribution}{#1}{}%
2397 }
```

`\print@contribution`

```

2398 \newcommand{\print@contribution}[1]{%
2399     \in@={#1}%
2400     \ifin@
2401         \ifnum\series@index=\@ne with \fi
2402         \RestrictedSetKeys{\bib}{%
2403             \bib@print@inner\setbib@contribution{\the\rsk@toks}%
2404             }{#1}%
2405     \else
2406         #1%
2407     \fi
2408 }
```

`\resolve@inner`

```

2409 \def\resolve@inner#1#2{%
2410     \in@={#2}%
2411     \ifin@
2412         \RestrictedSetKeys{\bib}{#1{\the\rsk@toks}}{#2}%
2413     \else
2414         \@ifundefined{bi@#2}{%
2415             \XRefWarning{#2}%
2416         }{%
2417             #1{\csname bi@#2\endcsname}%
2418         }%
2419     \fi
2420 }
```

`\PrintConference`

```

2421 \def\PrintConference{%
2422     \resolve@inner{\bib@print@inner\setbib@conference}
2423 }
```

```

\PrintConferenceDetails

2424 \def\PrintConferenceDetails#1{%
2425     \ifx\@empty\bib'address
2426         \ifx\@empty\bib'date
2427             \else
2428                 \PrintConferenceDetails@
2429             \fi
2430         \else
2431             \PrintConferenceDetails@
2432         \fi
2433 }

\PrintConferenceDetails@

2434 \def\PrintConferenceDetails@{%
2435     \ifnum\lastkern=\@ne\else\space\fi(\kern 1sp
2436     \ifx\@empty\bib'address
2437     \else
2438         \bib'address
2439     \fi
2440     \ifx\@empty\bib'date
2441     \else
2442         \SwapBreak{,}\space
2443         \print@date
2444     \fi
2445     )%\spacefactor\sfcode`\,%
2446 }

\PrintBook

2447 \def\PrintBook{%
2448     \resolve@inner{\bib@print@inner\setbib@innerbook}
2449 }

\PrintReprint

2450 \def\PrintReprint{%
2451     \resolve@inner{\bib@reprint}
2452 }

\bib@reprint

2453 \def\bib@reprint#1{%
2454     \begingroup
2455         #1\relax           % execute definitions locally
2456         \bib@resolve@xrefs
2457         \bib@field@patches
2458         \bib'setup
2459         \IfEmptyBibField{copula}{reprinted in}{\bib'copula} \nopunct
2460         \let\bib'language\@empty
2461         \setbib@book
2462     \endgroup
2463 }

\PrintTranslation

2464 \def\PrintTranslation{%
2465     \resolve@inner{\bib@translation}
2466 }

\bib@translation

2467 \def\bib@translation#1{%
2468     \begingroup

```

```

2469      #1\relax          % execute definitions locally
2470      \bib@resolve@xrefs
2471      \bib@field@patches
2472      \bib'setup
2473      \let\PrintPrimary\@empty
2474      \bib@append{;}{ % keep this space!
2475          \IfEmptyBibField{language}{English}{\bib'language} transl.%
2476          \IfEmptyBibField{pages}{ in \kern\@ne sp}{, }%
2477      }\bib'transition
2478      \let\bib'language\@empty
2479      \setbib@@
2480  \endgroup
2481 }

```

`\PrintTranslatorsC`

```

2482 \newcommand{\PrintTranslatorsC}[1]{%
2483   \PrintNames{translated by }{ }{#1}%
2484 }

```

`\PrintTranslatorsA`

```

2485 \newcommand{\PrintTranslatorsA}[1]{%
2486   \ifx\previous@primary\current@primary
2487     \sameauthors{(trans.)}%
2488   \else
2489     \def\current@bibfield{\bib'translator}%
2490     \PrintNames{}{(trans.)}{#1}%
2491   \fi
2492   \erase@field\bib'translator
2493 }

```

`\PrintTranslatorsB`

```

2494 \newcommand{\PrintTranslatorsB}[1]{
2495   \PrintNames*{ }{\SwapBreak{,}~tran\Plural{s}.)}%
2496 }

```

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

```

2497 \DefineName{alii}{\etaltext}
2498 \DefineName{others}{\etaltext}

```

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

```

2499 \newcommand{\etaltext}{et al.}
2500 \newcommand{\SubEtal}[1]{\etaltext}

```

6.23 The partial field

`\print@partial`

```

2501 \newcommand{\print@partial}{%
2502   \resolve@inner{\bib@print@inner\setbib@partial}
2503 }

```

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.

```

2504 \newcommand{\parenthesize}[1]{%
2505   \leavevmode\push@bracket\upn{ }{#1\pop@bracket
2506 }

```

`\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.)

```
2507 \providecommand{\upn}[1]{#1}
```

`\push@bracket`

`\pop@bracket`

```
2508 \let\bracket@stack\@empty
2509
2510 \def\push@bracket#1{%
2511     \xdef\bracket@stack{#1\bracket@stack}%
2512 }
2513
2514 \def\pop@bracket{%
2515     \iffalse{\fi
2516     \exp\pop@bracket@a\bracket@stack \@empty}%
2517 }
2518
2519 \def\pop@bracket@a#1{%
2520     \leavevmode/\upn{#1}%
2521     \xdef\bracket@stack{\iffalse}\fi
2522 }
```

`\bibquotes`

```
2523 \newcommand{\bibquotes}[1]{%
2524     \textquotedblleft#1%
2525     \gdef\deferredquotes{%
2526         \global\let\deferredquotes\@empty
2527         \textquotedblright
2528     }%
2529 }
```

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

`\ndash`

```
2530 \providecommand{\mdash}{\textemdash}
2531 \providecommand{\ndash}{\textendash}
```

`\strip@MRprefix`

```
2532 \def\strip@MRprefix#1#2#3#4\@nil{%
2533     \def\@tempa{#1#2#3#4}%
2534     \if#1M%
2535         \if#2R%
2536             \def\@tempa{#3#4}%
2537         \fi
2538     \fi
2539 }
```

`\MR`

```
2540 \def\MR#1{%
2541     \relax\ifhmode\unskip\spacefactor3000 \space\fi
2542     \begingroup
2543         \strip@MRprefix#1\@nil
2544         \edef\@tempa{MR\@nx\MRhref{\@tempa}{\@tempa}}%
2545     \exp\endgroup
2546     \@tempa
2547 }
```

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

```
2548 \providecommand{\MRhref}[2]{#1}
```

```

2549 \newcommand{\PrintReviews}[1]{%
2550     \PrintSeries{fld@elt}{},{ }{,{ }{ }}{#{1}}}%
2551 }
```

```

2552 \newcommand{\PrintPartials}[1]{%
2553     \PrintSeries
2554     {\fld@elt}
2555     {\print@partial}
2556     {;}{ \print@partial}
2557     {;}{ \print@partial}
2558     {;}{ \print@partial}{-}{#1}{-}%
2559 }

```

```

2560 \newcommand{\PrintISBNs}[1]{%
2561     \PrintSeries{\fld@elt}{-}{ }{ },{ }{-},{} }{ISBN }{#1}}{%
2562 }
```

```
2563 \newcommand{\voltext}{\IfEmptyBibField{series}{Vol.~}{vol.~}}
```

```
2564 \newcommand{\issuetext}{no.~}
```

```

2565 \newcommand{\DashPages}[1]{%
2566     p\pp@scan@a#1\ndash p\ndash{\pp@scan#1@-p@-{\}\@nil}\@nil.~#1%
2567 }
2568
2569 \def\pp@scan#1-#2@-#3#4\@nil{#3}
2570
2571 \def\pp@scan@a#1\ndash#2@\ndash#3#4\@nil{#3}

```

```

2572 \newcommand{\eprintpages}[1]{%
2573     #1\IfEmptyBibField{eprint}{-}\IfEmptyBibField{journal}{ pp.}{}}%
2574 }

```

```

2575 \def\PrintThesisType#1{%
2576     \thesis@type#1?\@nil{#1}%
2577 }
2578
2579 \def\thesis@type#1#2\@nil#3{%
2580     \ifx p#1%
2581         Ph.D. Thesis%
2582     \else
2583         \ifx m#1%

```

```

2584         Master's Thesis%
2585     \else
2586         #3%
2587     \fi
2588 \fi
2589 }

```

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

```

2590 \newcommand{\PrintDOI}[1]{%
2591     DOI #1%
2592     \IfEmptyBibField{volume}{, (to appear in print)}{}%
2593 }

```

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

```

2594 \newcommand{\PrintDatePV}[1]{%
2595     \IfEmptyBibField{doi}{%
2596         \let\@tempa\PrintDate
2597     }{%
2598         \IfEmptyBibField{volume}{%
2599             \let\@tempa\PrintDatePosted
2600         }{%
2601             \let\@tempa\PrintDate
2602         }%
2603     }%
2604     \@tempa{#1}%
2605 }

```

`\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.

```

2606 \newcommand{\PrintDate}[1]{(\print@date)}

```

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

```

2607 \newcommand{\PrintDateB}[1]{\print@date}

```

`\print@date`

```

2608 \def\print@date{%
2609     \ifx\bib@month\@empty
2610     \else
2611         \print@month@day
2612     \fi
2613     \bib@year
2614 }

```

`\print@month@day`

```

2615 \def\print@month@day{%
2616     \bib@monthname
2617     \ifx\@empty\bib@day \else \nobreakspace\number 0\bib@day,\fi
2618     \space
2619 }

```

`\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.

```

2620 \newcommand{\bib@monthname}{%
2621     \ifcase 0\bib@month
2622     \or January\or February\or March\or April\or May\or June\or
2623     July\or August\or September\or October\or November\or December\or
2624     Winter\or Spring\or Summer\or Fall\else Unknown Month%
2625     \fi
2626 }

```

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

```

2627 \newcommand{\PrintYear}[1]{\bib@year}

```

\PrintDatePosted This one is special for AMS use.

```

2628 \newcommand{\PrintDatePosted}[1]{\unskip, posted on \print@date}

```

\PrintEdition

```

2629 \newcommand{\PrintEdition}[1]{%
2630     \afterassignment\print@edition
2631     \count@ 0#1\relax\nil
2632 }

```

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

```

2633 \def\print@edition#1#2\nil{%
2634     \ifx\relax#1\relax
2635         \ifnum\count@>\z@
2636             \CardinalNumeric\count@
2637         \else
2638             ??th%
2639         \fi
2640     \ \editiontext
2641 \else
2642     \ifnum \count@>\z@ \number\count@ \fi
2643     #1#2\relax
2644 \fi
2645 }

```

\editiontext

```

2646 \newcommand{\editiontext}{ed.}

```

\CardinalNumber

```

2647 \newcommand{\CardinalNumeric}[1]{%
2648     \number#1\relax
2649     \if
2650         \ifnum#1<14
2651             \ifnum#1>\thr@@ T\else F\fi
2652         \else
2653             F%
2654         \fi
2655         T%
2656         th%
2657     \else
2658         \@xp\keep@last@digit\@xp#1\number#1\relax
2659         \ifcase#1th\or st\or nd\or rd\else th\fi
2660     \fi
2661 }

```


`\keep@last@digit`

```
2662 \def\keep@last@digit#1#2{%
2663     \ifx\relax#2%
2664         \@xp\@gobbletwo
2665     \else
2666         #1=#2\relax
2667     \fi
2668     \keep@last@digit#1%
2669 }
```

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

```
2670 \newcommand{\SentenceSpace}{\relax\ifhmode\spacefactor`\.\ \fi}
```

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

```
2671 \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.

```
2672 \let\bibliographystyle\@gobble
```

`\bibtex@style`

```
2673 \def\bibtex@style{amsrn}

2674 \AtBeginDocument{
2675     \if@files
2676         \immediate\write\@auxout{\string\bibstyle{\bibtex@style}}%
2677     \fi
2678 }
```

6.26 Implementing package options

6.26.1 The alphabetic option

```
2679 \IfOption{alphabetic}{%
2680     \def\bibtex@style{amsra}%
2681     \def\alpha@label{%
2682         \ifx\@empty\bib'label
2683             \def\thebib{\CurrentBib}%
2684         \else
2685             \let\thebib\bib'label
2686         \fi
2687     }%
2688     \let\generate@label\generate@alphalabel
2689     \let\process@citelist\process@citelist@unsorted
2690     \def\numeric@refs{01}%
2691 }
```

6.26.2 The shortalphabetic option

```
2692 \IfOption{shortalphabetic}{%
2693     \def\bibtex@style{amsrs}%
2694     \def\alpha@label{%
2695         \ifx\@empty\bib'label
2696             \def\thebib{\CurrentBib}%
2697         \fi
2698     }
```

```

2697         \else
2698             \let\thebib\bib'label
2699         \fi
2700     }%
2701     \let\@suffix@format\@arabic
2702     \def\calc@author@part{%
2703         \@xp\@multiauthorlabel\@xp{\@tempa}%
2704     }%
2705     \let\append@label@year\@empty
2706     \let\generate@label\generate@alphalabel
2707     \let\process@citelist\process@citelist@unsorted
2708     \def\numeric@refs{01}%
2709 }{}

```

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`.

```

2710 \AtBeginDocument{%
2711     \@ifpackageloaded{backref}{%
2712         \let\PrintBackRefs\print@backrefs
2713         \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.)

```

2714     \begingroup
2715         \makeatletter
2716         \input{\jobname.brf}%
2717         \if@filesw
2718             \newwrite\tf@brf
2719             \immediate\openout\tf@brf \jobname.brf\relax
2720         \fi
2721     \endgroup
2722 }{}%
2723 }

```

6.26.4 The citation-order option

```

2724 \IfOption{citation-order}{%
2725     \IfOption{alphabetic}{%
2726         \amsrefs@warning@nl{%
2727             The citation-order and alphabetic options are
2728             incompatible%
2729         }%
2730     }{
2731         \def\bibtex@style{amsru}%
2732     }
2733 }{}

```

6.26.5 The initials option

```

2734 \IfOption{initials}{% TRUE:
2735     \BibSpec{nameLE}{
2736         +{}{}{initials}
2737         +{}{\IfEmptyBibField{initials}{}{ }}{surname}
2738         +{}{ }{jr}
2739     }
2740

```

```

2741 \BibSpec{nameBE}{
2742     +{}{}{surname}
2743     +{}{ }{initials}
2744     %   +{}{ }{jr}
2745 }
2746
2747 \BibSpec{nameinverted}{
2748     +{} {} {surname}
2749     +{,}{ } {initials}
2750     +{,}{ } {jr}
2751 }
2752 }{% initials? FALSE:
2753 %   \let\extract@initials\@gobble
2754 } % end conditional code for initials option

```

6.26.6 The logical-quotes option

\deferredquotes

```

2755 \let\deferredquotes\@empty

```

\deferredquoteslogical

```

2756 \IfOption{logical-quotes}{%
2757     \def\deferredquoteslogical{\deferredquotes}%
2758 }{%
2759     \let\deferredquoteslogical\relax
2760 }

```

6.26.7 The non-compressed-cites option

```

2761 \IfOption{non-compressed-cites}{%
2762     \let\cite@compress\cite@print
2763 }{}

```

6.26.8 The non-sorted-cites option

```

2764 \IfOption{non-sorted-cites}{%
2765     \let\process@citelist\process@citelist@unsorted
2766 }{}

```

6.26.9 The short-journals option

```

2767 \IfOption{short-journals}{%
2768     \renewcommand{\DefineJournal}[4]{%
2769         \bib*{#1}{periodical}{
2770             issn={#2},
2771             journal={#3},
2772         }%
2773     }
2774 }{}

```

6.26.10 The short-publishers option

```

2775 \IfOption{short-publishers}{%
2776     \renewcommand{\DefinePublisher}[4]{%
2777         \bib*{#1}{publisher}{%
2778             publisher={#2},%

```

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

```

2779         address={#4},
2780     }%
2781 }%
2782 }{}

```

6.26.11 The short-months option

```

2783 \IfOption{short-months}{%
2784   \renewcommand{\bib@monthname}{%
2785     \ifcase 0\bib@month
2786     \or Jan.\or Feb.\or Mar.\or Apr.\or May\or June\or
2787     July\or Aug.\or Sep.\or Oct.\or Nov.\or Dec.\or
2788     Winter\or Spring\or Summer\or Fall\else Unknown Month%
2789   \fi
2790   }%
2791 }{}

```

6.26.12 The y2k option

```

2792 \IfOption{y2k}{%
2793   \IfOption{alphabetic}{%
2794     \def\year@short#1\@nil{#1}%
2795     \def\bibtex@style{amsry}%
2796   }{%
2797     \amsrefs@warning@nl{%
2798       The y2k option can only be used with the~J%
2799       alphabetic option%
2800     }%
2801   }
2802 }{}

```

6.26.13 The bibtex-style option

```

2803 \IfOption{bibtex-style}{%
2804   \RequirePackage{amsbst}
2805 }{}

```

6.26.14 The msc-links option

```

2806 \IfOption{msc-links}{%

```

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

```

2807   \@ifundefined{href}{}%
2808     \def\parse@MR#1 (#2)#3\@nil{%
2809       \def\MR@url{http://www.ams.org/mathscinet-getitem?mr=#1}%
2810       \def\@tempd{#1}%
2811       \def\@tempe{#2}%
2812     }%
2813     \def\MRhref#1#2{%
2814       \begingroup
2815         \parse@MR#1 ()\@empty\@nil%
2816         \href{\MR@url}{\@tempd\vphantom{()}}%
2817         \ifx\@tempe\@empty
2818           \else
2819             \ \href{\MR@url}{(\@tempe)}%
2820           \fi
2821       \endgroup
2822     }%
2823   }{}
2824 }{}

```

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.

```

2825 \IfOption{author-year}{}\PopCatcodes \endinput

```

```

\generate@label

```

```

2826 \def\generate@label{%

```

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

```
2827 \ifx\bib'label\@empty
2828 \begingroup
```

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

```
2829 \global\let\previous@stem\current@stem
2830 \global\let\current@stem\@empty
2831 \global\let\previous@year\current@year
2832 \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.

```
2833 \@apply\auto@protect\amsrefs@textsymbols
2834 \@apply\auto@protect\amsrefs@textaccents
2835 \def\name##1{\@nx\name{\lscan@a##1,\@nil}}%
2836 \auto@protect\etaltext
2837 \normalize@edef\current@stem{\current@primary}%
2838 \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.

```
2839 \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.)

```
2840 \edef\@tempa{%
2841 \def\@nx\cite@label{\current@stem}%
2842 \def\@nx\bib@label@year{%
2843 \current@year
2844 \alpha@label@suffix
2845 }%
2846 }
2847 \@xp\endgroup
2848 \@tempa
2849 \fi
2850 }
```

```
\lscan@a
```

```
2851 \def\lscan@a#1,#2\@nil{#1}
```

```
\citesel@author
```

```
2852 \def\citesel@author#1#2#3#4#5{\PrintCiteNames{#3}}
```

```
\citesel@authoryear
```

```
2853 \def\citesel@authoryear#1#2#3#4#5{\PrintCNY{#3}{#4}}
```

```
\citesel@object
```

```
2854 \def\citesel@object#1#2#3#4#5{\PrintCiteNames{#3} \citeleft#4}
```

```
\citesel
```

```
2855 \let\citesel\citesel@authoryear
```

```

\numeric@refs
2856 \def\numeric@refs{01}%

\citeleft
2857 \def\citeleft{()%

\citeright
2858 \def\citeright{}}%

\@citeleft
2859 \def\@citeleft{\ifx\citesel\citesel@object\else\citeleft\fi}%

\citepunct
2860 \def\citepunct{; }

\BibLabel
2861 \def\BibLabel{%
2862   \Hy@raisedlink{\hyper@anchorstart{cite.\CurrentBib}\relax\hyper@anchorend}%
2863 }

\process@citelist
2864 \let\process@citelist\process@citelist@unsorted

\ycite
2865 \DeclareRobustCommand{\ycite}[1]{%
2866   \star@{\cite@a\citesel@year{#1}}{}}%
2867 }

\ycites
2868 \DeclareRobustCommand{\ycites}[1]{%
2869   \begingroup
2870     \def\citepunct{, }%
2871     \let\citesel\citesel@year
2872     \cites{#1}%
2873   \endgroup
2874 }

\ocite
2875 \DeclareRobustCommand{\ocite}[1]{%
2876   \star@{\cite@a\citesel@object{#1}}{}}%
2877 }

\ocites
2878 \DeclareRobustCommand{\ocites}[1]{%
2879   \begingroup
2880     \let\@citelist\@ocitelist
2881     \cites{#1}%
2882   \endgroup
2883 }

\ocitelist
2884 \def\@ocitelist#1{%
2885   \PrintSeries{InnerCite}%
2886   {\ocite}%
2887   {}{ 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.

```
2888      {,}{ \ocite}%
2889      {,}{ and \ocite}%
2890      {}%
2891      {#1}%
2892      {}%
2893 }
```

`\citeauthor`

```
2894 \DeclareRobustCommand{\citeauthor}[1]{%
2895   \star@{\cite@a\citesel@author{#1}}{}}%
2896 }
```

`\citeauthory`

```
2897 \DeclareRobustCommand{\citeauthory}[1]{%
2898   \citeauthor{#1} \ycite{#1}%
2899 }
```

`\fullcite`

```
2900 \DeclareRobustCommand{\fullcite}[1]{%
2901   \begingroup
2902     \let\print@citenames\CiteNamesFull
2903     \star@{\cite@a\citesel@authoryear{#1}}{}}%
2904   \endgroup
2905 }
```

`\fullocite`

```
2906 \DeclareRobustCommand{\fullocite}[1]{%
2907   \begingroup
2908     \let\print@citenames\CiteNamesFull
2909     \star@{\cite@a\citesel@object{#1}}{}}%
2910   \endgroup
2911 }
```

Invert the first author’s name.

```
2912 \def\set@firstname#1{%
2913   \set@name{#1}\setbib@nameinverted
2914 }
```

`\PrintCNY`

```
2915 \def\PrintCNY#1#2{%
2916   \PrintCiteNames{#1}%
2917   \@ifnotempty{#2}{\@addpunct{,} #2}%
2918 }
```

`\PrintCiteNames`

```
2919 \def\PrintCiteNames#1{%
2920   \leavevmode
2921   \def\@tempa{#1}%
2922   \ifx\@tempa\prev@names
2923   \else
2924     \gdef\prev@names{#1}%
2925     \@xp\ifx\@car#1.\@nil\CitePrintUndefined
2926     #1\relax
2927   \else
2928     \print@citenames{#1}%
2929   \fi}
```

```

2929      \fi
2930      \fi
2931 }

```

`\CiteNames`

```

2932 \newcommand{\CiteNames}[1]{%
2933     \PrintSeries{name}%
2934     {}%
2935     {}{ 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.

```

2936     {}{\@gobble}%
2937     {}{ \etaltext\@gobble}%
2938     {}%
2939     {#1}%
2940     {}%
2941 }

```

`\print@citenames`

```

2942 \let\print@citenames\CiteNames

```

`\CiteNamesFull`

```

2943 \newcommand{\CiteNamesFull}[1]{%
2944     \PrintSeries{name}%
2945     {}%
2946     {}{ 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.

```

2947     {,}{ }%
2948     {,}{ and }%
2949     {}%
2950     {#1}%
2951     {}%
2952 }

```

`\PrintDate` No parentheses around the year.

```

2953 \renewcommand{\PrintDate}[1]{\bib@label@year}

```

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

```

2954 \def\print@date{%
2955     \IfEmptyBibField{date}{%
2956         \IfEmptyBibField{year}{\BibField{status}}{\bib@year}%
2957     }{%
2958         \bib@year
2959     }%
2960 }

```

```

2961 \BibSpec{article}{%
2962     +{ } {\PrintAuthors}          {author}
2963     +{.} { \PrintDate}             {date}
2964     +{.} { \textit}                {title}
2965     +{.} { }                       {part}
2966     +{:} { \textit}                {subtitle}
2967     +{,} { \PrintContributions}    {contribution}
2968     +{.} { \PrintPartials}         {partial}

```



```

2969 +{,} { } {journal}
2970 +{ } { \textbf} {volume}
2971 +{,} { \issuetext} {number}
2972 +{,} { \eprintpages} {pages}
2973 +{,} { } {status}
2974 +{,} { \PrintDOI} {doi}
2975 +{,} { available at \eprint} {eprint}
2976 +{ } { \parenthesize} {language}
2977 +{ } { \PrintTranslation} {translation}
2978 +{;} { \PrintReprint} {reprint}
2979 +{.} { } {note}
2980 +{.} {} {transition}
2981 +{ } {\SentenceSpace \PrintReviews} {review}
2982 }
2983
2984 \BibSpec{book}{%
2985 +{ } {\PrintPrimary} {transition}
2986 +{.} { \PrintDate} {date}
2987 +{.} { \textit} {title}
2988 +{.} { } {part}
2989 +{:} { \textit} {subtitle}
2990 +{,} { \PrintEdition} {edition}
2991 +{ } { \PrintEditorsB} {editor}
2992 +{,} { \PrintTranslatorsC} {translator}
2993 +{,} { \PrintContributions} {contribution}
2994 +{,} { } {series}
2995 +{,} { \voltext} {volume}
2996 +{,} { } {publisher}
2997 +{,} { } {organization}
2998 +{,} { } {address}
2999 +{,} { } {status}
3000 +{ } { \parenthesize} {language}
3001 +{ } { \PrintTranslation} {translation}
3002 +{;} { \PrintReprint} {reprint}
3003 +{.} { } {note}
3004 +{.} {} {transition}
3005 +{ } {\SentenceSpace \PrintReviews} {review}
3006 }
3007
3008 \BibSpec{collection.article}{%
3009 +{ } {\PrintAuthors} {author}
3010 +{.} { \PrintDate} {date}
3011 +{.} { \textit} {title}
3012 +{.} { } {part}
3013 +{:} { \textit} {subtitle}
3014 +{,} { \PrintContributions} {contribution}
3015 +{,} { \PrintConference} {conference}
3016 +{ } {\PrintBook} {book}
3017 +{,} { } {booktitle}
3018 +{,} { pp.~} {pages}
3019 +{,} { } {status}
3020 +{,} { \PrintDOI} {doi}
3021 +{,} { available at \eprint} {eprint}
3022 +{ } { \parenthesize} {language}
3023 +{ } { \PrintTranslation} {translation}
3024 +{;} { \PrintReprint} {reprint}
3025 +{.} { } {note}
3026 +{.} {} {transition}
3027 +{ } {\SentenceSpace \PrintReviews} {review}

```

```

3028 }
3029
3030 \BibSpec{report}{%
3031     +{} {\PrintPrimary}           {transition}
3032     +{.} { \PrintDate}             {date}
3033     +{.} { \textit}                {title}
3034     +{.} { }                       {part}
3035     +{:} { \textit}                {subtitle}
3036     +{,} { \PrintEdition}           {edition}
3037     +{,} { \PrintContributions}     {contribution}
3038     +{,} { Technical Report }      {number}
3039     +{,} { }                       {series}
3040     +{,} { }                       {organization}
3041     +{,} { }                       {address}
3042     +{,} { \eprint}                {eprint}
3043     +{,} { }                       {status}
3044     +{} { \parenthesize}           {language}
3045     +{} { \PrintTranslation}        {translation}
3046     +{;} { \PrintReprint}           {reprint}
3047     +{.} { }                       {note}
3048     +{.} {}                        {transition}
3049     +{} {\SentenceSpace \PrintReviews} {review}
3050 }
3051
3052 \BibSpec{thesis}{%
3053     +{} {\PrintAuthors}            {author}
3054     +{.} { \PrintDate}             {date}
3055     +{.} { \textit}                {title}
3056     +{:} { \textit}                {subtitle}
3057     +{,} { \PrintThesisType}        {type}
3058     +{,} { }                       {organization}
3059     +{,} { }                       {address}
3060     +{,} { \eprint}                {eprint}
3061     +{,} { }                       {status}
3062     +{} { \parenthesize}           {language}
3063     +{} { \PrintTranslation}        {translation}
3064     +{;} { \PrintReprint}           {reprint}
3065     +{.} { }                       {note}
3066     +{.} {}                        {transition}
3067     +{} {\SentenceSpace \PrintReviews} {review}
3068 }
3069 \PopCatcodes
3070 \</pkg>

```

6.27 The amsbst package

```

3071 \<{*bst}

3072 \NeedsTeXFormat{LaTeX2e}[1995/12/01]
3073 \ProvidesPackage{amsbst}[2012/05/08 v2.10]
3074 %\RequirePackage{amsrefs}[2004/03/29]

3075 \BibSpec{article}{%
3076     +{} {\PrintAuthors}            {author}
3077     +{.} { }                       {title}
3078     +{.} { }                       {part}
3079     +{:} { }                       {subtitle}
3080     +{.} { \PrintContributions}     {contribution}
3081     +{.} { \PrintPartials}          {partial}
3082     +{.} { \emph}                  {journal}
3083     +{} { }                       {volume}

```

```

3084   +{} { \parenthesize}           {number}
3085   +{:} {}                         {pages}
3086   +{,} { \PrintDateB}             {date}
3087   +{,} { }                        {status}
3088   +{.} { \PrintTranslation}        {translation}
3089   +{.} { Reprinted in \PrintReprint} {reprint}
3090   +{.} { }                        {note}
3091   +{.} {}                         {transition}
3092 }
3093
3094 \BibSpec{partial}{%
3095   +{} {}                         {part}
3096   +{:} { }                       {subtitle}
3097   +{.} { \PrintContributions}     {contribution}
3098   +{.} { \emph}                  {journal}
3099   +{} { }                        {volume}
3100   +{} { \parenthesize}           {number}
3101   +{:} {}                         {pages}
3102   +{,} { \PrintDateB}            {date}
3103 }
3104
3105 \BibSpec{book}{%
3106   +{} { \PrintPrimary}            {transition}
3107   +{.} { \emph}                  {title}
3108   +{.} { }                       {part}
3109   +{:} { \emph}                  {subtitle}
3110   +{.} { }                       {series}
3111   +{,} { \voltext}               {volume}
3112   +{.} { Edited by \PrintNameList} {editor}
3113   +{.} { Translated by \PrintNameList} {translator}
3114   +{.} { \PrintContributions}     {contribution}
3115   +{.} { }                       {publisher}
3116   +{.} { }                       {organization}
3117   +{,} { }                       {address}
3118   +{,} { \PrintEdition}          {edition}
3119   +{,} { \PrintDateB}            {date}
3120   +{.} { }                       {note}
3121   +{.} {}                        {transition}
3122   +{.} { \PrintTranslation}        {translation}
3123   +{.} { Reprinted in \PrintReprint} {reprint}
3124   +{.} {}                        {transition}
3125 }
3126
3127 \BibSpec{collection.article}{%
3128   +{} { \PrintAuthors}           {author}
3129   +{.} { }                       {title}
3130   +{.} { }                       {part}
3131   +{:} { }                       {subtitle}
3132   +{.} { \PrintContributions}     {contribution}
3133   +{.} { \PrintConference}        {conference}
3134   +{.} { \PrintBook}              {book}
3135   +{.} { In }                    {booktitle}
3136   +{,} { pages~}                 {pages}
3137   +{.} { \PrintDateB}             {date}
3138   +{.} { \PrintTranslation}        {translation}
3139   +{.} { Reprinted in \PrintReprint} {reprint}
3140   +{.} { }                       {note}
3141   +{.} {}                        {transition}
3142 }

```

```

3143
3144 \BibSpec{conference}{%
3145     +{} {} {title}
3146     +{} {\PrintConferenceDetails} {transition}
3147 }
3148
3149 \BibSpec{innerbook}{%
3150     +{.} { \emph} {title}
3151     +{.} { } {part}
3152     +{:} { \emph} {subtitle}
3153     +{.} { } {series}
3154     +{,} { \voltext} {volume}
3155     +{.} { Edited by \PrintNameList} {editor}
3156     +{.} { Translated by \PrintNameList} {translator}
3157     +{.} { \PrintContributions} {contribution}
3158     +{.} { } {publisher}
3159     +{.} { } {organization}
3160     +{,} { } {address}
3161     +{,} { \PrintEdition} {edition}
3162     +{,} { \PrintDateB} {date}
3163     +{.} { } {note}
3164     +{.} {} {transition}
3165 }
3166
3167 \BibSpec{report}{%
3168     +{} {\PrintPrimary} {transition}
3169     +{.} { \emph} {title}
3170     +{.} { } {part}
3171     +{:} { \emph} {subtitle}
3172     +{.} { \PrintContributions} {contribution}
3173     +{.} { Technical Report } {number}
3174     +{,} { } {series}
3175     +{.} { } {organization}
3176     +{,} { } {address}
3177     +{,} { \PrintDateB} {date}
3178     +{.} { \PrintTranslation} {translation}
3179     +{.} { Reprinted in \PrintReprint} {reprint}
3180     +{.} { } {note}
3181     +{.} {} {transition}
3182 }
3183
3184 \BibSpec{thesis}{%
3185     +{} {\PrintAuthors} {author}
3186     +{,} { \emph} {title}
3187     +{:} { \emph} {subtitle}
3188     +{.} { \PrintThesisType} {type}
3189     +{.} { } {organization}
3190     +{,} { } {address}
3191     +{,} { \PrintDateB} {date}
3192     +{.} { \PrintTranslation} {translation}
3193     +{.} { Reprinted in \PrintReprint} {reprint}
3194     +{.} { } {note}
3195     +{.} {} {transition}
3196 }

```

`\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.

```

3197 \renewcommand{\PrintEditorsA}[1]{%

```

```

3198   \def\current@bibfield{\bib'editor}%
3199   \PrintNames{},{, editor\Plural{s}}{#1}%
3200   \erase@field\bib'editor
3201 }

```

```
\PrintTranslatorsA
```

```

3202 \renewcommand{\PrintTranslatorsA}[1]{%
3203   \def\current@bibfield{\bib'translator}%
3204   \PrintNames{},{, translator\Plural{s}}{#1}%
3205   \erase@field\bib'translator
3206 }

```

```
3207 </bst>
```

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

```
3208 \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		
\ <i>57, 1692</i>	\@apply <i>68</i>
& <i>25</i>	\@arabic <i>2701</i>
' <i>40, 57, 1693</i>	\@bibdef <i>26, 26, 579,</i>
+ <i>24</i>	<i>589, 594, 606,</i>
, <i>24, 119, 2445</i>	<i>1166, 1169, 1185</i>
- <i>24</i>	\@biblist <i>486, 491, 499</i>
.	.. <i>24, 114, 1694, 2670</i>	\@car .. <i>2109, 2287, 2925</i>
.aux	<i>8, 8, 8, 8, 9, 9,</i>	\@carcube <i>2110</i>
	<i>9, 9, 9, 9, 9, 9,</i>	\@chomp ... <i>85, 1887,</i>
	<i>9, 9, 9, 9, 9, 10,</i>	<i>1899, 1952, 1953</i>
	<i>10, 10, 10, 10,</i>	\@chomper <i>13, 13, 88, 102</i>
	<i>10, 10, 43, 52,</i>	\@citeleft <i>1417, 1425, 2859</i>
	<i>54, 54, 54, 54, 55</i>	\@citelist <i>46,</i>
.bbl	. <i>8, 8, 9, 25, 27, 41</i>	<i>1340, 1341, 2880</i>
.bib <i>8</i>	\@citestyle <i>1344, 1361</i>
.brf <i>90</i>	\@citex <i>47</i>
.dtx <i>8</i>	\@close@bbl@file <i>1182,</i>
.ltb <i>25, 27, 40</i>	<i>1254, 1263, 1273</i>
:	... <i>24, 117, 628, 644</i>	\@concat <i>59, 2077</i>
; <i>24, 118</i>	\@count@chars <i>2061, 2066</i>
= <i>24, 1695</i>	\@currentlabel <i>1011, 1053</i>
? option <i>16</i> <i>1397, 1570,</i>
\@undefined <i>1390</i>	<i>1576, 1592, 1644</i>
\@MM	.. <i>2247, 2274, 2296</i>	\@currentx <i>135</i>
\@addpunct	.. <i>6, 14,</i>	\@currname <i>137</i>
	<i>14, 107, 1353, 2917</i>	\@empty <i>13,</i>
\@alph <i>2172</i>	<i>18, 18, 19, 42, 42</i>
		<i>1419, 2114, 2917</i>
		\@endbiblist <i>488, 493, 518</i>
		\@extract@initial <i>70,</i>
		<i>2027, 2029, 2041</i>
		\@filef@und . <i>1199,</i>
		<i>1202, 1205, 1219</i>
		\@firstofone <i>15, 36,</i>
		<i>987, 1317, 1339,</i>
		<i>1382, 1670, 1676</i>
		\@firstoftwo <i>48, 1001, 2230</i>
		\@firstone . <i>2109, 2111</i>
		\@firstthree <i>2110, 2133</i>
		\@for <i>33</i>
		\@gobblefour <i>696</i>
		\@gobblethree <i>1435</i>
		\@ifclassloaded <i>1110, 1111, 1112</i>
		\@ifempty <i>45,</i>
		<i>267, 1069,</i>
		<i>1286, 1311,</i>
		<i>1397, 1570,</i>
		<i>1576, 1592, 1644</i>
		\@ifnextchar <i>18</i>
		\@ifnotempty <i>45, 238,</i>
		<i>661, 685, 1298,</i>
		<i>1419, 2114, 2917</i>

- \@ifpackageloaded 1669,
1672, 1678, 2711
- \@ifpackagewith 5, 8, 1679
- \@ifsame@patterns 990, 996
- \@ifsamepat 997, 999
- \@input@ 1186
- \@lappend 61, 1690, 1719,
1720, 1722, 1723
- \@latex@warning 520, 1405
- \@let@token 64, 70
- \@listctr 507
- \@m 110, 512, 552
- \@marknames 2113, 2125
- \@maxlabelwidth 505,
524, 527, 528, 536
- \@mkboth 39, 1114
- \@multiauthorlabel 2106, 2143, 2703
- \@nametoken 2111,
2119, 2131, 2133
- \@nameuse 60
- \@nil 13, 36,
61, 62, 67, 67, 68
- \@nilgobble 63, 82, 1829, 2124
- \@nmbrlisttrue 506
- \@noitemerr 520
- \@nopunctsfcodes 122, 123, 2289
- \@ocitelist 2880, 2884
- \@oneauthorlabel 2104, 2128
- \@open@bbl@file 619,
1179, 1253, 1269
- \@pass@options 134
- \@prev@cite 1547
- \@secondoftwo 48, 1003, 2232
- \@sptoken 1841,
1916, 1966, 2014
- \@starttoc 90, 90, 90
- \@suffix@format 2172,
2183, 2194, 2701
- \@tabacckludge 59,
1749, 1761, 1780
- \@tempa 63
- \@tempcnta 13, 53
- \@tempcntb 53
- \@tempd 2810, 2816
- \@tempdima 526, 527, 528
- \@tempe 2811, 2817, 2819
- \@tempswafalse 582, 1044, 2174
- \@tempwattrue 578,
1062, 1065, 2178
- \@temptokena 63
- \@temptokenb 786, 792, 825
- \@testdef 39
- \@threeauthors 2139, 2153
- \@uclclist 58, 58
- \@urlbreak 25, 27
- \@urlfix 24, 32
- \@urlsetup 20, 22
- \@xifempty 45, 47
- \^ 106, 1696
- _ 1697
- \~ 29, 1698
- _ 1338, 2640, 2819
- A
- \A 550
- abbrv Bib_T_EX style 15
- \add@toks@ 60
- address field 3, 3, 3, 40, 40, 91
- \addto@defer@list 707, 711
- \addto@hook 60, 825
- \AE 1728
- \ae 58, 58, 1728
- \after@deleting@token 41, 1848,
1858, 1926,
1964, 1980, 2023
- \afterassignment 12,
42, 82, 718, 2630
- \aftergroup 19, 290
- alpha Bib_T_EX style 14
- \alpha@label 1012,
1015, 2681, 2694
- \alpha@label@suffix 2095,
2168, 2183,
2190, 2194, 2844
- \alpha@suffix 2170, 2171,
2182, 2183,
2184, 2191, 2194
- alphabetic option 14, 72, 74, 89
- amsart class 90
- amsbook class 45
- amsbst package 98
- amsgen package 14
- amsproc class 45
- 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, 40, 54,
78, 90, 92, 101, 101
- \amsrefs@bibcite 1615, 1620, 1624
- \amsrefs@error 105, 257, 307, 599
- \amsrefs@textaccents 58,
855, 1043, 1688,
1690, 1748,
1762, 1779,
1880, 2084, 2834
- \amsrefs@textsymbols 60, 854, 1042,
1717, 1719,
1720, 1722,
1723, 1747,
1763, 1777,
1879, 2083, 2833
- \amsrefs@warning 104, 898,
899, 1076, 1090,
1190, 1216,
1294, 1299,
1304, 1487,
1873, 1945, 2359
- \amsrefs@warning@nl 4, 6, 9, 157,
167, 2726, 2797
- amxport Bib_T_EX style 71, 72
- \append@label@year 2090, 2158, 2705
- \append@to@stem 2077, 2135,
2141, 2151, 2161
- \ar@hyperlink 1418, 1664, 1673
- \ar@SK@cite 1416,
1667, 1680, 1683
- article entry type 4, 7
- \AtBeginDocument 90,
594, 595, 1616,
1668, 2674, 2710
- \AtEndDocument 1106
- author field 5, 5
- author-year option 46,
74, 80, 80, 92, 92
- \AuthorList 7
- \auto@protect 12, 12,
12, 57, 60, 854,
855, 1024, 1041,
1042, 1043,
1747, 1762,
1770, 1777,
1876, 1877,

- 1879, 1880,
2083, 2084,
2085, 2086,
2833, 2834, 2836
- B**
- \b 1699
\b@ 37, 47, 48,
48, 51, 54, 55, 55
\b@* 1477
\b@aaa 51
\b@bbb 51
\b@moo 8, 9, 43
\b@whatever .. 9, 10, 48
\b@xyz 10
\b@zzz 51
babel package 36
\back@cite . 1652, 2713
\BackCite ... 1370,
1449, 1651, 2713
backref package ...
..... 90, 90, 90
backrefs option
..... 9, 90, 90, 90
\bbl@copy 624, 678
\bbl@copy@a .. 682, 684
\bbl@out 1251,
1261, 1262,
1264, 1272, 1275
\bbl@write
621, 625, 688,
1252, 1264, 1274
\bblname 1173, 1195, 1262
\begin 9, 9, 9
\bgroup
89, 1798, 1866,
1929, 1983, 2026
\bi@ 29
\bib .. 3, 3, 3, 5, 7,
7, 7, 10, 10, 10,
10, 25, 36, 40,
40, 42, 42, 42,
42, 43, 47, 60, 575
\bib' 33
\bib'bar 19
\bib'date 35
\bib'given 62, 62
\bib'initials 62
\bib'jr 62
\bib'language 35
\bib'pages 18
\bib'surname 62
\bib* 25, 27
\bib@append
... 18, 19, 19,
19, 245, 258, 2474
\bib@author 37
\bib@bar 19
\bib@cite 37, 731, 1013
\bib@cite@a 1018, 1021
\bib@cite@check ...
..... 1033, 1036
\bib@citeyear 2160, 2164
\bib@day
958, 965, 966, 2617
\bib@dbfile
.. 1196, 1199,
1202, 1205,
1224, 1227, 1230
\bib@div@mark
. 1109, 1124, 1133
\bib@end .. 31, 748, 765
\bib@exec .. 11, 26,
29, 595, 603, 608
\bib@field@patches
..... 727, 754,
909, 2457, 2471
\bib@label 72, 72, 93, 93
\bib@label@year ...
.. 1027, 1060,
1074, 1081,
1088, 2842, 2953
\bib@language
. 35, 979, 981,
984, 986, 987,
990, 991, 992, 1006
\bib@language@fixup
..... 954, 976
\bib@mklab ... 508, 525
\bib@month
957, 964, 967,
2609, 2621, 2785
\bib@monthname
. 2616, 2620, 2784
\bib@parsedate
.... 936, 939, 959
\bib@parsedate@a ..
..... 960, 962
\bib@print
.. 26, 30, 595, 722
\bib@print@inner ..
... 750, 2403,
2422, 2448, 2502
\bib@reprint 2451, 2453
\bib@reset 859
\bib@resolve@xrefs
.... 726, 753,
773, 2456, 2470
\bib@selectlanguage
..... 728, 989
\bib@start
.. 72, 93, 723, 761
\bib@store 26, 608, 717
\bib@translation ..
..... 2465, 2467
\bib@year . 37, 934,
943, 956, 963,
968, 2159, 2162,
2613, 2627,
2832, 2956, 2958
\BibAbbrevWarning .
.... 807, 842, 898
bibchapter environ-
ment 1119
\bibchapter 1140
\bibcite . 9, 9, 9, 9,
9, 9, 9, 10, 10,
10, 10, 39, 54,
54, 54, 1104,
1106, 1620, 1624
\bibcite@a . 1615, 1626
\bibcite@b . 1631, 1635
\bibcite@write
..... 1019, 1094
\bibdata 8, 9
\bibdeferlist 710,
714, 1183, 1184
bibdiv environment 1137
\bibdiv 1143
\BibField 7, 7, 184, 2956
\BibItem . 580, 583, 586
\bibitem 9, 9, 9
\BibLabel
503, 560, 566, 2861
\biblanguagedefault
..... 979, 1006
\biblanguagEnglish
..... 1006
\bibliography .. 8, 8, 9
\bibliographystyle
..... 8, 89, 2672
biblist environment
... 1, 23, 23, 484
\biblist 90, 1144
biblist* environment 490
\biblistfont . 495, 501
\bibName 39, 1107, 1119
\bibquotes 3, 6, 6, 6, 2523
bibsection environ-
ment 1128
\bibsection 1138
\BibSelect
. 1167, 1170, 1173
\bibselect
25, 28, 29, 49,
1164, 1191, 1192
\bibselect* ... 27, 49
\bibselect@msg ... 1176
\BibSpec ... 3, 4, 4,
5, 5, 5, 6, 7, 7,
7, 8, 18, 18, 18, 236
\bibspec@scan
18, 18, 18, 239, 244
\bibspec@scan@error
..... 253, 257
\BibSpecAlias .. 20,
304, 460, 461,

- 462, 463, 464,
465, 466, 467, 468
\bibstyle ... 8, 9, 2676
bibtex-style option 92
\bibtexstyle 2673,
2676, 2680,
2693, 2731, 2795
book class 40
\bracket@stack 2508,
2511, 2516, 2521
bst 3
\bud 1742
\bysame 7, 7, 7, 2369, 2370
- C**
- \c** 1700
\c@bib@env 536, 541, 544
\calc@alpha@suffix
. 2091, 2173, 2839
\calc@author@part .
. 2089, 2102, 2702
\CardinalNumber .. 2647
\CardinalNumeric ..
. 8, 8, 2636, 2647
\catcode 25
\cdprime 1741
\ch@ck 1266
\chapter* 39
\check@mathfonts .. 23
\chomp@ 95, 100
\citation 8, 8, 9, 9,
9, 10, 10, 10, 1428
citation-order op-
tion 72, 90
cite package 45
\cite 8, 8, 8, 9, 9, 9,
10, 10, 10, 37,
43, 44, 44, 45,
48, 48, 48, 48, 1284
\cite@a ... 46, 1368,
1369, 1475,
2866, 2876,
2895, 2903, 2909
\cite@b . 46, 1379, 1386
\cite@bc 47, 1387, 1389
\cite@begingroup ..
. 1346, 1362,
1371, 1448, 1455
\cite@cb 1415,
1426, 1464, 1551
\cite@cj 47, 51, 53, 1412
\cite@cjs .. 1499, 1560
\cite@compress
. 1500, 1568,
1588, 1612, 2762
\cite@compress@a ..
. 1572, 1575, 1605
\cite@compress@b ..
. 1581, 1591
\cite@dash 53, 1509,
1563, 1566,
1596, 1600, 1608
\cite@dest . 1636, 1637
\cite@dest@b 1641, 1643
\cite@endgroup 1351,
1363, 1423,
1450, 1456, 1470
\cite@label
. 1011, 1027,
1053, 1054,
1056, 1071,
1074, 1081, 2841
\cite@nobib@test ..
. 1392, 1396
\cite@print . 1550,
1557, 1569,
1580, 1583,
1587, 1598,
1603, 1611, 2762
\cite@single 1285, 1309
\cite@sort . 1503, 1513
\cite@sort@a 1517, 1520
\cite@sort@b 1526, 1536
\cite@sorted@e
. 52, 1480, 1506
\cite@sorted@s
. 51, 1480, 1496
\CiteAltPunct .. 45, 45
\citeAltPunct 1338, 1349
\citeauthor
. 1444, 2894, 2898
\citeauthority 1445, 2897
\citedest 9,
10, 10, 54, 55, 1636
\citeform .. 1339, 1663
\citeleft 1334, 1345,
1353, 1375,
1425, 1458,
2854, 2857, 2859
\citelist
. 48, 1299, 1320,
1340, 1422, 1451
\citemid
. 45, 45, 1334, 1419
\citen 1437
\CiteNames . 2932, 2942
\CiteNamesFull
. 2902, 2908, 2943
\CitePrintUndefined
. 1045, 1401,
1408, 1411,
1498, 1559, 2925
\citepunct 45, 1334,
1345, 1349,
1353, 2860, 2870
\citeright
. 1334, 1354,
1358, 1376,
1420, 1459, 2858
\cites .. 1295, 1436,
1439, 1441,
1452, 2872, 2881
\cites@a 1290,
1311, 1312, 1436
\cites@b ... 1314, 1323
\cites@c ... 1327, 1329
\cites@init . 1352,
1364, 1372, 1457
\cites@o ... 1287, 1310
\cites@oo .. 1310, 1311
\CITASEL 46
\citesel 37,
37, 43, 47, 48,
55, 55, 1282, 2855
\citesel@author ...
. 1374, 2852, 2895
\citesel@authoryear
. 2853, 2855, 2903
\citesel@number ...
. 1280, 1282, 1479
\citesel@object ...
. 2854,
2859, 2876, 2909
\citesel@other
. 1475, 1476
\citesel@update ...
. 1277, 1431
\citesel@write
. 1097, 1102
\citesel@year
. 1281, 2866, 2871
\CloseBBLFile 1263, 1271
\closein 1224
\clubpenalty . 513, 514
cmr font 46
cmsy font 6
collection entry type 7
\compare@stems
. 2175, 2199
\copy@bibdef
. 26, 49, 611,
701, 1166, 1467
\copy@bibdef@a
. 615, 618, 664
\count 13, 1258
\count@ 88
\count@chars 2049, 2058
\cprime 1740
\CPU@normal 1411, 1559
\CPU@sort .. 1495, 1498
\cs 59, 60, 60
\CurLine 42,
42, 42, 1230,
1231, 1237, 1248
\current@bibfield .
. 20, 266,

- 294, 2365, 2382,
2489, 3198, 3203
`\current@citekey` ..
 759, 763, 770, 2192
`\current@primary` 72,
 81, 93, 900,
 906, 912, 915,
 919, 2087, 2352,
 2355, 2362,
 2379, 2486, 2837
`\current@stem` .. 72,
 93, 2074, 2077,
 2081, 2082,
 2094, 2175,
 2829, 2830,
 2837, 2838, 2841
`\current@year`
 74, 2076, 2177,
 2831, 2832, 2843
`\CurrentBib` .. 559,
 562, 767, 1014,
 1092, 1104,
 2683, 2696, 2862
`\CurrentBibType` 27, 597
`\cydot` 1743
- ### D
- `\d` 1701
`\DashPages` .. 6, 6, 2565
`\dateLANGUAGE` 36
`\Dbar` 1737
`\dbar` 1737
`\DeclareBooleanOption`
 127, 128,
 129, 130, 131,
 132, 142, 143,
 144, 145, 146, 147
`\DeclareExclusiveOptions`
 125,
 126, 139, 140, 141
`\DeclareNameAccent`
 1689,
 1692, 1693,
 1694, 1695,
 1696, 1697,
 1698, 1699,
 1700, 1701,
 1702, 1703,
 1704, 1705,
 1706, 1707,
 1708, 1709,
 1710, 1711,
 1712, 1713,
 1714, 1715, 1716
`\DeclareNameSymbol`
 1718,
 1726, 1727,
 1728, 1729,
 1730, 1731,
 1732, 1733,
 1734, 1735,
 1736, 1737,
 1738, 1739,
 1740, 1741,
 1742, 1743, 1744
`\DeclareOption` ... 133
`\DeclareRobustCommand`
 18, 1340, 1408,
 1436, 2865,
 2868, 2875,
 2878, 2894,
 2897, 2900, 2906
`\def` 20, 48, 48
`\defer@bibdef`
 ... 613, 703, 1221
`\deferredquotes` ...
 .. 2295, 2525,
 2526, 2755, 2757
`\deferredquoteslogical`
 2284, 2756
`\DefineAdditiveKey`
 188,
 189, 190, 191,
 192, 193, 194, 195
`\DefineDummyKey` .. 233
`\DefineJournal` 3, 15,
 32, 40, 1152, 2768
`\DefineName`
 . 3, 31, 40, 42,
 1149, 2497, 2498
`\DefinePublisher` ..
 15, 32,
 40, 40, 1158, 2776
`\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` 1731
`\dh` 1731
`\DJ` 1732
`\dj` 1732
`\do` 12, 13, 16, 57
`docstrip` 101
`\dospecials` 13
`\DSK@append` 857
`\DSK@def` 856
`\dudot` 1710
`\DuplicateBibKeyWarning`
 1038, 1089
- ### E
- `\edef` 59, 59
`\editiontext`
 6, 6, 7, 2640, 2646
`editor field` 19
`\emph` 3082,
 3098, 3107,
 3109, 3150,
 3152, 3169,
 3171, 3186, 3187
`\empty@cite` 1391, 1395
`\EmptyNameWarning` ..
 1823, 1873
`\EmptyPrimaryWarning`
 2353, 2358
`\endbibchapter` ... 1140
`\endbibdiv` 1147
`\endbiblist` 1146
`\endbibsection` ... 1138
`\endgroup` 19
`\endinput` 42, 101
`\endlist` 522
`environments:`
 bibchapter ... 1119
 bibdiv 1137
 biblist 1, 23, 23, 484
 biblist* 490
 bibsection ... 1128
 tabbing 60
 thebibliography
 1142
`\eprint` 328, 390, 433,
 451, 2671, 2975,
 3021, 3042, 3060
`\eprintpages` . 325,
 345, 2572, 2972
`\erase@field`
 ... 19, 19, 19,
 19, 289, 2385,
 2492, 3200, 3205
`\errmessage` 262
`\etalchar` 46,
 58, 59, 1715,
 2141, 2146, 2156
`\etaltext` 58,
 1024, 1041,
 1716, 2086,
 2145, 2241,
 2497, 2498,
 2499, 2836, 2937
`\ExecuteOptions` .. 148
`export-bibtex pack-`
 age 27
`\extr@cite` .. 1076,
 1401, 1406,

- 1428, 1435,
 1654, 1665, 1667
 \extract@initial ..
 2001,
 2006, 2023, 2038
 \extract@initial@a
 ... 67, 2008, 2010
 \extract@initial@b
 2015, 2022
 \extract@initial@c
 2017, 2025
 \extract@initial@d
 2032, 2034
 \extract@initials ..
 . 1827, 1874, 2753
 \extract@surnames ..
 . 2123, 2130, 2149
- F**
- \fld@elt 186,
 191, 192, 193,
 194, 195, 2392,
 2550, 2554, 2561
 \font 23
 \fontseries 1409
 \fontshape 1409
 \foo 59
 \format@cite
 . 1663, 1665, 1673
 \frenchspacing
 . 79, 89, 113, 515
 \fsa@n 67
 \fullcite .. 1442, 2900
 \fullocite .. 1443, 2906
 \futurelet 62
- G**
- \G@refundefinedtrue
 1398
 \g@undef .. 58, 668, 1183
 \generate@alphalabel
 . 2078, 2688, 2706
 \generate@label ...
 ... 729, 1010,
 2688, 2706, 2826
 \get@current@properties
 ... 275, 292, 2341
 \get@namepart
 ... 63, 63, 63,
 67, 1814, 1817,
 1820, 1831, 2124
 \get@namepart@a ...
 64, 67,
 1834, 1836, 1848
 \get@namepart@b ...
 64, 67, 1838, 1840
 \get@namepart@c ...
 ... 67, 1842, 1847
 \get@namepart@d ...
 ... 62, 1844, 1850
- \get@namepart@e ...
 1852, 1854
 \get@namepart@f 64,
 64, 68, 1862, 1865
 \get@nth@property 294
 \get@numberof .. 72,
 2088, 2131, 2227
 \get@property 823
- H**
- \H 1702
 har.bst 8
 \hbox 2371
 \hfil 530
 \hfill 561
 hij.bib 8
 \href 2816, 2819
 \hrulefill 2371
 \Hy@backout 1656
 \Hy@raisedlink
 .. 562, 1676, 2862
 \hyper@@link 1665
 \hyper@anchorend ..
 .. 562, 1675, 2862
 \hyper@anchorstart
 .. 562, 1674, 2862
 hyperref package ..
 ... 54, 54, 56, 56
 \hyph@to@space
 2112, 2125
 \hyphenpenalty 30
- I**
- \i .. 59, 60, 1726, 1781
 \if@tempwa .. 612,
 622, 1068, 2181
 \ifBR@verbose 1650, 1653
 \IfEmptyBibField ..
 7, 7,
 17, 185, 2305,
 2459, 2475,
 2476, 2563,
 2573, 2592,
 2595, 2598,
 2737, 2955, 2956
 \IfFileExists
 . 11, 154, 164,
 1198, 1201, 1204
 \ifin@ 674,
 879, 1046, 1348,
 2046, 2400, 2411
 ifoption package .. 14
 \ifsame@stems 2176, 2198
 \ifx 12
 \in@ 673,
 878, 1045, 1347,
 2045, 2399, 2410
 \ini@hyphen
 .. 1876, 1893,
- 1899, 1930,
 1932, 2042, 2043
 initials option . 62, 90
 \InnerCite .. 1325,
 1330, 1355,
 1368, 1460, 2885
 inputenc package .. 59
 \interlinepenalty 512
 inverted attribute .. 63
 isbn field 7
 \issuetext 6, 6, 324,
 344, 2564, 2971
 \itemindent 570
- J**
- \j .. 59, 60, 1727, 1782
 journal field 32
- K**
- \k 1703
 \keep@last@digit ..
 2658, 2662
 \kern 732,
 2372, 2435, 2476
 klm.bib 8
- L**
- \L 1733
 \l 1733
 label field 72, 93
 \labelsep 502, 546, 572
 \labelwidth .. 544,
 545, 566, 567, 568
 \language name 990, 1006
 \lasp 1738
 \lastkern 268, 2277,
 2285, 2297, 2435
 \lastpenalty 2278, 2280
 \lc@do 1763, 1768
 \lc@edef 1759, 2203, 2204
 \lccode 29
 \lcsymbol 58
 \leftmargin 545, 546,
 567, 569, 570, 572
 \let 20, 20, 48, 48
 \lfhook 1711
 \linebreak 5
 \list 503
 lite option 16
 \lncan@a .. 2835, 2851
 logical-quotes op-
 tion 6, 91
 \lowercase 29, 654,
 801, 838, 986, 1764
 ltoutenc.dtx 60
- M**
- \m@ne 265, 1000
 \m@th 1361
 \macrotext . 52, 55, 590

- `\makeatletter` 90
`\makelabel` 508
`\MakeUppercase` . . . 1114
`\mathchar` 30
`\mathcode` 28, 33
`mathscinet` package .
 15, 57, 58, 59, 59
`\mathsurround` 46
`\mdash` 8, 8, 2530
`\mddefault` 1409
`\mdseries` 1361
`\meaning` 20, 52
`\MessageBreak`
 . . 106, 1296, 1305
`\MessageBreakNS` . . .
 106, 1092
`\modify@xref@fields`
 630, 648
`\ModifyBibLabel` . . .
 2186, 2213
`\month@` 87
`\moo` 18, 19, 19
`\MR` 2540
`\MR@url` 2809, 2816, 2819
`\MRhref` 2544, 2548, 2813
`msc-links` option . . . 92
`\MultipleBibSelectWarning`
 1187, 1189
`\MultipleCiteKeyWarning`
 1285, 1293
- N**
- `name` entry type 3
`\name` . . . 72, 72, 72,
 72, 93, 93, 93,
 93, 186, 188,
 189, 190, 788,
 804, 808, 814,
 831, 2085, 2088,
 2129, 2139,
 2140, 2144,
 2147, 2322,
 2835, 2933, 2944
`\name@split` . . . 62,
 63, 63, 1813, 2336
`\name@split@finish`
 1820, 1822
`\name@split@given` .
 . . . 63, 1814, 1816
`\name@split@jr` . . .
 1817, 1819
`\NameCheck` 1825
`\ndash` 8, 8, 86,
 2530, 2566, 2571
`\NeedsTeXFormat` 2, 3072
`\newcount` . . 2170, 2216
`\newcounter` . . 483, 532
`\newdimen` 524
`\newif` 2198
- O**
- `\newlabel` 536
`\newread` 1196
`\NG` 1734
`\ng` 1734
`\NoBibDBFile` 1208, 1215
`\nobreak` 66
`\nobreakspace` 1877, 2617
`nobysame` option . . . 128
`\nocite` 49, 1446
`non-compressed-cites`
 option 91
`non-sorted-cites` op-
 tion 15, 91
`\NonNumericCiteWarning`
 1482, 1486
`\nopunct` . . 6, 6, 122,
 1345, 1565, 2459
`\normal@bibdef`
 26, 26,
 594, 596, 609, 1185
`\NormalCatcodes` . . . 37
`\normalfont` 496
`\normalize@edef` . . .
 1745,
 1881, 2087, 2837
`\number`
 30, 2296, 2617,
 2642, 2648, 2658
`\numeric@refs` . 721,
 2690, 2708, 2856
- P**
- `\PackageError` 105
`\PackageInfo` 1654
`\PackageWarning` . . 104
`\PackageWarningNoLine`
 4
`\parenthesize` . . . 5,
 5, 5, 6, 6, 84,
 329, 369, 391,
 435, 453, 2504,
 2976, 3000,
 3022, 3044,
 3062, 3084, 3100
`\parse@MR` . . 2808, 2815
`partial` field 84
`pcatcode` package . . 16
`place` field 3, 3, 3
`\Plural` . . . 7, 7, 77,
 79, 2301, 2380,
 2383, 2388,
 2495, 3199, 3204
`\polhk` 1713
`\pop@bracket` 2505, 2508
`\pop@bracket@a`
 2516, 2519
`\PopCatcodes` 2825, 3069
`\pp@scan` . . . 2566, 2569
`\pp@scan@a` . . 2566, 2571
`\prev@cite` . . . 52,
 52, 52, 52, 52,
 52, 1510, 1545,
 1553, 1580,
 1598, 1601,
 1602, 1609, 1610
`\prev@cite@cb`
 . . . 52, 1546, 1597
`\prev@citekey`
 . . . 760, 770, 2186
`\prev@names`
 . . 1365, 2922, 2924
`\previous@primary` .
 81,
 521, 901, 906,
 2362, 2379, 2486
`\previous@stem` 2073,
 2081, 2175, 2829
`\previous@year` . 74,
 2075, 2177, 2831
`\prime` 6
`\print@backrefs` . . .
 1658, 2712

- \print@citenames .. 3119, 3137, 3162, 3177, 3191
- 2902, 2908, 2928, 2942
- \print@contribution .. 2393, 2394, 2395, 2396, 2398
- \print@date . 2443, 2606, 2607, 2608, 2628, 2954
- \print@edition 2630, 2633
- \print@month@day .. 2611, 2615
- \print@one@dash ... 1564, 1596
- \print@partial 2501, 2555, 2556, 2557, 2558
- \print@primary 913, 916, 920, 2355
- \PrintAuthors ... 5, 5, 6, 6, 6, 315, 378, 444, 476, 920, 2361, 2962, 3009, 3053, 3076, 3128, 3185
- \PrintBackRefs 767, 1661, 2712
- \PrintBook ... 384, 2447, 3016, 3134
- \PrintCiteNames ... 2852, 2854, 2916, 2919
- \PrintCNY .. 2853, 2915
- \PrintConference .. 383, 2421, 3015, 3133
- \PrintConferenceDetails .. 401, 2424, 3146
- \PrintConferenceDetails@ . 2428, 2431, 2434
- \PrintContributions 319, 340, 361, 382, 411, 427, 2390, 2967, 2993, 3014, 3037, 3080, 3097, 3114, 3132, 3157, 3172
- \PrintDate ... 8, 8, 8, 2596, 2601, 2606, 2953, 2963, 2986, 3010, 3032, 3054
- \PrintDateB .. 367, 386, 417, 432, 450, 2607, 3086, 3102,
- 3119, 3137, 3162, 3177, 3191
- \PrintDatePosted .. 2599, 2628
- \PrintDatePV 323, 343, 2594
- \PrintDOI . 327, 389, 2590, 2974, 3020
- \PrintEdition 6, 7, 7, 7, 8, 358, 408, 426, 2629, 2990, 3036, 3118, 3161
- \PrintEditorsA .. 6, 6, 7, 7, 7, 80, 81, 916, 2378, 3197
- \PrintEditorsB 6, 7, 7, 359, 409, 2387, 2991
- \PrintEditorsC 7, 7, 7, 2375
- \PrinteSeries@a .. 2226
- \PrintISBNs 2560
- \PrintNameList 350, 2374, 3112, 3113, 3155, 3156
- \PrintNames ... 80, 2318, 2366, 2374, 2376, 2383, 2388, 2483, 2490, 2495, 3199, 3204
- \PrintNames@a 2319, 2321
- \PrintPartials 320, 2552, 2968, 3081
- \PrintPrimary . 6, 6, 354, 422, 2351, 2473, 2985, 3031, 3106, 3168
- \printref 1662
- \PrintReprint . 331, 371, 393, 437, 455, 2450, 2978, 3002, 3024, 3046, 3064, 3089, 3123, 3139, 3179, 3193
- \PrintReviews . 7, 7, 334, 374, 396, 440, 458, 2549, 2981, 3005, 3027, 3049, 3067
- \PrintSeries .. 75, 76, 77, 77, 80, 2217, 2322, 2391, 2550, 2553, 2561, 2885, 2933, 2944
- \PrintSeries@a 77, 2224, 2226
- \PrintThesisType .. 447, 2575, 3057, 3188
- \PrintTranslation . 330, 370, 392, 436, 454, 2464, 2977, 3001, 3023, 3045, 3063, 3088, 3122, 3138, 3178, 3192
- \PrintTranslatorsA 7, 7, 81, 913, 2485, 3202
- \PrintTranslatorsB 7, 7, 2494
- \PrintTranslatorsC 7, 7, 360, 410, 2482, 2992
- \PrintYear 8, 8, 8, 88, 2627
- proceedings entry type 7
- \process@bare@dot . 1964, 1974
- \process@citelist . 1356, 1494, 2689, 2707, 2765, 2864
- \process@citelist@sorted 1478, 1494
- \process@citelist@unsorted 1483, 1491, 2689, 2707, 2765, 2864
- \process@dots .. 66, 66, 69, 1885, 1947
- \process@hyphens 66, 68, 69, 1884, 1894
- \process@names 1886, 1998
- \process@one@dot .. 68, 1950, 1958, 1980, 1995
- \process@one@dot@a 1960, 1962
- \process@one@dot@b 1967, 1979
- \process@one@dot@c 1969, 1982
- \process@one@dot@d . 1977, 1989, 1991
- \process@one@hyphen 67, 1897, 1904, 1926, 1941
- \process@one@hyphen@a 68, 69, 1906, 1908
- \process@one@hyphen@b . 1914, 1917, 1925

- \process@one@hyphen@c 296, 602, 624, 2240, 2242,
 1919, 1928 648, 2402, 2412 2247, 2249,
 \process@one@hyphen@d review field 5, 7 2259, 2262, 2401
 ... 67, 1935, 1937 rkeyval package ... \series@total
 \process@xrefs 620, 646 5, 10, 19, 67 .. 2228, 2229,
 \ProcessExclusiveOptions \rkv@DSAK 19 2253, 2259, 2262
 152 \rkv@setter .. 874, 894 \set 631,
 \ProcessOptions .. 150 \rkvIfAdditive ... 274 632, 633, 634,
 \prop ... 284, 286, 2344 \rkvIfEmpty ... 16, 185 635, 636, 637,
 \prop@reset 297 \romannumeral 106 638, 639, 640, 641
 \protected@edef ... \rsk@resume 690 \set@firstname
 1025, \rsk@set 27, 624 . 2319, 2329, 2912
 1079, 1121, 1130 \rsk@set@a 19 \set@name ... 2330,
 \ProvidesFile 1219 \rsk@set@b 19 2333, 2335, 2913
 \ProvidesPackage 3, 3073 \rsk@toks \set@name@a
 publisher field 32 10, 11, 26, 30, ... 80, 2337, 2339
 \purge@accent 1779, 1810 30, 297, 603, \set@namea 2339
 \purge@edef 648, 2403, 2412 \set@othername
 . 1775, 2201, 2202 .. 2319, 2324,
 \purge@edef@ 2325, 2326, 2332
 .. 1785, 1790, \safe@set 81, \setbib@ 26, 26, 26,
 1800, 1801, 1805 1514, 1521, 27, 29, 29, 30, 30
 \purge@edef@a 1537, 1571, \setbib@@ 724,
 ... 61, 1792, 1794 1578, 1594, 2159 733, 738, 741,
 \push@bracket 2505, 2508 \same@stemfalse . 2208 744, 747, 946, 2479
 \PushCatcodes 37 \same@stemtrue .. 2206 \setbib@article 733, 946
- R**
- \r 1704 \save@labelwidth .. \setbib@book 2461
 \rasp 1739 519, 533 \setbib@conference 2422
 \read 1230 \save@primary . 769, 902 \setbib@contribution
 2403
 \ReadBibData 1180, 1197 \section* 39 \setbib@incollection
 1209, 1213, 1218 469, 738, 741, 744
 \ReadBibLoop \selbibdef@a . 693, 695 \setbib@innerbook 2448
 1222, 1226, 1240 \selbibdef@b . 697, 700 \setbib@inproceedings
 472
 \ReadBibLoop@a \select@auxlanguage \setbib@misc 600
 1231, 1235 ... 276, 283, 2342 \setbib@nameBE
 2308, 2345
 \ReadBibLoop@b \selective@bibdef . \setbib@nameinverted
 1242, 1246 26, 2313, 2913
 49, 692, 1169, 1467 \setbib@nameLE
 35, 36, 286, 992 . 2303, 2330, 2333
 \ReadBibLoop@e ... 1235 \SentenceSpace 334, \setbib@partial .. 2502
 4, 12, 374, 396, 440, \setbib@TYPE 18
 36, 124, 155, 458, 1659, 2670, \setcounter 485
 165, 178, 181, 2981, 3005, \settowidth .. 526, 566
 182, 2804, 3074 3027, 3049, 3067 \sfcode 78,
 \reset@nth@property 826 \series@add 2235, 2239 114, 115, 116,
 \resetbiblist 77 117, 118, 119,
 ... 542, 565, 1144 \series@add@a 2219, 2251 552, 2287, 2445
 \ResetCapSFCodes .. \series@add@b 2220, 2254 \shade@cite
 516, 549 . 2221, 2256, 2263 . 1662, 1663, 1670
 \resolve@inner 2409, \series@add@d 2222, 2260 shaderef package .. 56
 2422, 2448, \series@add@e 2223, 2245 short-journals op-
 2451, 2465, 2502 \series@index tion 15, 91
 \restore@labelwidth 20, 265, short-months option 91
 504, 540 294, 787, 799, short-publishers op-
 \RestrictedSetKeys 826, 2216, 2234, tion .. 15, 91, 91
 11, 19,

- shortalphabetic option 89
 showkeys package .. 2966, 2987, 2989, 3011, 3013, 3033, 3035, 3055, 3056
 \SingularPlural ... 7, 7, 77, 79, 2230, 2232, 2301
 \sist@@n ... 1260, 1266
 \SK@ .. 1017, 1667, 1685
 \SK@@label .. 1017, 1684
 \SK@@ref 1667
 \soft 1714
 sorted option 14
 \space 67
 \spacefactor .. 110, 122, 2287, 2289, 2445, 2541, 2670
 \SS 1735
 \ss 1735
 \star@ 682, 1310, 1368, 2866, 2876, 2895, 2903, 2909
 \star@bibdef .. 26, 579, 607, 664, 1221
 \stepcounter 500
 \string 60
 \strip@MRprefix ... 2532, 2543
 \strip@prefix ... 20, 52
 \SubEtal 2243, 2245, 2499
 \surround@names ... 2114, 2116
 \SwapBreak 78, 78, 270, 2220, 2221, 2222, 2223, 2272, 2388, 2442, 2495
 \symbol 58, 58

T
 \t 1705
 tabbing environment 60
 \textbf 5, 322, 342, 2970
 textcmds package .. 8, 15, 85
 \textemdash ... 8, 2530
 \textendash 8, 1565, 2531
 \textfont 23
 \textit 78, 316, 318, 339, 355, 357, 379, 381, 423, 425, 445, 446, 2964, 2987, 2989, 3011, 3013, 3033, 3035, 3055, 3056
 \textquotedblleft 2524
 \textquotedblright 2527
 \textsc 76
 \textsubscript 6
 \textsuperscript ... 6
 \texttt 20
 \tf@brf 2718, 2719
 \TH 1736
 \th 1736
 \the 26, 30, 30
 \thebib 563, 566, 1015, 2683, 2685, 2696, 2698
 thebibliography environment ... 1142
 \thebibliography .. 90
 \thesis@type 2576, 2579
 \thinspace 2371
 \thr@@ 2253, 2651
 \toks@ 18, 63
 \tprime 6, 6, 6, 6
 \TrailingHyphenWarning 1913, 1944
 transition field ... 5, 5, 5, 17, 31
 \tsub 6, 6
 \tsup 6, 6
 \ttfamily 1409
 \tw@ 2184, 2242
 \typeout 172, 1175, 1177

U
 \u 1706
 \uarc 1709
 \udot 1712
 \UndefinedCiteWarning 1399, 1404
 \unkern 2277
 \unpenalty 2281
 \unskip ... 66, 1893, 2276, 2541, 2628
 \uparrow 1659
 \upn 84, 85, 2505, 2507, 2520
 upref package 85
 \upshape 1361

 \url 18, 2671
 \use@accent 1749, 1755, 1761
 \utilde 1708

V
 \v 1707
 \vdef ... 53, 1315, 1380
 \voltext ... 6, 6, 6, 363, 413, 2563, 2995, 3111, 3154
 \vphantom 2816

W
 \widowpenalty 514
 \wrap@accent 61, 1748, 1756
 wrapper macros, definition 12

X
 xref field 32, 33
 \xref@add@toks 856, 871
 \xref@append .. 857, 892
 \xref@check@a 775, 776, 777, 781
 \xref@check@aa 788, 798
 \xref@check@ab 810, 817
 \xref@check@b 778, 779, 834
 \xref@check@c 32, 774, 845, 850
 \xref@ID 860, 861, 862, 864
 \XRefWarning 862, 899, 2415
 \XrefWarning 899
 xyz.aux 8
 xyz.bbl 9, 9
 xyz.tex 8

Y
 y2k option .. 71, 92, 127
 \ycite 48, 1438, 1440, 1444, 1445, 2865, 2898
 \ycites 1439, 2868
 \year@short 2157, 2164, 2794

Z
 \Z 554
 \zap@space .. 1317, 1382