libbibneat

A C++ Library for BibTeX Manipulation

Lorenzo Peri

June 8, 2025

Abstract

libbibneat is a C++ shared library for parsing, transforming, and writing BibTeX files. It powers the bibneat and bibgrab tools, and is designed to be a flexible foundation for your own bibliography-processing projects.

1 Philosophy and Design

The core idea of libbibneat is to provide a modular, composable API for working with BibTeX data. The main container is a BibDB, which holds many BibEntry objects (each representing a single BibTeX entry) via a vector of std::unique_ptr. All transformations and parsing are performed by objects that operate on a BibDB.

2 Core Data Structures

2.1 BibEntry and BibDB

- BibEntry (lib/bibneat/database/bibfile.hpp): Represents a single BibTeX entry. Stores the entry type, key, and all fields as strings. Provides methods to add attributes and determine BibTeX type.
- BibDB (lib/bibneat/database/bibfile.hpp): A container for many BibEntry objects (and special entries). Provides methods to add, clear, and manage entries.

3 API Components

All major API classes take a std::shared_ptr<BibDB> in their constructor and operate on the database in-place.

3.1 Parsing

Header: lib/bibneat/database/parser.hpp

- Parser: Parses BibTeX files (Parser::parse(std::string filename)) or strings (Parser::parseString(std::string bibTex)) and adds entries to a BibDB.

3.2 API Calls

Header: lib/bibneat/recipes/apicalls.hpp

- ApiCaller: Connects to arXiv and doi.org APIs.

- `ApiCaller::checkArXiv(bool replaceWithArxiv, bool replaceWithDOI)` – Checks arXiv entries, can replace with published version.
- `ApiCaller::checkDOI(bool replaceWithDOI)` – Checks DOI entries, can replace with published version.

3.3 Field Filtering

Header: `lib/bibneat/recipes/fieldfilter.hpp`

- `FieldFilter`: Removes fields unknown to BibTeX. Use `FieldFilter::keepBibTex()` to keep only standard fields.

3.4 Unicode and Normalization

Header: `lib/bibneat/uni/fieldnormalization.hpp`

- `FieldNormalizer`: Handles Unicode normalization and LaTeX encoding.
  - `FieldNormalizer::NFCNormalize()` – Canonical normalization.
  - `FieldNormalizer::NFKCNormalize()` – Compatibility normalization.
  - `FieldNormalizer::uni2latex()` – Converts Unicode to LaTeX encoding (NFD decomposition, combining character mapping, special character mapping, NFC composition).
  - `FieldNormalizer::addUTF8Preamble()` – Adds a UTF-8 encoding preamble to the BibDB.

3.5 Printing

Header: `lib/bibneat/database/printer.hpp`

- `Printer`: Writes the BibDB to a .bib file (`Printer::toBibFile(std::string filename, bool overwrite=true)`).

4 Example Usage

```cpp
#include <memory>
#include "lib/bibneat/database/bibfile.hpp"
#include "lib/bibneat/database/parser.hpp"
#include "lib/bibneat/recipes/apicalls.hpp"
#include "lib/bibneat/recipes/fieldfilter.hpp"
#include "lib/bibneat/uni/fieldnormalization.hpp"
#include "lib/bibneat/database/printer.hpp"

int main() {
    // Create a new BibDB (the main bibliography container)
    auto db = std::make_shared<BibDB>();
    // Parse a .bib file and populate the BibDB
    Parser parser(db);
    parser.parse("myrefs.bib");
    // Set up API caller for arXiv/DOI lookups (timeouts in seconds)
    long connTimeout=15L;
    long totTimeout=20L;
    ApiCaller api(db, connTimeout, totTimeout);
    // Check arXiv entries and replace with published version if available
    bool replaceWithArxiv=true;
    bool replaceWithDOI=true;
```

```
    api.checkArXiv(replaceWithArxiv, replaceWithDOI);
    // Remove non-standard BibTeX fields
    FieldFilter filter(db);
    filter.keepBibTex();
    // Normalize Unicode and convert to LaTeX encoding
    FieldNormalizer normalizer(db);
    normalizer.NFCNormalize();
    normalizer.uni2latex();
    // Add UTF-8 encoding preamble
    normalizer.addUTF8Preamble();
    // Write the cleaned bibliography to a file
    Printer printer(db);
    printer.toBibFile("cleaned.bib");
}
```

5 Extending libbibneat

The API is designed to be modular and extensible. You can add your own transformation classes that operate on a `BibDB`, or use the provided utilities for string and Unicode handling (see `lib/bibneat/utils/`).

6 License

`libbibneat` is open source and distributed under the GLP3 License. Have fun!