

# The `ref2bib` Package

Rob J. de Boer,  
Theoretical Biology,  
Utrecht University,  
Email: `R.J.DeBoer@bio.uu.nl`

This document accompanies `ref2bib-2.0`

## Abstract

`ref2bib` is a package for converting queries from literature data bases (such as Medlars, Nature and ISI) into `BIBTEX` files. The packages automatically names each entry, reports conflicts with existing entries, searches for keywords in the queries, and allows for keyword searching in the `BIBTEX` file.

## 1 Overview

Automatic conversion of literature searches from data bases such as Medline saves one the tedious job of typing references. This packages reads queries from such databases, converts a file of queries into a `BIBTEX`-file, splits this `BIBTEX`-file into its individual `BIBTEX`-items to test for possible conflicts with pre-existing items, and merges all your `BIBTEX`-items into your personal `BIBTEX`-file.

Meanwhile the package scans the query and its abstract for keywords that you have pre-defined in the `keyword`-file. With simple “or” and “and” like searches you can make selections from your personal `BIBTEX`-file based upon these keywords, and/or any element from your `BIBTEX`-items.

The package consists of about 10 scripts and C-programs that need to be installed on your Unix system in some directory in in your `path` (e.g., in `/usr/local/bin` or `username/bin`). Each individual user needs to set one environment variable `BIBDIR`, which defines the location of the personal `BIBTEX`-file, `keyword`-file, `sedfile`, and two directories with the individual

BIB<sub>T</sub>E<sub>X</sub>-items. The main directory is called `Files` and contains all accepted BIB<sub>T</sub>E<sub>X</sub>-items as individual files. The directory `Conflicts` contains BIB<sub>T</sub>E<sub>X</sub>-items having the same name, but different contents, as an item in the `Files` directory. Such conflicts are reported when BIB<sub>T</sub>E<sub>X</sub>-file is split up, and have to be resolved manually.

To minimize the occurrence of conflicts BIB<sub>T</sub>E<sub>X</sub>-items receive quite informative names. For instance the 1999 paper by De Boer *et al.* in *Proc. Nat. Acad. Sci. USA* will be named `DeBoer.pnas99`. This avoids conflicts with other publications from the same author in the same year. The rule for naming items is: first author, dot, name of journal abbreviated by the first letters of first four words, followed by the last two digits of the year of publication (no millenium problems in BIB<sub>T</sub>E<sub>X</sub>). Thus the 1953 Watson and Crick paper in *Nature* will be called `Watson.n53` by `ref2bib`. Obviously, in your L<sup>A</sup>T<sub>E</sub>X document you refer to these two papers as `\cite{DeBoer.pnas99, Watson.n53}`.

## 2 Installation

If `ref2bib` is already installed on your system you only need to make a directory, to set the `BIBDIR` environment variable to this directory, and to call the `bsetup` tool. Thus if you want your BIB<sub>T</sub>E<sub>X</sub>-file to reside in a directory called `bibtex` in your home-directory, you do the following:

```
cd
mkdir bibtex
setenv BIBDIR ~/bibtex
bsetup
```

The line `setenv BIBDIR ~/bibtex` should also be added to your `.cshrc` file to make this definition permanent. `bsetup` will make two empty directories and an example keyword file. Edit this keyword file with your own set of keywords. Then you are ready to go.

If `ref2bib` is not installed on your system you should download the `tarfile` or `rpm` from <http://theory.bio.uu.nl/rdb/software.html>. Unpack this tarfile with `tar xf ref2bib.tar` or install the rpm with `rpm -Uhv ref2bib-2.0-1.i386.rpm`. When using the tarfile, edit the `makefile` in the `ref2bib` directory such that the `BINDIR` variable contains the directory you want to have the `ref2bib` scripts and executables in. After editing you simply type `make install`, which will call the C-compiler and copy everything into the `BINDIR`. You may test the compilation first with a `make all`, and may clean up afterwards with a `make clean`.

## 3 Manual

To convert a data base query into BIB<sub>T</sub>E<sub>X</sub>-entries you need to call three programs. The first `bconvert` converts the query into a BIB<sub>T</sub>E<sub>X</sub>-file. The second `bsplit` splits this BIB<sub>T</sub>E<sub>X</sub>-file up into the BIBDIR/Files directory, or reports conflicts that are stored in the BIBDIR/Conflicts directory. Finally you may merge the new items with the old ones by calling `bmerge` which merges all files into a BIB<sub>T</sub>E<sub>X</sub>-file named BIBDIR/username.bib. Thus the sequence is `bconvert query; bsplit query.bib; bmerge`. This whole sequence can be executed at once with `bupdate query`.

Conflicts can easily be resolved by combining the conflicting items into one file, renaming all items by adding a letter to their name (e.g., `Watson.n53a`) and subsequently splitting this file with another call of `bsplit`. Having a `Watson.n53a` the next `Watson.n53` will be recognized as a conflict when the file is different from both `Watson.n53` and `Watson.n53a`. We advise you to rename both the original and the new item (`Watson.n53a` and `Watson.n53b`), but only renaming the new item is also valid (`Watson.n53` and `Watson.n53a`).

Finding BIB<sub>T</sub>E<sub>X</sub>-items in your BIB<sub>T</sub>E<sub>X</sub>-file can be done with `bfind` and `bnot`. For instance `bfind -i BIBDIR/username.bib word1 word2 | bfind word3 | bnot -o tmp.bib word4` makes a BIB<sub>T</sub>E<sub>X</sub>-file with the name `tmp.bib` containing the words “word1” or “word2” and “word3” but not “word4”. The unix `|` symbol pipes the output from the first `bfind` into the next.

The keyword-file can be edited with any texteditor. See the Section on keywords for examples. You can keep your keywords sorted with `bsortkw` and make a listing of the file with `blistkw`

Detailed info on the options of all commands are available online with the `bmanual` call. Here is its output:

### `bsetup`

Creates the required files in BIBDIR

### `bconvert [-v -k keyword] file1 file2`

Converts citations to file1.bib file2.bib

-k adds a keyword, -v means verbose

### `bsplit file1.bib file2.bib`

Splits bibtex files into individual entries in BIBDIR/Files

### `bmerge`

Merges all entries in BIBDIR/Files into BIBDIR/USER.bib

bupdate file1 file2  
Calls bconvert, bsplit, bmerge for all files.

bkeys  
Reads a text and reports keywords

bfind [-i ifile -o ofile] word1 word2  
Reads a bibtex file and makes a bibtex file  
of those entries that contain any of the listed words.  
For example to list entries containing word1 & word2 use  
bfind -i BIBDIR/USER.bib word1 | bfind -o test.bib word2

bnot [-i ifile -o ofile] word1 word2  
As bfind but selects entries NOT containing word1 word2.

bsortkw  
Sorts the keyword file

blistkw  
Lists the keyword file

blist file  
Transforms a file.bib into a nice file.dvi listing.

blat file  
Calls latex, bibtex, latex, latex.

## 4 keywords

The keyword file lists keywords and their synonyms line by line. For example  
virus: virus\*,virion\*;  
will add the keyword “virus” whenever the string “virus” or the string “virion” (possibly followed by several characters) is found in the query. Sequences of words can also be defined. For example

mhc: major\_histocompatibility\_complex  
will add the keyowrd “mhc” whenever the three words Major Histocompatibility Complex are found in a row (separated by spaces or other non-alphabetic characters).

A small sample from my own keyword file:

```
aff_mat: affinity_maturation*, somatic_mutat*, hyper_mutat*;  
ca: cellular_automat*;  
chemotaxis: chemotactic;
```

```

clonal_deletion;
dth:  delayed_type_hypersens*;
flu:  influenza, Parainfluenza;
hiv:  human_immunodeficiency_virus, aids;

```

## 5 sedfile

In your BIBDIR directory you will find a file called `sedfile`. This file is called by the `bconvert` program for additional editing of the query. This is most important for data bases calling “J. Exp. Med.” “J Exp Med” and “Science” “Science”. Because one has to add the dots in the case of “J Exp Med” you will also get entries with “Science.” as an incorrect journal name. Thus by means of the `sedfile` we manually substitute “Science.” by “Science” again.

Just check the `sedfile` for its format, note that we make several more changes, and adjust it to your own needs if necessary. For example, to get the names of organisms in italics, add to the `sedfile`:

```
s/Dictyostelium/{\emph{Dictyostelium}}/
```

etc.

To be able to write the Von-part in lower case, add to the `sedfile`:

```

/AUTHOR/s/Van /{\uppercase{v}an} /
/AUTHOR/s/van /{\uppercase{v}an} /
/AUTHOR/s/Von /{\uppercase{v}on} /
/AUTHOR/s/von /{\uppercase{v}on} /
/AUTHOR/s/De /{\uppercase{d}e} /
/AUTHOR/s/de /{\uppercase{d}e} /
/AUTHOR/s/Der /{\uppercase{d}er} /
/AUTHOR/s/der /{\uppercase{d}er} /
/AUTHOR/s/Da /{\uppercase{d}a} /
/AUTHOR/s/da /{\uppercase{d}a} /

```

If you then prefer to write the Von-part in lower case, simply precede the `\bibliography{}` line with: `\renewcommand{\uppercase}{\lowercase}`.

## 6 Example formats

In the `formats` directory examples are given of the formats, on which our program is tested:

<code>query.fcgi</code>	Medline format
<code>isi.txt</code>	ISI format
<code>nature.txt</code>	Nature format
<code>citmgr_medlars</code>	PNAS format

## 7 Medline

The `bconvert` program by default reads queries in the Medline/Unix textfile format. In the PubMed data-base you save your Medline queries by selecting the **Medline format** and click the **save to file**. Next convert this query with either `bconvert query` or `bconvert -m query`.

## 8 ISI

Export your citations from the “marked list” with the default “Field Tagged” options clicking the **source abbrev.** box.

## 9 Credits

Stan Marée wrote various modules.

# Contents

1	Overview	1
2	Installation	2
3	Manual	3
4	keywords	4
5	sedfile	5
6	Example formats	6
7	Medline	6
8	ISI	6
9	Credits	6