# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Info</td>
<td>3</td>
</tr>
<tr>
<td>2 Install</td>
<td>5</td>
</tr>
<tr>
<td>3 For Devs</td>
<td>7</td>
</tr>
<tr>
<td>3.1 API's Main Namespaces</td>
<td>7</td>
</tr>
<tr>
<td>3.2 Merge Metadata</td>
<td>8</td>
</tr>
<tr>
<td>3.3 A full featured app!</td>
<td>9</td>
</tr>
<tr>
<td>4 Known Issues</td>
<td>11</td>
</tr>
<tr>
<td>5 ISBN</td>
<td>13</td>
</tr>
<tr>
<td>6 Code</td>
<td>15</td>
</tr>
<tr>
<td>6.1 Search</td>
<td>15</td>
</tr>
<tr>
<td>6.2 Status</td>
<td>15</td>
</tr>
<tr>
<td>6.3 How to Contribute</td>
<td>15</td>
</tr>
</tbody>
</table>
isbnlib provides several useful methods and functions to validate, clean, transform, hyphenate and get metadata for ISBN strings.

Contents:
Typical usage (as library):

```python
#!/usr/bin/env python
import isbnlib
...
```

Just for fun, suppose I want the most spoken about book with certain words in his title. For a quick-and-dirty solution, enter the following code in a file and save it as isbn_tmsa_book.py.

```python
#!/usr/bin/env python
import sys
from isbnlib import *

query = sys.argv[1].replace(' ', '+')
isbn = isbn_from_words(query)

print("The ISBN of the most 'spoken-about' book with this title is $s" % isbn)
print("
print("... and the book is:")
print("
print((meta(isbn)))
```

Then in a command line (in the same directory):

```
$ python isbn_tmsa_book.py 'noise'
```

In my case I get:

The ISBN of the most 'spoken-about' book with this title is 9780143105985

... and the book is:

```
```

Have fun!
CHAPTER 2

Install

From the command line enter (in some cases you have to preced the command with `sudo`):

```bash
$ pip install isbnlib
```

or:

```bash
$ easy_install isbnlib
```

or:

```bash
$ pip install isbnlib-3.4.5.tar.gz
```

(first you have to download the file!)
For Devs

3.1 API’s Main Namespaces

In the namespace isbnlib you have access to the core methods:

- **is_isbn10(isbn10like)** Validate as ISBN-10.
- **to_isbn10(isbn13)** Transform isbn-13 to isbn-10.
- **to_isbn13(isbn10)** Transform isbn-10 to isbn-13.
- **canonical(isbnlike)** Keep only numbers and X. You will get strings like 9780321534965.
- **clean(isbnlike)** Clean ISBN (only legal characters).
- **notisbn(isbnlike, level=’strict’)** Check with the goal to invalidate isbn-like.
- **get_isbnlike(text, level=’normal’)** Extract all substrings that seem like ISBNs (very useful for scraping).
- **get_canonical_isbn(isbnlike, output=’bouth’)** Extract ISBNs and transform them to the canonical form.
- **EAN13(isbnlike)** Transform an isbnlike string into an EAN13 number (validated canonical ISBN-13).
- **info(isbn)** Get language or country assigned to this ISBN.
- **mask(isbn, separator=’-’)** Mask (hyphenate) a canonical ISBN.
- **meta(isbn, service=’default’, cache=’default’)** Gives you the main metadata associated with the ISBN. As service parameter you can use: ‘wcat’ uses worldcat.org (no key is needed), ‘goob’ uses the Google Books service (no key is needed), ‘isbndb’ uses the isbndb.com service (an api key is needed), ‘openl’ uses the OpenLibrary.org api (no key is needed), merge uses a merged record of wcat and goob records (no key is needed) and is the default option. You can get an API key for the isbndb.com service here_._ You can enter API keys with isbnlib.config.add_apikey(service, apikey). The output can be formatted as bibtex, msword, endnote, refworks, opf or json (BibJSON) bibliographic formats with isbnlib.registry.bibformatters.cache only allows two values: ‘default’ or None. You can change the kind of cache by using isbnlib.registry.set_cache (see below).
- **editions(isbn)** Return the list of ISBNs of editions related with this ISBN.
- **isbn_from_words(words)** Return the most probable ISBN from a list of words (for your geographic area).
- **goom(words)** Return a list of references from Google Books multiple references.
The exceptions raised by these methods can all be caught using ISBNLibException. You should\’t raise this exception in your code, only raise the specific exceptions exposed in isbnlib.dev whose name end in Error.

In isbnlib.dev.helpers you can find several methods, that we found very useful, some of then are only used in isbntools (an app and framework that uses \"isbnlib\")

With isbnlib.registry you can change the metadata service to be used by default (setdefaultservice), add a new service (add_service), access bibliographic formatters for metadata (bibformatters), set the default formatter (setdefaultbibformatter), add new formatters (add_bibformatter) and set a new cache (set_cache) (e.g. to switch off the cache set_cache(None)). The cache only works for calls through isbnlib.meta. These changes only work for the \‘current session\’, so should be done always before calling other methods.

Finally, from isbnlib.config you can read and set configuration options: change timeouts with setsocketstimeout and setthreadstimeout, access api keys with apikeys and add new one with add_apikey and access and set generic and user-defined options with options and set_option.

All these classes follow a simple design pattern and, if you follow it, will be very easy to integrate your classes with the rest of the lib.

### 3.2 Merge Metadata

The original quality of metadata, at the several services, is not very good! If you need high quality metadata in your app, the only solution is to use polling & merge of several providers and a lot of cleaning and standardization for fields like Authors and Publisher. A simple merge provider is now the default in isbnlib.meta. It gives priority to wcat but overwrites the Authors field with the value from goob. Uses the merge method of Metadata and serial calls to services by default (faster for fast Internet connections).

You can change that by using vias\’s other methods (e.g. isbnlib.config.set_option(\'VIAS_MERGE\’, \’multi\’)).

You can write your own merging scheme by creating a new provider (see merge for an example).

**Take Note**: These classes are optimized for one-calls to services and not for batch calls.
3.3 A full featured app!

If you want a full featured app, that uses isbnlib, with end user apps, configuration files and a framework to further development, take a look at isbntools.

You can browse the code, in a very structured way, at sourcegraph or GitHub.
Known Issues

1. The `meta` method sometimes give a wrong result (this is due to errors on the chosen service), in alternative you could try one of the others services.

2. The `isbnlib` works internally with unicode, however this doesn’t solve errors of lost information due to bad encode/decode at the origin!

3. Periodically, agencies, issue new blocks of ISBNs. The `range` of these blocks is on a database that `mask` uses. So it could happen, if you have a version of `isbnlib` that is too old, `mask` doesn’t work for valid (recent) issued ISBNs. The solution? **Update isbnlib often!**

4. Calls to metadata services are cached by default. You can change that by calling `isbnlib.meta(isbn, service='default', cache=None)`

Any issue that you would like to report, please do it at [github](https://github.com) or post a question on [stackoverflow](https://stackoverflow.com) (with tag `isbnlib`).
ISBN

To know about ISBN:

- http://www.isbn-international.org/
6.1 Search

Search/Browse the code at sourcegraph or github

6.2 Status

6.3 How to Contribute

isbnlib has a very small code base, so it is a good project to begin your adventure in open-source...

6.3.1 Main Steps

1. Make sure you have a GitHub account
2. Submit a ticket for your issue or idea, on GitHub issues (if possible wait for some feedback before any serious commitment... :)
3. Fork the repository on GitHub
4. `pip install -r requirements-dev.txt`
5. Do your code... (remember the code must run on python 2.6, 2.7, 3.3, 3.4, pypy and be OS independent) (you will find travis-ci.org very handy for this!)
6. Write tests for your code using nose and put then in the directory isbnlib/test
7. Pass all tests and with coverage > 90%. Check the coverage in Coveralls.
8. Check if all requirements are fulfilled!
9. Make a pull request on github...
6.3.2 Important

If you don’t have experience in these issues, don’t be put off by these requirements, see them as a learning opportunity. Thanks!

  For full instructions read the CONTRIBUTING doc.