

Crossrefware documentation*

Boris Veytsman[†]

May 26, 2025

Contents

1	Introduction	1
2	ltx2crossrefxml.pl	2
3	bibdoiadd.pl	7
4	bibmradd.pl	9
5	bibzbladd.pl	10
6	biburl2doi.pl	11
7	bbl2bib.pl	12

1 Introduction

These scripts can be used to create files for submission to Crossref, check and add doi numbers, MathSciNet numbers and ZbMath numbers to papers, and to convert ‘bbl’ files to ‘bib’ files.

Development sources and issue tracker are on github: <https://github.com/borisveytsman/crossrefware>. Releases are made on CTAN: <https://ctan.org/pkg/crossrefware> and from there included in T_EX Live and other distributions.

The script `ltx2crossrefxml` extracts information from `.rpi` files and (if present) `.bbl` files and generates an XML file suitable for submission to crossref.org. (Crossref is the organization that handles DOI numbers for scholarly papers.) It does not actually upload the submission, just outputs XML.

This `.rpi` file is a plain text representation of the metadata for one article. It is written by the `resphilosophica` package (<https://ctan.org/pkg/resphilosophica>) and the TUGboat publication procedure (<https://tug.org/TUGboat/repository.html>). It can also be created by hand.

*This work was commissioned by Saint Louis University and Princeton University (Mathematics Department)

[†]borisv@lk.net, boris@varphi.com

Several scripts, `bibdoiadd`, `bibmradd` and `bibzbladd` take a `bib` file, and add to each entry a DOI, MR or ZBL number correspondingly, if they can find this entry in the corresponding database. The output of these scripts reformats the BibTeX entries where the respective fields were not already present.

The `bb12bib` script tries to reconstruct a `bib` file from the corresponding `thebibliography` environment. One can argue that this operation is akin to reconstructing the cow from a steak. The way the script does it is by searching for the entry in the MR database, and creating the corresponding BibTeX fields.

I am grateful to Josko Plazonic from the Princeton mathematics department whose (unpublished) Python script was an inspiration for this suite.

Following are manual pages for these scripts. See also the `BibTeX::Parser` package (<https://ctan.org/pkg/bibtexperllibs>).

2 ltx2crossrefxml.pl

Create XML files for submitting to crossref.org

SYNOPSIS

```
ltx2crossrefxml [-c config_file] [-o output_file] [-input-is-xml] latex_file1 latex_file2 ...
```

OPTIONS

-c *config_file*

Configuration file. If this file is absent, defaults are used. See below for its format.

-o *output_file*

Output file. If this option is not used, the XML is output to stdout.

-rpi-is-xml

Do not transform author and title input strings, assume they are valid XML.

The usual `--help` and `--version` options are also supported. Options can begin with either `-` or `--`, and ordered arbitrarily.

DESCRIPTION

For each given *latex_file*, this script reads `.rpi` and (if they exist) `.bbl` and `.aux` files and outputs corresponding XML that can be uploaded to Crossref (<https://crossref.org>). Any extension of *latex_file* is ignored, and *latex_file* itself is not read (and need not even exist).

Each `.rpi` file specifies the metadata for a single article to be uploaded to Crossref (a `journal_article` element in their schema); an example is below. These files are output by the `resphilosophica` package (<https://ctan.org/pkg/resphilosophica>), `aomart` package (<https://ctan.org/pkg/aomart>), the TUGboat publication procedure (<https://tug.org/TUGboat/repository.html>) and other packages, but (as always) can also be created by hand or by whatever other method you implement.

Any `.bbl` and `.aux` files present are used for the citation information in the output XML. See the *CITATIONS* section below.

Unless `--rpi-is-xml` is specified, for all text (authors, title, citations), standard TeX control sequences are replaced with plain text or UTF-8 or eliminated, as appropriate. The `LaTeX::ToUnicode::convert` routine is used for this (<https://ctan.org/pkg/bibtexperllibs>). Tricky TeX control sequences will almost surely not be handled correctly.

If `--rpi-is-xml` is given, the author and title strings from the `rpi` files are output as-is, assuming they are valid XML; no checking is done.

Citation text from `.bbl` files is always converted from LaTeX to plain text.

This script just writes an XML file. It's up to you to do the uploading to Crossref; for example, you can use their Java tool `crossref-upload-tool.jar` (<https://www.crossref.org/education/member-setup/direct-deposit-xml/https-post>).

For the definition of the Crossref schema currently output by this script, see https://data.crossref.org/reports/help/schema_doc/5.3.1/index.html with additional links and information at <https://www.crossref.org/documentation/schema-library/metadata-deposit-schema-5-3-1/>.

CONFIGURATION FILE FORMAT

The configuration file is read as Perl code. Thus, comment lines starting with # and blank lines are ignored. The other lines are typically assignments in the form (spaces are optional):

```
$variable = value ;
```

Usually the value is a "string" enclosed in ASCII double-quote or single-quote characters, per Perl syntax. The idea is to specify the user-specific and journal-specific values needed for the Crossref upload. The variables which are used are these:

```
$depositorName = "Depositor Name";
$depositorEmail = 'depositor@example.org';
$registrant = 'Registrant'; # organization name
$fullTitle = "FULL TITLE"; # journal name
$issn = "1234-5678"; # required
$abbrevTitle = "ABBR. TTL."; # optional
$coden = "CODEN"; # optional
```

For a given run, all .rpi data read is assumed to belong to the journal that is specified in the configuration file. More precisely, the configuration data is written as a `journal.metadata` element, with given `full_title`, `issn`, etc., and then each .rpi is written as `journal.issue` plus `journal.article` elements.

The configuration file can also define one Perl function: `LaTeX.ToUnicode.convert.hook`. If it is defined, it is called at the beginning of the procedure that converts LaTeX text to Unicode, which is done with the `LaTeX::ToUnicode` module, from the `bibtexperl` package (<https://ctan.org/pkg/bibtexperl>). The function must accept one string (the LaTeX text), and return one string (presumably the transformed string). The standard conversions are then applied to the returned string, so the configured function need only handle special cases, such as control sequences particular to the journal at hand.

RPI FILE FORMAT

Here's the (relevant part of the) .rpi file corresponding to the `rpsample.tex` example in the `resphilosophica` package (<https://ctan.org/pkg/resphilosophica>):

```
%authors=Boris Veytsman\and A. U. Th{\o }r\and C. O. R\"espondent
%title=A Sample Paper:\ \emph {A Template}
%year=2012
%volume=90
%issue=1--2
%startpage=1
%endpage=1
```

```
%doi=10.11612/resphil.A31245
%paperUrl=http://borisv.lk.net/paper12
%publicationType=full_text
```

Other lines, some not beginning with %, are ignored (and not shown). For more details on processing, see the code.

The %paperUrl value is what will be associated with the given %doi (output as the **resource** element). Crossref strongly recommends that the url be for a so-called landing page, and not directly for a pdf (<https://www.crossref.org/education/member-setup/creating-a-landing-page/>). Special case: if the url is not specified, and the journal is *Res Philosophica*, a special-purpose search url using *pdcnnet.org* is returned. Any other journal must always specify this.

The %authors field is split at \and (ignoring whitespace before and after), and output as the **contributors** element, using **sequence="first"** for the first listed, **sequence="additional"** for the remainder. The authors are parsed using `BibTeX::Parser::Author` (<https://ctan.org/pkg/bibtexperllibs>).

If the %publicationType is not specified, it defaults to **full_text**, since that has historically been the case; **full_text** can also be given explicitly. The other values allowed by the Crossref schema are **abstract_only** and **bibliographic_record**. Finally, if the value is **omit**, the **publication_type** attribute is omitted entirely from the given **journal_article** element.

Each .rpi must contain information for only one article, but multiple files can be read in a single run. It would not be difficult to support multiple articles in a single .rpi file, but it makes debugging and error correction easier to keep the input to one article per file.

MORE ABOUT AUTHOR NAMES

The three formats for names recognized are (not coincidentally) the same as BibTeX:

```
First von Last
von Last, First
von Last, Jr., First
```

The forms can be freely intermixed within a single %authors line, separated with \and (including the backslash). Commas as name separators are not supported, unlike BibTeX.

In short, you may almost always use the first form; you shouldn't if either there's a Jr part, or the Last part has multiple tokens but there's no von part. See the `btxdoc` ("BibTeXing" by Oren Patashnik) document for details. The authors are parsed using `BibTeX::Parser::Author` (<https://ctan.org/pkg/bibtexperllibs>).

In the %authors line of a .rpi file, some secondary directives are recognized, indicated by | characters. Easiest to explain with an example:

```
%authors=|organization|\LaTeX\ Project Team \and Alex Brown|orcid=123
```

Thus: 1) if `|organization|` is specified, the author name will be output as an **organization** contributor, instead of the usual **person_name**, as the Crossref schema requires.

2) If `|orcid=value|` is specified, the *value* is output as an ORCID element for that **person_name**.

These two directives, `|organization|` and `|orcid|` are mutually exclusive, because that's how the Crossref schema defines them. The = sign after `orcid` is required, while all spaces after the

orcid keyword are ignored. Other than that, the ORCID value is output literally. (E.g., the ORCID value of 123 above is clearly invalid, but it would be output anyway, with no warning.)

Extra | characters, at the beginning or end of the entire %authors string, or doubled in the middle, are accepted and ignored. Whitespace is ignored around all | characters.

CITATIONS

Each .bbl file corresponding to an input .rpi file is read and used to output a citation_list element for that journal_article in the output XML. If no .bbl file exists for a given .rpi, no citation_list is output for that article.

The .bbl processing is rudimentary: only so-called unstructured_citation references are produced for Crossref, that is, the contents of the citation (each paragraph in the .bbl) is dumped as a single flat string without markup.

Bibliography text is unconditionally converted from TeX to XML, via the method described above. It is not unusual for the conversion to be incomplete or incorrect. It is up to you to check for this; e.g., if any backslashes remain in the output, it is most likely an error.

Furthermore, it is assumed that the .bbl file contains a sequence of references, each starting with \bibitem{KEY} (which itself must be at the beginning of a line, preceded only by whitespace), and the whole bibliography ending with \end{thebibliography} (similarly at the beginning of a line). A bibliography not following this format will not produce useful results. Bibliographies can be created by hand, or with BibTeX, or any other method.

The key attribute for the citation element is taken as the KEY argument to the \bibitem command. The sequential number of the citation (1, 2, ...) is appended. The argument to \bibitem can be empty (\bibitem{}), and the sequence number will be used on its own. Although TeX will not handle empty \bibitem keys, it can be convenient when creating a .bbl purely for Crossref.

The .rpi file is also checked for the bibliography information, in this same format.

The structured citations according to the Crossref scheme (https://data.crossref.org/reports/help/schema_doc/5.3.1/common5_3_1_xsd.html#citation) are added by the following methods.

If an .aux file is present, it is checked for any bibdata commands. The bib files in these commands are read, and the information there is used to generate XML entries. The script uses kpsewhich to look for bibfiles, so the usual BibTeX conventions for the search paths are followed.

If there is a .crbib file with the structured citations, for example, generated by Norman Gray's beastie program (<https://heptapod.host/nxg/beastie>) via beastie extract-bib.scm -O crossref \$(doc).aux, as invoked in the TUGboat Common.mak file, the entries in it take precedence.

EXAMPLES

```
ltx2crossrefxml.pl ../paper1/paper1.tex ../paper2/paper2.tex \  
-o result.xml
```

```
ltx2crossrefxml.pl -c myconfig.cfg paper.tex -o paper.xml
```

AUTHOR

Boris Veytsman <https://github.com/borisveytsman/crossrefware>

COPYRIGHT AND LICENSE

Copyright (C) 2012-2025 Boris Veytsman

This is free software. You may redistribute copies of it under the terms of the GNU General Public License (any version) <https://www.gnu.org/licenses/gpl.html>. There is NO WARRANTY, to the extent permitted by law.

3 bibdoiadd.pl

Add DOI numbers to papers in a given bib file

SYNOPSIS

```
bibdoiadd [-c config_file] [-C 1|0] [-e 1|0] [-f] [-o output] bib_file
```

OPTIONS

-c *config_file*

Configuration file. If this file is absent, some defaults are used. See below for its format.

-C 1|0

Whether to canonicalize names in the output (1) or not (0). By default, 1.

-e

If 1 (default), add empty doi if a doi cannot be found. This prevents repeated searches for the same entries if you add new entries to the file. Calling **-e 0** suppresses this behavior.

-f

Force checking doi number even if one is present

-o *output*

Output file. If this option is not used, the name for the output file is formed by adding `_doi` to the input file

DESCRIPTION

The script reads a BibTeX file. It checks whether the entries have DOIs. If not, it tries to contact <http://www.crossref.org> to get the corresponding DOI. The result is a BibTeX file with the fields `doi=...` added.

The name of the output file is either set by the **-o** option or is derived by adding the suffix `_doi` to the output file.

Every BibTeX record in the input is parsed, using `BibTeX::Parser`, but only the ones that do not have the `doi` field (or `mrnumber` or `zblnumber` for the sibling scripts) are processed. These entries without the requested field are written back, as described in `BibTeX::Parser::Entry`.

The bib records that are not processed (because they already have the requested field) are written back as-is, without any reformatting.

There are (were?) two options for making queries with Crossref: free account and paid membership. In the first case you still must register with Crossref and are limited to a small number of queries, see the agreement at http://www.crossref.org/01company/free_services_agreement.html. In the second case you have a username and password, and can use them for automatic queries. I am not sure whether the use of this script is allowed for the free account holders. At any rate, if you want to add DOIs to a large number of entries, you should register as a paid member.

CONFIGURATION FILE

The configuration file relates to the Crossref queries, and is mostly self-explanatory: it has comments (starting with #) and assignments in the form

```
$field = value ;
```

The important parameters are \$mode ('free' or 'paid'), \$email (for free users) and \$username & \$password for paid members.

EXAMPLES

```
bibdoiadd -c bibdoiadd.cfg -o - citations.bib > result.bib  
bibdoiadd -c bibdoiadd.cfg -o result.bib citations.bib
```

AUTHOR

Boris Veytsman

COPYRIGHT AND LICENSE

Copyright (C) 2014-2024 Boris Veytsman

This is free software. You may redistribute copies of it under the terms of the GNU General Public License <http://www.gnu.org/licenses/gpl.html>. There is NO WARRANTY, to the extent permitted by law.

4 **bibmradd.pl**

Add MR numbers to papers in a given bib file

SYNOPSIS

`bibmradd [-d] [-f] [-e 1|0] [-o output] bib_file`

OPTIONS

-d

Debug mode

-e

If 1 (default), add an empty mrnumber if a mr cannot be found. This prevents repeated searches for the same entries if you add new entries to the file. Calling **-e 0** suppresses this behavior.

-f

Force searching for MR numbers even if the entry already has one.

-o *output*

Output file. If this option is not used, the name for the output file is formed by adding `_mr` to the input file

DESCRIPTION

The script reads a BibTeX file. It checks whether the entries have mrnumbers. If not, it tries to find the numbers from Internet sites. The result is a BibTeX file with `mrnumber=...` fields added.

The name of the output file is either set by the **-o** option or is derived by adding the suffix `_mr` to the output file.

See the `bibdoiadd` script for more details on the processing.

AUTHOR

Boris Veytsman

COPYRIGHT AND LICENSE

Copyright (C) 2014-2024 Boris Veytsman

This is free software. You may redistribute copies of it under the terms of the GNU General Public License <http://www.gnu.org/licenses/gpl.html>. There is NO WARRANTY, to the extent permitted by law.

5 bibzbladd.pl

Add Zbl numbers to papers in a given bib file

SYNOPSIS

```
bibzbladd [-d] [-f] [-e 1|0] [-o output] bib_file
```

OPTIONS

-d

Debug mode

-e

If 1 (default), add an empty zblnumber if a zbl cannot be found. This prevents repeated searches for the same entries if you add new entries to the file. Calling **-e 0** suppresses this behavior.

-f

Force searching for Zbl numbers even if the entry already has one.

-o *output*

Output file. If this option is not used, the name for the output file is formed by adding **_zbl** to the input file

DESCRIPTION

The script reads a BibTeX file. It checks whether the entries have Zbls. If not, it tries to find the numbers from Internet sites. The result is a BibTeX file with **zblnumber=...** fields added.

The name of the output file is either set by the **-o** option or is derived by adding the suffix **_zbl** to the output file.

See the **bibdoiadd** script for more details on the processing.

AUTHOR

Boris Veytsman

COPYRIGHT AND LICENSE

Copyright (C) 2014-2024 Boris Veytsman

This is free software. You may redistribute copies of it under the terms of the GNU General Public License <http://www.gnu.org/licenses/gpl.html>. There is NO WARRANTY, to the extent permitted by law.

6 **biburl2doi.pl**

Convert URLs pointing to doi.org to DOIs

SYNOPSIS

`biburl2doi [-D] [-o output] bib_file`

OPTIONS

-D

Do not delete URLs converted to DOIs

-o *output*

Output file. If this option is not used, the name for the output file is formed by adding `_cleaned` to the input file

DESCRIPTION

The script recognizes URL fields of the kind `http://dx.doi.org` and their variants and converts them to DOI fields.

AUTHOR

Boris Veytsman

COPYRIGHT AND LICENSE

Copyright (C) 2021 Boris Veytsman

This is free software. You may redistribute copies of it under the terms of the GNU General Public License <http://www.gnu.org/licenses/gpl.html>. There is NO WARRANTY, to the extent permitted by law.

7 bbl2bib.pl

Convert thebibliography environment to a bib file

SYNOPSIS

bbl2bib.pl [-d] [-u] [-o *output*] *file*

OPTIONS

[-d]

Send debugging output to stdout

-o *output*

Output file. If this option is not used, the name for the output file is formed by changing the extension to `.bib`

-u

Do not clean URL fields.

Normally `bbl2bib` recognizes URL fields of the kind `http://dx.doi.org` and their variants and converts them to DOI fields (see also `biburl2doi(1)` script). The switch `-u` suppresses this cleanup.

DESCRIPTION

The script tries to reconstruct a `bib` file from the corresponding `thebibliography` environment. One can argue that this operation is akin to reconstructing a cow from the steak. The way the script does it is searching for the entry in the MR database, and creating the corresponding BibTeX fields.

The script reads a TeX or Bbl file and extracts from it the `thebibliography` environment. For each bibitem it creates a plain text bibliography entry, and then tries to match it in the database.

INPUT FILE

We assume some structure of the input file:

1. The bibliography is contained between the lines

```
\begin{thebibliography}...
```

and

```
\end{thebibliography}
```

2. Each bibliography item starts from the line

```
\bibitem[...]{....}
```

EXAMPLES

```
bb12bib -o - file.tex > result.bib
bb12bib -o result.bib file.bbl
bb12bib file.tex
```

AUTHOR

Boris Veytsman

COPYRIGHT AND LICENSE

Copyright (C) 2014-2021 Boris Veytsman

This is free software. You may redistribute copies of it under the terms of the GNU General Public License <http://www.gnu.org/licenses/gpl.html>. There is NO WARRANTY, to the extent permitted by law.

Index

- bb12bib.pl
 - AUTHOR, 13
 - COPYRIGHT AND LICENSE, 13
 - DESCRIPTION, 12
 - EXAMPLES, 13
 - INPUT FILE, 12
 - OPTIONS, 12
 - SYNOPSIS, 12
- bb12bib.pl, 12
- bbdoiadd.pl
 - AUTHOR, 8
 - CONFIGURATION FILE, 8
 - COPYRIGHT AND LICENSE, 8
 - DESCRIPTION, 7
 - EXAMPLES, 8
 - OPTIONS, 7
 - SYNOPSIS, 7
- bbdoiadd.pl, 7
- bbmradd.pl
 - AUTHOR, 9
 - COPYRIGHT AND LICENSE, 9
 - DESCRIPTION, 9
 - OPTIONS, 9
 - SYNOPSIS, 9
- bbmradd.pl, 9
- bburl2doi.pl
 - AUTHOR, 11
 - COPYRIGHT AND LICENSE, 11
 - DESCRIPTION, 11
 - OPTIONS, 11
 - SYNOPSIS, 11
- bburl2doi.pl, 11
- bbzbladd.pl
 - AUTHOR, 10
 - COPYRIGHT AND LICENSE, 10
 - DESCRIPTION, 10
 - OPTIONS, 10
 - SYNOPSIS, 10
- bbzbladd.pl, 10
- ltx2crossrefxml.pl
 - AUTHOR, 5
 - CITATIONS, 5
 - CONFIGURATION FILE FORMAT, 3
 - COPYRIGHT AND LICENSE, 6
 - DESCRIPTION, 2
 - EXAMPLES, 5
 - MORE ABOUT AUTHOR NAMES, 4
 - OPTIONS, 2
 - RPI FILE FORMAT, 3
 - SYNOPSIS, 2
- ltx2crossrefxml.pl, 2