BRAINIACS
BRAIN IMAGING AND COMPUTING SCIENCES

2023 Volume 4
Issue 2 Edoc K562CB81C

BabbleNewt: A Simplified, Consistent, and Interoperable Reference Citation Format for Bibliographic Metadata*

S. Koby Taswell, Aniruddh Anand, Max Montes-Soza, Carl Taswell†

Abstract

Of the diverse bibliographic metadata formats, BibTeX and BibLaTeX have been dominant across mathematics, computing, and engineering due to their use with the TeX and LaTeX typesetting compilers. Despite success in these fields as well as the publishing industry, both BibTeX and BibLaTeX have some deficiencies, notably inconsistencies in the format definitions and use of macros, pseudo-records, programs and processing methods across different software implementations and installations. These inconsistencies contribute to bibliography parsing and document typesetting errors especially problematic with difficult debugging for large bibliography files. A subproject within the PORTAL-DOORS Project (PDP), the BabbleNewt Project aims to address these concerns by designing a set of formats which iterate on the original BibTeX and BibLaTeX formats while enabling easy conversion between them and a newly designed simplified, consistent, and interoperable format called BabbleNewt. The set of related formats implemented for bibliography processors by PDP BabbleNewt includes two formats PdpBibtex and PdpBiblatex corresponding to the original BibTeX and BibLaTeX, two generalized transition formats PdpBibtexgen and PdpBiblatexgen, and the novel format PdpBabblenewt.

Keywords

Bibliographic metadata, interoperability, file formats, PdpBibtex, PdpBibtexgen, PdpBiblatex, PdpBiblatexgen, PdpBabblenewt.

Contents

Introduction

In February 1983, Oren Patashnik began work on BibTeX, "a tool for automating your list of references", intended to accompany LaTeX document typesetting (Patashnik 1998; Patashnik 2003; Fenn 2006). Patashnik's original format BibTeX was accompanied by a parsing utility of the same name, often written in lower case as the command name `bibtex` to run the parser. Since the original development of `bibtex`, various other tools for the format BibTeX have also been implemented including `bibtex8`, `biber`, BibTeXu, CL-dfBibtex, MLBibTeX, and Bibulous. Whereas the original parser `bibtex` supported only 7-bit ASCII characters, `bibtex8` supports 8-bit ASCII characters and BibTeXu supports the UTF-8 character set. Apart from differences in processing character sets, most of the BibTeX parser alternatives have not departed from the original `bibtex` parser intended for the original BibTeX format. In contrast, the parsing tool `biber` was developed for the BibLaTeX format, designed as an extended superset of the BibTeX format (Kime and Wemheuer 2023; Mittelbach 2023). The original BibTeX format has a fixed set of entry types where an entry type declares the type of reference (eg, article, book, etc.) described within the bibliographic metadata record that includes required and optional fields for that entry type such as author, title, publisher, etc. The extended format BibLaTeX improved the usefulness of the format with the addition of many more entry types and metadata fields.

These tools are used throughout mathematics, computing, and engineering fields where LaTeX document typesetting has become the standard expected for publication of manuscripts. Despite how widely these tools are now used throughout these communities, challenges still exist compromising both formats BibTeX and BibLaTeX (Markey 2009; Rees 2017; Mittelbach 2023) in a manner that derives from the original design which lacks the simplicity and consistency of a JSON-style format. Here is a sample record of a *.bib file in the BibTeX format with double quotes for the field values:

```
@article{Patashnik1998bibtex,
  author  = "Oren Patashnik",
  journal = "TUGboat",
  number  = "2",
```

*Presented 2023-10-09 with slides and and video at Guardians 2023

†Authors affiliated with Brain Health Alliance Virtual Institute, Ladera Ranch, CA 92694 USA; correspondence to CTaswell at Brain Health Alliance.

 © 2023 BHA

Table 1: Syntax for PDP BibCitRef Formats with Placeholder Symbols
`Etyp`, `Ekey`, `Anam`, `Aval` for Entity Type and Key, Attribute Name and Value

Format	File Extension	Entity Opener	Attribute Name-Value Pair	Entity Closer	Attribute List
PdpBibtex	*.pbtx	`@Etyp{Ekey,`	`Anam= "Aval",`	`}`	specified
PdpBibtexgen	*.pbtg	`@Etyp{Ekey,`	`Anam= {Aval},`	`}`	unconstrained
PdpBiblatex	*.pblt	`@Etyp{Ekey,`	`Anam= {Aval},`	`}`	specified
PdpBiblatexgen	*.pblg	`@Etyp{Ekey,`	`Anam= [Aval],`	`}`	unconstrained
PdpBabblenewt	*.pbbn	`@{`	`Anam= [Aval],`	`}@`	unconstrained

```
  pages    = "204-207",
  title    = "BIBTEX 101",
  volume   = "19",
  year     = "1998",
}
```

Here is a sample record of a *.bib file in the BibLaTeX format with curly braces for the field values:

```
@article{Patashnik1998bibtex,
  author       = {Oren Patashnik},
  date         = {1998-03-22},
  journaltitle = {TUGboat},
  number       = {2},
  pages        = {204-207},
  title        = {BIBTEX 101},
  volume       = {19},
}
```

The PDP BabbleNewt Project maintains a set of five different but related PDP BibCitRef formats called PdpBibtex, PdpBibtexgen, PdpBiblatex, PdpBiblatexgen, and PdpBabblenewt, intended for use with bibliography file types denoted by the file extensions *.pbtx, *.pbtg, *.pblt, *.pblg, and *.pbbn, respectively. These related formats support both backward and forward compatibility and conversion between a collection of interoperable bibliographic metadata formats. The PdpBibtex and PdpBiblatex formats correspond to the original BibTeX and BibLaTeX formats. The PdpBibtexgen and PdpBiblatexgen formats serve as generalized variant formats for didactic, development, and test purposes. The PdpBabblenewt format provides a simplified, consistent, and interoperable format with a clean separation of data from code that should maximize parsing efficiency, minimize programming errors, and simplify debugging of both parsers and data.

Format Description

The BabbleNewt Project set of PDP BibCitRef formats remain related to each other in a progressive transition to facilitate migration and conversion of bibliography files from one format to another. In a bibliography file for any of these formats, a bibliographic citation record for a bibliographic reference entity consists of an entity opener, a list of attribute name-value pairs, and an entity closer. The related set of formats differ with respect to the syntax required for the entity opener/-closer pair and the list of attribute name-value pairs, also importantly, whether the format specifies the list of attribute pairs or allows the list of attribute pairs to be unconstrained (see Table 1). The formats PdpBibtex and PdpBiblatex specify the lists of entity types and lists of attribute name-value pairs for each entity type, whereas the generalized formats PdpBibtexgen, PdpBiblatexgen, and PdpBabblenewt allow these lists to be unconstrained. Lists of attribute name-value pairs are separated by commas with each name and value in a pair separated by an equal sign.

For the progressive sequence of formats PdpBibtex, PdpBibtexgen, PdpBiblatex, and PdpBiblatexgen, the entity opener/closer pairs remain the same as in the PdpBibtex format for the entity type with entity key and curly braces. In contrast, the entity opener/closer has been simplified and made symmetric for the format PdpBabblenewt with the opener "@{" and closer "}@" to create a more consistent JSON-style scheme for the bibliography file, while also allowing for unconstrained attribute name-value pairs in any order. The entity type and key from the PdpBibtex format have been mapped, respectively, to the attribute name-value pairs with names "referencetype" and "citationkey" in the PdpBabblenewt format.

Moreover for attribute values, the curly brace delimiters in the PdpBibtexgen and PdpBiblatex formats have been changed to square bracket delimiters in the PdpBiblatexgen and PdpBabblenewt formats. Switching from curly braces to square brackets as delimiters for attribute values improves human readability and also improves computer parsing by avoiding the nesting of curly braces, thus simplifying parsing with regular expressions, requirements for escape sequences, and reducing programming errors.

Inspired by a simplified JSON-style approach, the scheme found in the PdpBabblenewt format should permit development of faster parsers with fewer errors. While perhaps not important for small bibliography files with only a few dozen records, error-free efficiency becomes much more important for millions of records in large-scale databases. Indeed, Nurseitov et al. (2009) found that the data processing rates for JSON were much faster and less resource intensive than for XML with parsing of JSON up to 100 times faster than XML.

ReferenceType and CitationKey

The ReferenceType for PDP BibCitRef formats is defined as the type of cited reference such as article, book, report, etc. To describe the reference entity, the ReferenceType determines the list of allowed attibute name-value pairs for the reference entity in those bibliography formats (PdpBibtex and PdpBiblatex) that require them, and corresponds to what has been called the "entry type" in the past. ReferenceTypes, when used with bibliography styles that permissively allow both required and optional attibute name-value pairs for each ReferenceType can be better supported with the generalized and unconstrained bibliography formats (PdpBibtexgen, PdpBiblatexgen, and PdpBabblenewt). For more robust parsing, ReferenceTypes should be considered case-insensitive when processed in algorithms.

Each reference entity record in a bibliography file should always have both a ReferenceType and a CitationKey as a unique identifier to assure disambiguation of references. All PDP BibCitRef formats require, and generate if necessary, a unique CitationKey for each reference entity with a ReferenceType. In general, the CitationKey may be any arbitrary unique character string of arbitrary length. Long identifiers

 © 2023 BHA

for references quickly become inconvenient when typing the source for manuscripts, whereas use of a max-length patterned generator related to the bibliographic metadata for the reference entity provides a consistent mechanism that facilitates easier recognition of CitationKeys. PDP BibCitRef formats generate CitationKeys with a pattern comprised of 3 components with a 16-char-max identifier for provenance (from LastNameFirstAuthor, LastNameEditor, or OrganizationName), an 8-char-max identifier for date (from Date or Year), and an 8-char-max identifier for title (from AcronymFromTitle or WordFromTitle), yielding a CitationKey with a maximum length of 32 characters.

No Macros or Pseudo-Records

A pseudo-record in a bibliography file does not describe a bibliographic reference, but instead provides some other functionality. Pseudo-records can be macros to perform actions such as basic substitution or commands to trigger more complex actions, which interact with other inputs, outputs, or data files. Neither macros nor other kinds of pseudo-records are allowed in PDP BibCitRef bibliography files because the BabbleNewt Project maintains a guiding principle of imposing consistency on the set of related PDP BibCitRef formats in a simplified JSON-like style such that data and code do not mix. This guiding principle implies maintaining a clear boundary between data and code, ie, between the formatted and structured data in data files and the processing algorithms implemented in lexing, parsing, and other utilities of software libraries. Therefore, the PdpBabblenewt format will remain clean with only data and without any code, macros, or pseudo-records. Conversions to the PdpBabblenewt format from other formats requiring expansions of incomplete and/or abbreviated data should be pre-processed with the necessary macro substitutions.

Formatted Record Examples

PdpBibtex (*.pbtx)

```
@article{Patashnik2003BYTT,
  author = "Oren Patashnik",
  journal = "TUGboat",
  title = {BibTeX yesterday, today, and tomorrow},
  volume = "24",
  year = "2003",
}
```

PdpBibtexgen (*.pbtg)

```
@article{Patashnik2003BYTT,
  author = {Oren Patashnik},
  journal = {TUGboat},
  title = {BibTeX yesterday, today, and tomorrow},
  volume = {24},
  year = {2003},
}
```

PdpBiblatex (*.pblt)

```
@article{Patashnik2003BYTT,
  author = {Oren Patashnik},
  date = {2003},
  journaltitle = {TUGboat},
  title = {BibTeX yesterday, today, and tomorrow},
  volume = {24},
}
```

PdpBiblatexgen (*.pblg)

```
@article{Patashnik2003BYTT,
  author = [Oren Patashnik],
  date = [2003],
  journaltitle = [TUGboat],
  title = [BibTeX yesterday, today, and tomorrow],
  volume =[24],
}
```

PdpBabblenewt (*.pbbn)

```
@{
  referencetype = [article],
  citationkey = [Patashnik2003BYTT],
  author = [Oren Patashnik],
  date = [2003],
  journaltitle = [TUGboat],
  title = [BibTeX yesterday, today, and tomorrow],
  volume  =[24],
}@
```

Format Interoperability

Citation Style Language (CSL), developed by Zelle (2015), is an XML-based language for use with citations of references in bibliographies. Similar to BibTeX and BibLaTeX, CSL allows mixing of both code and data in the same file. The BabbleNewt format differs from CSL, BibTeX and BibLaTeX by requiring strict adherence to a data-only principle for the bibilography, thus disallowing macros, commands, styles, pseudo-records or other kinds of code mixed into the data file. As a JSON-like data format, BabbleNewt also differs from CSL implemented as an XML-based language. CSL uses a “CitationKey” but not a “ReferenceType”. Differences between formats for entity-attribute names (aka record field names), such as “ReferenceType” and “CitationKey” regardless of punctuation use and letter casing in the names, can be accommodated by mappings for the related entity attributes when processing transforms from one format to another with import, export, and convert utilities. Thus, the BabbleNewt format is interoperable with CSL and any other bibliographic metadata format including both backward and forward compatibility with the BibTeX and BibLaTeX formats.

The BabbleNewt format maintains adherence to principles for simplifying the format design in order to reduce errors in both data and code, thereby improving reliability and efficiency of processing utilities. To be compatible with requirements for the PrincipalTags of NPDS resource entities (C. Taswell 2007; C. Taswell 2010), and to map a CitationKey for a BibCitRef record to the corresponding PrincipalTag for an NPDS record, use of punctuation symbols such as the hyphen must be avoided in both the attribute name and attribute value. The BabbleNewt format imposes this same requirement on all other attribute names (eg, “ReferenceType” and not “reference-type”) but not on other attribute values for which it would be impractical. This no-punctuation rule for both value and name of an attribute only applies to the CitationKey.

This simplifying rule imposed on the CitationKey implements an important design principle: Avoid use of unnecessary escape symbols, punctuation, and characters that may complicate processing and contribute to additional requirements for more complexity in lexers and parsers. Unnecessary complexity only worsens the probability of coding errors in the software and faulty processing of the data. This simplified design of the BabbleNewt format with a consistent JSON-like style will support more robust lexing and parsing algorithms with greater portability across different programming languages.

Format Performance

Read-write accuracy and efficiency tests were performed on bibiliography files in each of the 5 related formats BibTeX, PdpBibtexgen,

 © 2023 BHA

Table 2: Format Median Round Trip Timing Experiments per Number Records with Lossless Transfer in Seconds

Timing Tests	PdpBibtex	PdpBibtexgen	PdpBiblatex	PdpBiblatexgen	PdpBabblenewt
Initialization	0.37	0.36	0.33	0.40	0.32
8 records	0.38	0.38	0.35	0.42	0.33
80 records	0.61	0.55	0.57	0.61	0.52
800 records	2.62	2.07	2.35	2.36	1.95
8000 records	20.99	16.52	18.90	16.95	14.75

PdpBiblatex, PdpBiblatexgen and PdpBabblenewt corresponding to the same bibliography database of more than 8000 records. The experimental protocol involved several steps: 1) Initialize the BabbleNewt lexer for each format, 2) Measure time for a round-trip cycle of read from import file on disk to record list in memory then write back the records to export file on disk, and 3) Examine and compare export file to import file for any differences in lines or characters. These tests were repeated for varying counts of 8, 80, 800, and 8000 records for each of the 5 bibliography database formats. At all size counts from 8 to 8000, and for all 5 formats, the export files were observed to match the import files exactly with perfect reproducibility. Table 2 summarizes the processing times for these efficiency tests which shows that the PdpBabblenewt format was the most efficient.

Conclusion

The PDP BabbleNewt Project has developed a set of 5 related bibliography database formats called PdpBibtex, PdpBibtexgen, PdpBiblatex, PdpBiblatexgen, and PdpBabblenewt which iterate, extend, and generalize the original BibTeX and BibLaTeX formats while maintaining both backward and forward compatibility as well as supporting progressive transitional migrations between the formats. This set of related formats and the accompanying BabbleNewt lexer have been designed with adherence to the software engineering principle of separating the data files for the bibliographic data from the code files for algorithms implemented in utilities and programs that process the data. Guiding principles for design and implementation for both data formats and processing utilities in the BabbleNewt Project emphasize the concepts of simplicity, consistency, reproducibility, and interoperability with a JSON-like style. Whereas the BabbleNewt Project with its BabbleNewt lexer focuses on processing for interoperability between the set of 5 related formats presented herein, the BabbleBird Project with its BabbleBird parser (S. K. Taswell and C. Taswell 2024) focuses on processing for interoperability between other bibliography database repositories such as IEEE Xplore, NLM PubMed, and Unpaywall, as well as other bibliographic metadata formats such as BIBFRAME, MARC, and RIS (S. K. Taswell, Uhegbu, et al. 2020).

Citation

Brainiacs 2023 Volume 4 Issue 2 Edoc K562CB81C
Title: "BabbleNewt: A Simplified, Consistent, and Interoperable Reference Citation Format for Bibliographic Metadata"
Authors: S. Koby Taswell, Aniruddh Anand, Max Montes-Soza, Carl Taswell
Dates: created 2021-08-19, received 2023-06-28, presented 2023-10-09, updated 2023-12-18, published 2023-12-18, endorsed 2023-12-30
Copyright: © 2023 Brain Health Alliance
Contact: CTaswell at Brain Health Alliance
URL: BrainiacsJournal.org/arc/pub/Taswell2023BBNewt
PDP: /Nexus/Brainiacs/Taswell2023BBNewt
DOI: /10.48085/K562CB81C

References

[1] J. Fenn. "Managing citations and your bibliography with BibTeX." *The PracTEX Journal* 4 (2006) (cited p. 1).

[2] P. Kime and M. Wemheuer. *BibLaTeX — Sophisticated Bibliographies in LaTeX*. Developed and maintained 2006–2012 by Philipp Lehman; 2012–2017 by Philip Kime, Audrey Boruvka, Joseph Wright; 2018–2023 by Philip Kime, Moritz Wemheuer. 2023. URL: https://ctan.org/pkg/biblatex (cited p. 1).

[3] N. Markey. *TameTheBeaST — A manual about bibliographies and especially BibTeX*. Oct. 11, 2009. URL: https://ctan.org/pkg/tamethebeast (visited on 02/27/2022) (cited p. 1).

[4] F. Mittelbach. *The LaTeX companion*. Ed. by U. Fischer. Third edition. Tools and techniques for computer typesetting. Parts I & II. Boston: Addison-Wesley, 2023. ISBN: 013816648X (cited p. 1).

[5] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta. "Comparison of JSON and XML data interchange formats: a case study." *Caine* 9 (2009), pp. 157–162 (cited p. 2).

[6] O. Patashnik. "BIBTEX 101." *TUGboat* 19.2 (Mar. 22, 1998), pp. 204–207 (cited p. 1).

[7] O. Patashnik. "BibTeX yesterday, today, and tomorrow." *TUGboat* 24.1 (2003), pp. 25–30 (cited p. 1).

[8] C. F. Rees. *BibLaTeX/Biber Cheat Sheet*. CTAN, June 24, 2017. URL: https://ctan.org/pkg/biblatex-cheatsheet (visited on 02/27/2022) (cited p. 1).

[9] C. Taswell. "DOORS to the Semantic Web and Grid with a PORTAL for Biomedical Computing." *IEEE Transactions on Information Technology in Biomedicine* 12.2 (2 Mar. 2007). In the Special Section on Bio-Grid published online 3 Aug. 2007, pp. 191–204. ISSN: 1089-7771. DOI: 10.1109/TITB.2007.905861 (cited p. 3).

[10] C. Taswell. "A Distributed Infrastructure for Metadata about Metadata: The HDMM Architectural Style and PORTAL-DOORS System." *Future Internet* 2.2 (2010), pp. 156–189. ISSN: 1999-5903. DOI: 10.3390/FI2020156. URL: https://www.mdpi.com/1999-5903/2/2/156/ (cited p. 3).

[11] S. K. Taswell and C. Taswell. "BabbleBird: A Flexible Software Library for Converting Diverse Bibliographic Formats" (2024). Manuscript in preparation. (cited p. 4).

[12] S. K. Taswell, K. Uhegbu, S. Mashkoor, S. Dutta, and C. Taswell. "Storing bibliographic data in multiple formats with the NPDS cyberinfrastructure." *Proceedings of the Association for Information Science and Technology* 57.1 (Oct. 2020). DOI: 10.1002/pra2.428 (cited p. 4).

[13] R. M. Zelle. *Citation Style Language Primer — An Introduction to CSL*. 2015. URL: https://docs.citationstyles.org/en/stable/primer.html (cited p. 3).