

NAME

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` 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>>) and the TUGboat publication procedure (<<https://tug.org/TUGboat/repository.html>>), but (as always) can also be created by hand or by whatever other method you implement.

Any `.bbl` 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 `pdcn` 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/bibtexperl>).

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 `btldoc` ("BibTeXing" by Oren Patashnik) document for details. The authors are parsed using `BibTeX::Parser::Author` (<https://ctan.org/pkg/bibtexperl>).

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.

Feature request: if anyone is interested in figuring out how to generate structured citations (https://data.crossref.org/reports/help/schema_doc/5.3.1/common5_3_1_xsd.html#citation), that would be great. The schema does not support many useful fields, so we also want to keep the unstructured text output.

Norman Gray's beastie program (<https://heptapod.host/nxg/beastie>) supports this, via `beastie extract-bib.scm -O crossref $(doc).aux`, as invoked in the TUGboat `Common.mak` file. Work in progress.

By the way, if for some reason we have to switch away from using beastie, the most viable approach is probably to change `tugboat.bst` to output no-op TeX commands like `\tubibauthor`, `\tubibtitle`, etc. (a la `biblatex`), and use those commands to discern the various crossref field values. We can't start from the `.bib` because then we'd have to reimplement Bib(La)TeX.

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) 2012–2024 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.