

From BibT_EX to ConT_EXt mkiv

Alan Braslau
CEA, France

1 BibT_EX

“was written by Oren Patashnik, in consultation with Leslie Lamport, to be used with Lamport’s L^AT_EX document preparation system.” (Version 0.98f was released in March 1985; updated 0.99d 8 December 2010.)

“BIBT_EX is a preprocessor (with elements of postprocessing as explained below) for the L^AT_EX document-preparation system. It handles most of the formatting decisions required to produce a reference list, outputting a `.bbl` file that a user can edit to add any finishing touches BIBT_EX isn’t designed to handle (in practice, such editing almost never is needed); with this file L^AT_EX actually produces the reference list.

Here’s how BIBT_EX works. It takes as input (a) an `.aux` file produced by L^AT_EX on an earlier run; (b) a `.bst` file (the style file), which specifies the general reference-list style and specifies how to format individual entries, and which is written by a style designer (called a wizard throughout this program) in a special-purpose language described in the BIBT_EX documentation—see the file `btxdoc.tex`; and (c) `.bib` file(s) constituting a database of all reference-list entries the user might ever hope to use. BIBT_EX chooses from the `.bib` file(s) only those entries specified by the `.aux` file (that is, those given by L^AT_EX’s `\cite` or `\nocite` commands), and creates as output a `.bbl` file containing these entries together with the formatting commands specified by the `.bst` file (BIBT_EX also creates a `.blg` log file, which includes any error or warning messages, but this file isn’t used by any program). L^AT_EX will use the `.bbl` file,

One richness of BIBT_EX is a wealth of `.btx` style perhaps edited by the user, to produce the reference list.”

Why BibT_EX?

- It is a T_EX standard;
- many `.bib` reference database files exist;
- many reference database resources can produce `.bib` files;
- many tools exist to manage `.bib` database files;
- *Why reinvent the wheel?*

What are its limitations?

- 7-bit ASCII
- 16-bit code

1.1 bibtex-x (bibtex8)

“8-bit BibT_EX is an enhanced, portable C version of BibT_EX 0.99. It has been enhanced in these areas:

- conversion to "big" (32-bit) capacity
- capacity selectable at run time
- flexible support for non-English languages using 8-bit character sets
- well matched to L^AT_EX2e and its "inputenc" package

Oren Patashnik, the creator of BibT_EX, is working on a new BibT_EX 1.0 that will be a modern implementation supporting large capacities and non-English languages (see TUGboat, pages 269–274, volume 15, number 3, September 1994). He is content for this version to be released, but hopes that people will eventually migrate to BibT_EX 1.0 when it is released. Its release date is uncertain at the moment.”

1.2 MLBibT_EX

Multi-lingual BibT_EX, a ‘better BibT_EX’
(Jean-Michel Hufflen, University of Franche-Comté).

- design began in October 2000;
- public availability of MLBibT_EX version 1.3 in May 2007;
- some support for ConT_EXt.

1.3 ConT_EXt mkii

(t-bib, since integrated in the core: bibl-tra.mkii and .mkiv)

“The original was developed independently by Taco Hoekwater while still working for Kluwer Academic publishers (it still used the Dutch interface then). Development continued after he left Kluwer, and in January 2005, the then already internationalized file was merged with the core distribution by Hans Hagen. The current version is once again by Taco.”

Relies on the external program `bibtex`.

“Unfortunately, BibT_EX is not the best configurable program around. The names of the commands it parses as well as the `.aux` extension to the file name are both hardwired.

This means ConT_EXt has to write a L^AT_EX-style auxiliary file, yuk! The good news is that it can be rather short. We’ll just ask BibT_EX to output the entire database(s) into the `.bbl` file.”

2 ConT_EXt mkiv

The mkii BibT_EX interface was well maintained, particularly in the period 2006–2009. However, as users requested more than minor corrections to the system as well as new functionality, the need to liberate ConT_EXt from the `bibtex` program became more and more apparent. Hans wanted to move everything to lua, but this was *not* a question of rewriting `bibtex`.

There was good reason, nevertheless, to continue supporting the `.bib` database format, supplemented by other (standard) formats as well as (native) lua tables. Indeed, this becomes simply a question of parsing input to be stored and manipulated internally in the form of lua tables.

There was some discussion about using a real relational database engine. But this was dismissed as the performance of the lua programming is spectacular (and quite flexible).

See, for some test examples, the TUG bibliography archive:

<http://ftp.math.utah.edu/pub/tex/bib/index.html>

“By early 2014, the TUG bibliography archive covers scores of subjects, and hundreds of journals, with more than 925,000 BibTeX entries.”

See also, **Beebe, Nelson H. F. (2009). BibTeX meets relational databases. *TUGboat*, 30(1), 1001-1020.**

“Having them available in multiple SQL databases has made it possible to automate the job of identifying new entries that are candidates for inclusion in author- or subject-specific bibliographies, as well as to run numerous sanity checks on the entire corpus of data. As a result, both the quality and the quantity of the bibliographic data continue to grow.”

The new bibliography system

- multiple database namespaces;
- each namespace can be augmented by multiple source files;
- source files can be of many, mixed formats (.bib, lua, xml, ris, refer, .enw, ...);
- bibliography lists can be placed anywhere in the document;
- rendering of bibliography lists can be fully controlled (standard formats include APA, APS, Vancouver, ...);
- citations can be of many styles (number, authoryear, ...);
- crosslinks are created between citations and list entries;
- indexes of authors and/or keywords can be created;
- **Work in progress!**

Weaknesses:

- does **not** use any BibT_EX .bst bibliography style files (many of which contain L^AT_EX-specific commands);
- does not provide the full functionality of the bibtex program and its special purpose language;
- many existing .bib database files are not strictly exchangeable (since many contain embedded L^AT_EX commands), although the ConT_EXt parser makes effort to handle the most common such commands;
- is **not** yet complete...

3 Datasets

```
\definebtxdataset [standard]  
\usebtxdataset      [standard] [TUGboat.bib]
```

```
@Article{Beebe2009,  
  author   = "Beebe, Nelson H. F.",  
  title    = "BibTeX meets relational databases",  
  journal  = "TUGboat",  
  volume   = "30",  
  number   = "1",  
  pages    = "1001--1020",  
  year     = "2009",  
  ISSN     = "0896-3207",  
  url      = "https://tug.org/tug2009/preprints/beebe.pdf",  
}
```

```
\startbuffer [TUGboat]
@article{Beebe2009,
  author = {Beebe, Nelson H. F.},
  title = {BibTeX meets relational databases},
  journal = {TUGboat},
  volume = {30},
  number = {1},
  pages = {1001--1020},
  year = {2009},
  ISSN = {0896-3207},
  url = {https://tug.org/tug2009/preprints/beebe.pdf}
}
\stopbuffer
\usebtxdataset [standard] [TUGboat.buffer]
```

```
\definebtxdataset [standard]
```

```
\usebtxdataset [standard] [TUGboat.lua]
```

```
return {
```

```
  ["Beebe2009"] = {
```

```
    category = "article"
```

```
    author   = "Beebe, Nelson H. F.",
```

```
    title    = "BibTeX meets relational databases",
```

```
    journal  = "TUGboat",
```

```
    volume   = "30",
```

```
    number   = "1",
```

```
    pages    = "1001--1020",
```

```
    year     = "2009",
```

```
    ISSN     = "0896-3207",
```

```
    url      = "https://tug.org/tug2009/preprints/beebe.pdf",
```

```
  },
```

```
}
```

```
\definebtxdataset [standard]
```

```
\usebtxdataset [standard] [TUGboat.xml]
```

```
<?xml version="2.0" standalone="yes" ? >
```

```
<bibtex>
```

```
  <entry tag="Beebe2009" category="article">
```

```
    <field name="author">Beebe, Nelson H. F.</field>
```

```
    <field name="title">BibTeX meets relational databases</field>
```

```
    <field name="journal">TUGboat</field>
```

```
    <field name="volume">30</field>
```

```
    <field name="number">1</field>
```

```
    <field name="pages">1001--1020</field>
```

```
    <field name="year">2009</field>
```

```
    <field name="ISSN">0896-3207</field>
```

```
    <field name="url">https://tug.org/tug2009/preprints/beebe.pdf</
```

```
  </entry>
```

```
</bibtex>
```

```
\definebtxdataset [standard]  
\usebtxdataset      [standard] [TUGboat.bib]
```

Comments:

- `\definebtxdataset[standard]` is here optional;
- `\usebtxdataset[standard] [TUGboat.bib]` can occur anywhere in the input file (in principle): several passes are required to establish cross-references and the dataset is remembered across runs;
- I find the name `btx` to be somewhat unfortunate (I would prefer ‘bibliography’ or ‘publications’).

```
\definebtxdataset [standard]  
\usebtxdataset [standard] [TUGboat.bib]  
  
\setbtxdataset [standard]
```

This sets the currently active dataset.

(`\setbtxdataset[standard]` is here optional, as it is defined and activated by default.)

Note that most publication-related commands accept (optional) arguments that denote the dataset. Furthermore, references to dataset entries can be prefixed with a dataset identifier (*i.e.* `standard::`).

\showbtxdatasetfields [standard]

tag	category	fields
Beebe2009	article	author index issn journal number pages title url volume year

\showbtxfields [rotation=30]

	<i>article</i>	<i>book</i>	<i>booklet</i>	<i>inbook</i>	<i>incollection</i>	<i>inproceedings</i>	<i>manual</i>	<i>mastersthesis</i>	<i>misc</i>	<i>patent</i>	<i>phdthesis</i>	<i>proceedings</i>	<i>techreport</i>	<i>unpublished</i>
address		*	*	*	*	*	*	*		*	*	*	*	
assignee										*				
author	*		*		*	*	*	*	*	*	*		*	*
author + editor		*		*										
booktitle					*	*								
chapter					*									
chapter + pages				*										
day										*				
dayfiled										*				
edition		*		*	*		*							
editor					*	*						*		
howpublished			*						*					
institution													*	
journal	*													
language										*				
month	*	*	*	*	*	*	*	*	*	*	*	*	*	*
monthfiled										*				
nationality										*				
note	*	*	*	*	*	*	*	*	*	*	*	*	*	*
number	*									*			*	
organization						*	*					*		
pages	*				*	*								
publisher		*		*	*	*						*		
school								*			*			
series		*		*	*	*					*			
title	*	*	*	*	*	*	*	*	*	*	*	*	*	*
type				*	*			*		*	*		*	
volume	*													
volume + number		*		*	*	*						*		
year	*	*	*	*	*	*	*	*	*	*	*	*	*	*
yearfiled										*				

`\showbtxdatasetcompleteness [standard]`

article	Beebe2009
author	Beebe, Nelson H. F.
title	BibTeX meets relational databases
journal	TUGboat
year	2009
volume	30
number	1
pages	1001-1020
+index	1
+issn	0896-3207
+url	https://tug.org/tug2009/preprints/beebe.pdf

4 Publication list rendering

```
\placelistofpublications % \placebtxrendering  
[standard]
```

```
1 Beebe, N.H.F. (2009). BibTeX meets relational databases.  
TUGboat 30, 1001-1020.
```

```
\setupbtxrendering [standard] [numbering=no]  
\placelistofpublications [standard]
```

```
\setupbtxrendering [standard] [alternative=aps]  
\placelistofpublications [standard]
```

Currently, only the alternatives apa (by default) and aps have been programmed; Other standard styles, such as Vancouver (widely used in biomedical publications), will be progressively implemented as the needs arise ... Custom styles can be easily programmed and many standard formatting and processing aids have been put into place.

“One richness of Bib_{T_EX} is a wealth of .bt_x style perhaps edited by the user, to produce the reference list.”

Even though many editors have provided Bib_{T_EX} style definitions, I have yet to find even **one** that accepts Bib_{T_EX} submissions. This has proven to be too complicated and bug-prone for the author+editor production chain, so the .bt_x style files have been produced and distributed only as a guide. This reflects the complications of the task, and the particularities of the specific Bib_{T_EX} syntax.

The importance for most users is the possibility to control the details of the rendering of a publication list. Many stylistic demands exist (see, for example the various universities' requirements for thesis manuscripts), so it is essential that the user be given tools to manage this. Such detail requires some $\text{T}_{\text{E}}\text{X}$ programming skills, but many $\text{ConT}_{\text{E}}\text{Xt}$ macro helpers using

- variables
- texdefinitions
- setups

are provided vastly simplifying this task.

More and more 'standard' renderings will be made available as the system matures.

```
\startsetups btx:apa:article
  \texdefinition{btx:apa:author-or-key-and-year}
  \texdefinition{btx:apa:title-it}
  \texdefinition{btx:apa:journal-volume-issue}
  \texdefinition{btx:apa:pages}
  \texdefinition{btx:apa:doi-or-url}
  \texdefinition{btx:apa:note}
  \texdefinition{btx:apa:comment}
\stopsetups
```

where, for example,

```
\starttexdefinition btx:apa:author-or-key-and-year
  \btxdoifelse {author} {
    \btxflushauthor{author}
  } {
    \btxdoif {key} {
```



```
    \btxlbracket
    \btxsetup{btx:format:key}
    \btxrbracket
  }
}
\btxdoif {year} {
  \btxlparent
  \btxflush{year}
  \btxdoif {suffix} {
    \btxflush{suffix}
  }
  \btxrparent
}
\btxperiod
\stoptexdefinition
```

Hans has translated the logic taken from the `mkii bib` package using `mkiv` tools. Further cleanup and simplifications are necessary in order to move the programming of bibliography styles from the realm of specialists to that of users.

5 Citations

“Scientific knowledge represents the accomplishments of many researchers over time. A critical part of the writing process is helping readers place your contribution in context by citing researchers who influenced you.” (APA Style Guide)

A further, important role of citations and citation indexes is their use as research tools in the creation of knowledge; they can be used to uncover lines of reasoning and to discover original associations of ideas. A statistical analysis of scholarly citations can even itself provide data that can serve to create understanding.

Bibliography lists can orient readers to other influential works either globally, structured by parts, chapters, or sections, or through specific citation references in the text. This is all a question of style.

The APA Style Guide demands that **all** bibliographical references be cited at least once in the text (and that all citations, of course, have a corresponding bibliographical reference).

Whereas this is good practice, other publishing styles, textbooks in particular, might also include a list of general references or for ‘further reading’.

Citation styles fall into two general categories:

1. author, year or
2. numbered.

The bibliographic references often either appear as a bibliography list (located at the end of the work, at the end of the chapter, or the end of the section), or else as footnotes. The footnotes themselves may be placed at the bottom of each page or else be collected for later placement in a list of footnotes.

The present bibliographic system does **not** handle bottom-of-the-page footnote reference lists.

We use the traditional `\cite`:

```
\cite{key}  
\nocite{key}
```

```
\cite[scheme][key]  
\nocite[key]
```

```
\citation [scheme] [key]  
\nocitation [key] % or \usecitation [key]
```

The first pair of commands follow L^AT_EX-style syntax and are **not** encouraged; The second pair respects ConT_EXt-style (where [scheme] is optional). The final pair are synonyms that are more tolerant with respect to spacing.

These commands select a reference from the current dataset (or from any dataset when prefixed, as in `dataset::key`) and place a rendering (number, key, author, date, ...) including a hyperlink; The `\no...` versions select but do not place any rendering. This is useful, for example, when selecting a reference or references at some point in the text where a floating object such as a table or figure is called, whereas the rendering is to appear within the table or in the figure caption. They are synonyms for

`\usecitation [key]`

Thus,

`\cite[Beebe2009]` and `\cite[authoryear][Beebe2009]`

might yield: [1] and (Beebe, 2009).

`\setupbtxcitevariant [alternative=authoryears]`

can be used to set the default `\cite` rendering style, and

`\definebtxcitevariant [altnum] [num] [left=(,right=)]`

yielding: () and

`\definebtxcitevariant [superscript] [num]`
`[left={\raisebox{.7ex}\hbox\bgroup\text},`
`right={\egroup}]`

can be used to fine-tune the rendering.⁽¹⁾

Finally,

```
\definebtxcitevariant % ?
```

can be used to create new alternatives. (*How does one create a full reference or a `\startfootnote\stopfootnote` variant?*)

```
\onlincite[key] % \cite[baseline][key] ?
```

```
\textcite[key] % \cite[authoryearalt][key] ?
```

`\cite` can take a list of keys:

`\cite[Knuth1986,Hobby1992,Hagen2010]`

`\placelistofpublications [more]`

[2-4]

- 2 Knuth, D.E. (1986). *The TeXbook*, volume A of *Computers & Typesetting*. Reading, MA, Addison Wesley
- 3 Hobby, J.D. (1992). A User's manual for MetaPost. Computing Science Technical Report, Murray Hill, New Jersey, AT&T Bell Laboratories
- 4 Hagen, H. (2010). *Metafun*. Hasselt, NL, Boekplan

And references can be combined into a single citation:

```
\cite[Hobby1992+Hagen2010]  
\placelistofpublications [more2]
```

[5]

5 Hobby, J.D. (1992). A User's manual for MetaPost. Computing Science Technical Report, Murray Hill, New Jersey, AT&T Bell Laboratories; Hagen, H. (2010). Metafun. Technical report, Hasselt, NL

(Oops! The second is not a technical report.)

Text before and/or after a reference:

```
\cite[reference=key, lefttext=LEFT, righttext=RIGHT]
    [foo=bar]
```

(as with `\startchapter`, etc.)

Alternatively,

```
\cite[scheme][ "Left text "+key+" right text." ]
```

("" or {}, is this easily parsed?) Of course, `separator=` would need to be suppressed.

```
\cite[scheme][text(Left text )+match(author:Author
and year:YEAR)+text( right text.)]
```

6 Notes, notations and bibliography lists

“Endnotes created with the `\footnote` command are automatically interleaved with the bibliographic references.”

How? Using the note(s) and notation(s) handlers?*

```
\setupnotations [footnote] [location=dataset]
                    % or btxrendering name ?
\setupnotes     [footnote] ...
```

What should be the connections between notes/notations, lists, enumerations and citations/bibliographies?

* ConT_EXt wiki: “There is a bit of terminology mess concerning notes.”

7 Conclusions

This is only the beginning.

Being a system for handling (bibliographic) databases, the tools can be used and abused imaginatively.

For example, I can see creating a database of chemical substances, the twenty proteinogenic amino acids for example, containing names, formulas, schematic representations, data, ... then choosing items according to pertinence. A list of substances selected from this database can be rendered.

Or, one might imagine a database of images. Mojca suggested something like that.