



Podręcznik użytkownika calibre

Wydanie 7.9.0

Kovid Goyal

kwietnia 19, 2024

Spis treści

1	Interfejs użytkownika	3
2	Dodawanie ulubionej strony z wiadomościami	31
3	The E-book viewer	51
4	Konwersja książek	59
5	Edycja e-booków	79
6	Serwer zawartości calibre	113
7	Porównanie e-booków	121
8	Edycja metadanych e-booków	125
9	Często zadawane pytania	129
10	Samouczki	153
11	Schemat adresu URL calibre://	251
12	Dostosowywanie calibre	255
13	Linia poleceń	303
14	Konfigurowanie środowiska programistycznego dla calibre	345
15	Zarządzanie prawami cyfrowymi (DRM)	375
16	Słownik	379
	Indeks modułów Pythona	381
	Indeks	383

Calibre jest menedżerem bibliotek e-booków. Może przeglądać, konwertować i katalogować książki elektroniczne w większości głównych formatów książek elektronicznych. Może także komunikować się z wieloma czytnikami e-booków. Może wyjść do Internetu i pobrać metadane do książek. Może pobierać gazety i konwertować je na e-książki w celu wygodnego czytania. Jest wieloplatformowy, działa na systemach Linux, Windows i macOS.

Właśnie uruchomiłeś calibre. Co zrobić teraz? Zanim calibre będzie mogło zrobić cokolwiek z twoimi e-bookami, najpierw musi o nich wiedzieć. Przeciągnij kilka e-booków do okna calibre lub kliknij przycisk „Dodaj książki” i znajdź e-booki z którymi chcesz pracować. Gdy dodasz książki, wyświetlą się w głównym widoku mniej więcej w ten sposób:

110	The Trouble With Physics	Lee Smolin	18 Mar 2011	0.9	★★★★★
111	The Wise Man's Fear	Patrick Rothfuss	08 Mar 2011	1.4	★★★★★
112	The Heroes	Joe Abercrombie	08 Mar 2011	1.2	★★★

Po podziwianiu listy książek, które właśnie dodałeś do treści swojego serca, prawdopodobnie będziesz chciał je przeczytać. W tym celu musisz przekonwertować książkę na format zrozumiały dla czytnika. Przy pierwszym uruchomieniu kalibru uruchamia się: guilabel: `Welcome Wizard «i ustawia kaliber dla twojego czytnika. Konwersja to pestka. Po prostu wybierz książkę, którą chcesz przekonwertować, a następnie kliknij przycisk „Konwertuj książki”. Zignoruj na razie wszystkie opcje i kliknij „OK”. Mała ikona w prawym dolnym rogu zacznie się obracać. Po zakończeniu wirowania przekonwertowana książka jest gotowa. Kliknij przycisk „Widok”, aby przeczytać książkę.

Jeśli chcesz przeczytać książkę na swoim czytniku, podłącz czytnik do komputera, poczekaj, aż kaliber ją wykryje (10-20 sekund), a następnie kliknij przycisk „Wyślij do urządzenia”. Gdy ikona ponownie przestanie się obracać, odłącz czytnik i czytaj dalej! Jeśli nie przekonwertowałeś książki w poprzednim kroku, Calibre automatycznie skonwertuje ją do formatu obsługiwanego przez Twoje urządzenie czytnika.

Aby dokładniej zapoznać się z programem powinieneś przeczytać *Interfejs użytkownika* (strona 3). By uzyskać więcej informacji o możliwościach calibre zapoznaj się z *Linia poleceń* (strona 303). Wiele użytecznych informacji znajdziesz również w *Często zadawane pytania* (strona 129).

Jeśli masz więcej pytań, chcesz podyskutować o calibre z innymi użytkownikami albo poprosić o pomoc z określoną rzeczą, są :website: fora oraz inne zasoby wiedzy dostępne<help>`.

Sekcje

Interfejs użytkownika

Graficzny interfejs użytkownika (*GUI*) zapewnia dostęp do wszystkich funkcji zarządzania biblioteką i konwersji formatu e-booków. Podstawowym przebiegiem pracy przy użyciu calibre jest najpierw dodanie książek do biblioteki z dysku twardego. calibre automatycznie spróbuje odczytać metadane z książek i dodać je do swojej wewnętrznej bazy danych. Gdy już znajdują się w bazie danych, możesz wykonywać na nich różne akcje, które obejmują konwersję z jednego formatu na inny, przesyłanie na urządzenie odczytujące, przeglądanie na komputerze i edycję metadanych. Ta ostatnia obejmuje między innymi modyfikację okładki, opisu i tagów. Pamiętaj, że calibre tworzy kopie plików, które do niego dodajesz. Twoje oryginalne pliki pozostają nietknięte.

Interfejs jest podzielony na różne sekcje:

- *Działania* (strona 4)
- *Ustawienia* (strona 10)
- *Katalogi* (strona 11)
- *Search & sort* (strona 11)
- *The search interface* (strona 12)
- *Zapisywanie wyszukiwań* (strona 17)
- *Searching the full text of all books* (strona 18)
- *Wirtualne biblioteki* (strona 18)
- *Temporarily marking books* (strona 18)
- *Guessing metadata from file names* (strona 18)
- *Szczegóły książki* (strona 20)
- *Przeglądarka etykiet* (strona 22)
- *Widok okładek* (strona 24)
- *Przeglądarka okładek* (strona 25)

- *Adding notes for authors, series, etc.* (strona 26)
- *Podgląd* (strona 27)
- *Zadania* (strona 28)
- *Skróty klawiszowe* (strona 28)

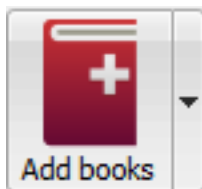
1.1 Działania



Pasek narzędzi akcji zapewnia wygodne skróty do często używanych akcji. Jeśli klikniesz prawym przyciskiem myszy przyciski, możesz wykonać wariacje na temat domyślnej akcji. Pamiętaj, że pasek narzędzi działań będzie wyglądał nieco inaczej w zależności od tego, czy masz podłączony do komputera czytnik e-booków.

- *Dodaj książki* (strona 4)
- *Edytuj metadane* (strona 5)
- *Konwertuj książki* (strona 6)
- *Wyświetl* (strona 6)
- *Prześlij do urządzenia* (strona 7)
- *Pobierz wiadomości* (strona 7)
- *Biblioteka* (strona 8)
- *Urządzenie* (strona 8)
- *Zapisz na dysku* (strona 9)
- *Połącz/uodostępnij* (strona 9)
- *Usuń książki* (strona 10)

1.1.1 Dodaj książki



Przycisk *Dodaj książki* kryje siedem różnych wersji, dostępnych po kliknięciu prawym klawiszem.

1. **Add books from a single folder:** Opens a file chooser dialog and allows you to specify which books in a folder should be added. This action is *context sensitive*, i.e. it depends on which *catalog* (strona 11) you have selected. If you have selected the *Library*, books will be added to the library. If you have selected the e-book reader device, the books will be uploaded to the device, and so on.
2. **Add books from folders and sub-folders:** Allows you to choose a folder. The folder and all its sub-folders are scanned recursively, and any e-books found are added to the library. You can choose whether to have calibre add

all files present in a single folder to a single book record or multiple book records. calibre assumes that each folder contains a single book. All e-book files in a folder are assumed to be the same book in different formats. This action is the inverse of the [Save to disk](#) (strona 9) action, i.e. you can *Save to disk*, delete the books and re-add them in single book per folder mode, with no lost information except for the date (this assumes you have not changed any of the setting for the Save to disk action).

3. **Dodaj wiele książek z archiwum (ZIP/RAR):** umożliwia dodawanie wielu e-booków przechowywanych w wybranych plikach ZIP lub RAR. Jest to wygodny skrót, który pozwala uniknąć konieczności rozpakowywania archiwum, a następnie dodawania książek za pomocą jednej z dwóch powyższych opcji.
4. **Dodaj pustą książkę (wpis książki bez formatów):** umożliwia utworzenie pustego rekordu książki. Można to wykorzystać do ręcznego wypełnienia informacji o książce, której być może nie masz jeszcze w swojej kolekcji.
5. **** Dodaj z ISBN **:** Pozwala na dodanie jednej lub więcej książek, wpisując ich numery ISBN.
6. **** Dodaj pliki do wybranych wpisów **:** Pozwala dodać lub zaktualizować pliki związane z istniejącą książką w bibliotece.
7. **Add data files to selected book records:** Allows you to add any number of extra files that will be stored in a data sub-directory in the book directory. See [Adding extra data files to a book](#) (strona 128) for details.
8. **Dodaj pusty plik do wybranych rekordów książki:** Umożliwia dodanie pustego pliku o określonym formacie do wybranych rekordów książki.

Akcja *Dodaj książki* może odczytywać metadane z wielu różnych formatów e-booków. Ponadto próbuje odgadnąć metadane z nazwy pliku. Zobacz sekcję [Guessing metadata from file names](#) (strona 18), aby dowiedzieć się, jak to skonfigurować.

Aby dodać dodatkowy format do istniejącej książki możesz zrobić jedną z trzech rzeczy:

1. Przeciągnij i upuść plik na panel Szczegóły książki po prawej stronie okna głównego
2. Kliknij prawym przyciskiem myszy przycisk *Dodaj książki* i wybierz *Dodaj pliki do wybranych książek*.
3. Kliknij przycisk *Dodaj książki* w prawym górnym obszarze okna dialogowego *Edytuj metadane*, do którego dostęp uzyskuje się za pomocą akcji [Edytuj metadane](#) (strona 5).

1.1.2 Edytuj metadane

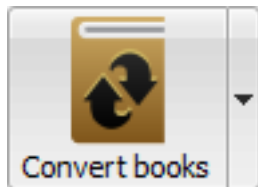


The *Edit metadata* action has four variations which can be accessed by doing a right-click on the button.

1. **Edit metadata individually:** Allows you to edit the metadata of books one-by-one with the option of fetching metadata, including covers, from the Internet. It also allows you to add or remove particular e-book formats from a book.
2. **Edit metadata in bulk:** Allows you to edit common metadata fields for large numbers of books simultaneously. It operates on all the books you have selected in the [Library view](#) (strona 11).
3. **Download metadata and covers:** Downloads metadata and covers (if available) for the books that are selected in the book list.
4. **Merge book records:** Gives you the capability of merging the metadata and formats of two or more book records. You can choose to either delete or keep the records that were not clicked first.
5. **Manage data files:** Manage the extra data files associated with the selected books.

For more details, see *Edycja metadanych e-booków* (strona 125).

1.1.3 Konwertuj książki



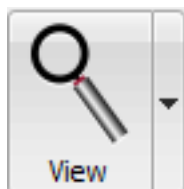
E-books can be converted from a number of formats into whatever format your e-book reader prefers. Many e-books available for purchase will be protected by *Digital Rights Management* (strona 375) (*DRM*) technology. calibre will not convert these e-books. It is easy to remove the DRM from many formats, but as this may be illegal, you will have to find tools to liberate your books yourself and then use calibre to convert them.

Dla większości ludzi konwersja powinna być prosta jak bułka z masłem. Jeśli chcesz dowiedzieć się więcej o procesie konwersji, patrz konwersja.

Akcja *Konwertuj książki* ma trzy warianty, dostępne po naciśnięciu prawego przycisku myszy.

1. **Convert individually:** Allows you to specify conversion options to customize the conversion of each selected e-book.
2. **Bulk convert:** Allows you to specify options only once to convert a number of e-books in bulk.
3. **Create a catalog of the books in your calibre library:** Allows you to generate a complete listing of the books in your library, including all metadata, in several formats such as XML, CSV, BiBTeX, EPUB and MOBI. The catalog will contain all the books currently showing in the library view. This allows you to use the search features to limit the books to be catalogued. In addition, if you select multiple books using the mouse, only those books will be added to the catalog. If you generate the catalog in an e-book format such as EPUB, MOBI or AZW3, the next time you connect your e-book reader the catalog will be automatically sent to the device. For more information on how catalogs work, read the *Tworzenie katalogów AZW3, EPUB, MOBI* (strona 242).

1.1.4 Wyświetl



The *View* action displays the book in an e-book viewer program. calibre has a built-in viewer for many e-book formats. For other formats it uses the default operating system application. You can configure which formats should open with the internal viewer via *Preferences* → *Interface* → *Behavior*. If a book has more than one format, you can view a particular format by doing a right-click on the button.

1.1.5 Prześlij do urządzenia



The *Send to device* action has eight variations, accessed by doing a right-click on the button.

1. **Send to main memory:** The selected books are transferred to the main memory of the e-book reader.
2. **Send to card (A):** The selected books are transferred to the storage card (A) on the e-book reader.
3. **Send to card (B):** The selected books are transferred to the storage card (B) on the e-book reader.
4. **Send specific format to:** The selected books are transferred to the selected storage location on the device, in the format that you specify.
5. **Eject device:** Detaches the device from calibre.
6. **Set default send to device action:** Allows you to specify which of the options, 1 through 5 above or 7 below, will be the default action when you click the main button.
7. **Send and delete from library:** The selected books are transferred to the selected storage location on the device and then **deleted** from the Library.
8. **Fetch Annotations (experimental):** Transfers annotations you may have made on an e-book on your device to the comments metadata of the book in the calibre library.

You can control the file name and folder structure of files sent to the device by setting up a template in *Preferences* → *Import/export* → *Sending books to devices*. Also see *Język szablonów calibre* (strona 161).

1.1.6 Pobierz wiadomości



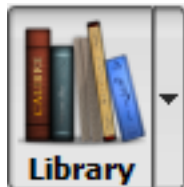
The *Fetch news* action downloads news from various websites and converts it into an e-book that can be read on your e-book reader. Normally, the newly created e-book is added to your e-book library, but if an e-book reader is connected at the time the download finishes, the news is also uploaded to the reader automatically.

The *Fetch news* action uses simple recipes (10-15 lines of code) for each news site. To learn how to create recipes for your own news sources, see *Dodawanie ulubionej strony z wiadomościami* (strona 31).

The *Fetch news* action has three variations, accessed by doing a right-click on the button.

1. **Schedule news download:** Allows you to schedule the download of your selected news sources from a list of hundreds available. Scheduling can be set individually for each news source you select and the scheduling is flexible allowing you to select specific days of the week or a frequency of days between downloads.
2. **Add a custom news source:** Allows you to create a simple recipe for downloading news from a custom news site that you wish to access. Creating the recipe can be as simple as specifying an RSS news feed URL, or you can be more prescriptive by creating Python-based code for the task. For more information, see *Dodawanie ulubionej strony z wiadomościami* (strona 31).
3. **Download all scheduled news sources:** Causes calibre to immediately begin downloading all news sources that you have scheduled.

1.1.7 Biblioteka



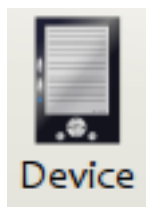
The *Library* action allows you to create, switch between, rename or remove a Library. calibre allows you to create as many libraries as you wish. You could, for instance, create a fiction library, a non-fiction library, a foreign language library, a project library, or any structure that suits your needs. Libraries are the highest organizational structure within calibre. Each library has its own set of books, tags, categories and base storage location.

1. **Switch/create library...:** Allows you to; a) connect to a pre-existing calibre library at another location, b) create an empty library at a new location or, c) move the current library to a newly specified location.
2. **Quick switch:** Allows you to switch between libraries that have been registered or created within calibre.
3. **Rename library:** Allows you to rename a Library.
4. **Pick a random book:** Chooses a random book in the library for you
5. **Remove library:** Allows you to unregister a library from calibre.
6. **Export/import all calibre data:** Allows you to either export calibre data for migration to a new computer or import previously exported data.
7. **<library name>:** Actions 7, 8 etc... give you immediate switch access between multiple libraries that you have created or attached to. This list contains only the 5 most frequently used libraries. For the complete list, use the Quick Switch menu.
8. **Library maintenance:** Allows you to check the current library for data consistency issues and restore the current library's database from backups.

Informacja: Metadata about your e-books, e.g. title, author, and tags, is stored in a single file in your calibre library folder called metadata.db. If this file gets corrupted (a very rare event), you can lose the metadata. Fortunately, calibre automatically backs up the metadata for every individual book in the book's folder as an OPF file. By using the Restore database action under Library Maintenance described above, you can have calibre rebuild the metadata.db file from the individual OPF files for you.

You can copy or move books between different libraries (once you have more than one library setup) by right clicking on the book and selecting the action *Copy to library*.

1.1.8 Urządzenie



The *Device* action allows you to view the books in the main memory or storage cards of your device, or to eject the device (detach it from calibre). This icon shows up automatically on the main calibre toolbar when you connect a supported device. You can click on it to see the books on your device. You can also drag and drop books from your calibre library onto the icon to transfer them to your device. Conversely, you can drag and drop books from your device onto the library icon on the toolbar to transfer books from your device to the calibre library.

1.1.9 Zapisz na dysku



The *Save to disk* action has five variations, accessed by doing a right-click on the button.

1. **Save to disk:** Saves the selected books to disk organized in folders. The folder structure looks like:

```
Author_(sort)
  Title
    Book Files
```

You can control the file name and folder structure of files saved to disk by setting up a template in *Preferences* → *Import/export* → *Saving books to disk*. Also see *Język szablonów calibre* (strona 161).

2. **Save to disk in a single folder:** Saves the selected books to disk in a single folder.

For 1. and 2., all available formats, as well as metadata, are stored to disk for each selected book. Metadata is stored in an OPF file. Saved books can be re-imported to the library without any loss of information by using the *Add books* (strona 4) action.

3. **Save only *<your preferred>* format to disk:** Saves the selected books to disk in the folder structure as shown in (1.) but only in your preferred e-book format. You can set your preferred format in *Preferences* → *Interface* → *Behaviour* → *Preferred output format*
4. **Save only *<your preferred>* format to disk in a single folder:** Saves the selected books to disk in a single folder but only in your preferred e-book format. You can set your preferred format in *Preferences* → *Interface* → *Behaviour* → *Preferred output format*
5. **Save single format to disk...:** Saves the selected books to disk in the folder structure as shown in (1.) but only in the format you select from the popup list.

1.1.10 Połącz/udostępnij



The *Connect/share* action allows you to manually connect to a device or folder on your computer. It also allows you to set up your calibre library for access via a web browser or email.

The *Connect/share* action has four variations, accessed by doing a right-click on the button.

1. **Connect to folder:** Allows you to connect to any folder on your computer as though it were a device and use all the facilities calibre has for devices with that folder. Useful if your device cannot be supported by calibre but is available as a USB disk.
2. **Start Content server:** Starts calibre's built-in web server. When started, your calibre library will be accessible via a web browser from the Internet (if you choose). You can configure how the web server is accessed by setting preferences at *Preferences* → *Sharing* → *Sharing over the net*
3. **Setup email based sharing of books:** Allows sharing of books and news feeds by email. After setting up email addresses for this option, calibre will send news updates and book updates to the entered email addresses. You can configure how calibre sends email by setting preferences at

Preferences → *Sharing* → *Sharing books by email*. Once you have set up one or more email addresses, this menu entry will be replaced by menu entries to send books to the configured email addresses.

1.1.11 Usuń książki

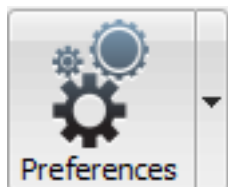


The *Remove books* action **deletes books permanently**, so use it with care. It is *context sensitive*, i.e. it depends on which *catalog* (strona 11) you have selected. If you have selected the *Library*, books will be removed from the library. If you have selected the e-book reader device, books will be removed from the device. To remove only a particular format for a given book use the *Edytuj metadane* (strona 5) action. Remove books also has five variations which can be accessed by doing a right-click on the button.

1. **Remove selected books:** Allows you to **permanently** remove all books that are selected in the book list.
2. **Remove files of a specific format from selected books...:** Allows you to **permanently** remove e-book files of a specified format from books that are selected in the book list.
3. **Remove all formats from selected books, except...:** Allows you to **permanently** remove e-book files of any format except a specified format from books that are selected in the book list.
4. **Remove all formats from selected books:** Allows you to **permanently** remove all e-book files from books that are selected in the book list. Only the metadata will remain.
5. **Remove covers from selected books:** Allows you to **permanently** remove cover image files from books that are selected in the book list.
6. **Remove matching books from device:** Allows you to remove e-book files from a connected device that match the books that are selected in the book list.
7. **Restore recently deleted:** Allows you to undo the removal of books or formats.

Informacja: Note that when you use *Remove books* to delete books from your calibre library, the book record is deleted, but the books are temporarily stored, for a few days, in a trash folder. You can undo the delete by right clicking the *Remove books* button and choosing to *Restore recently deleted* books.

1.2 Ustawienia

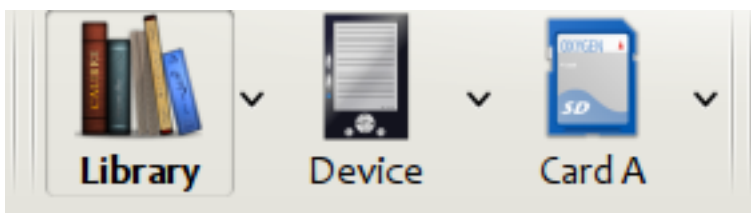


The *Preferences* action allows you to change the way various aspects of calibre work. It has four variations, accessed by doing a right-click on the button.

1. **Preferences:** Allows you to change the way various aspects of calibre work. Clicking the button also performs this action.
2. **Run Welcome wizard:** Allows you to start the *Welcome wizard* which appeared the first time you started calibre.

3. **Get plugins to enhance calibre:** Opens a new window that shows plugins for calibre. These plugins are developed by third parties to extend calibre's functionality.
4. **Restart in debug mode:** Allows you to enable a debugging mode that can assist the calibre developers in solving problems you encounter with the program. For most users this should remain disabled unless instructed by a developer to enable it.

1.3 Katalogi



A *catalog* is a collection of books. calibre can manage two types of different catalogs:

1. **Library:** This is a collection of books stored in your calibre library on your computer.
2. **Device:** This is a collection of books stored in your e-book reader. It will be available when you connect the reader to your computer.

Many operations, such as adding books, deleting, viewing, etc., are context sensitive. So, for example, if you click the *View* button when you have the **Device** catalog selected, calibre will open the files on the device to view. If you have the **Library** catalog selected, files in your calibre library will be opened instead.

1.4 Search & sort

Search: <input type="text"/>								
	Title	Author(s)	Size (MB)	Date	Rating ▾	Publisher	Tags	Series
1	The Complete Works of William Shakespeare	William Shakespeare	2.4	02 Jan 2007	★★★★★	manybooks.net		
2	Stalky and Co.	Rudyard Kipling	0.2	19 Jan 2007	★★★★★	manybooks.net		
3	The Comedies of William Shakespeare	William Shakespeare	2.1	15 Mar 2007	★★★★★			
4	The Histories of William Shakespeare	William Shakespeare	1.5	15 Mar 2007	★★★★★		england, historical fiction	
5	The Tragedies of William Shakespeare	William Shakespeare	1.6	15 Mar 2007	★★★★★			
6	War and Peace	Leo Tolstoy	3.1	22 Aug 2007	★★★★★	gutenberg.org	classic	
7	Anna Karenina	Leo Tolstoy	1.9	22 Aug 2007	★★★★★	gutenberg.org	classic	
8	Guns, germs, and steel: the fates of human societies	Jared Diamond	0.4	29 Nov 2007	★★★★★	New York : W.W. Norton, c1997.		
9	A Game of Thrones	George R. R. Martin	1.3	23 Jan 2007	★★★★★		fantasy	
10	A Clash of Kings	George R. R. Martin	1.4	25 Jan 2007	★★★★★		fantasy	
11	A Storm of Swords	George R. R. Martin	1.9	27 Jan 2007	★★★★★		fantasy	
12	A Feast for Crows	George R. R. Martin	1.7	29 Jan 2007	★★★★★		fantasy	Song of Ice and Fire [4]
13	Don Quixote	Jacqueline Carey	0.8	09 May 2007	★★★★★		fantasy	The Sundering [1]

The Search & Sort section allows you to perform several powerful actions on your book collections.

- You can sort them by title, author, date, rating, etc. by clicking on the column titles. You can also sub-sort, i.e. sort on multiple columns. For example, if you click on the title column and then the author column, the book will be sorted by author and then all the entries for the same author will be sorted by title.
- You can search for a particular book or set of books using the Search bar. More on that below.
- You can quickly and conveniently edit metadata by selecting the entry you want changed in the list and pressing the **E** key.
- You can perform *Działania* (strona 4) on sets of books. To select multiple books you can either:
 - Keep the **Ctrl** key pressed and click on the books you want selected.
 - Keep the **Shift** key pressed and click on the starting and ending book of a range of books you want selected.
- You can configure which fields you want displayed by using the *Ustawienia* (strona 10) dialog.
- To perform complex multiple column based sub-sorting add the *Sort by* tool to a toolbar via *Preferences* → *Toolbars & menus*.

1.5 The search interface

You can search all book metadata by entering search terms in the Search bar. For example:

```
Asimov Foundation format:lrf
```

This will match all books in your library that have *Asimov* and *Foundation* in their metadata and are available in the LRF format. Some more examples:

```
author:Asimov and not series:Foundation
title:"The Ring" or "This book is about a ring"
format:epub publisher:feedbooks.com
```

Search kinds

There are four search kinds: *contains*, *equality*, *regular expression* (see [regular expressions](#)¹), and *character variant*. You choose the search kind with a prefix character.

«Contains» searches

Searches with no prefix character are *contains* and are by default case insensitive. An item matches if the search string appears anywhere in the indicated metadata. You can make *contains* searches case sensitive by checking the option *Case sensitive searching* in *Preferences / Searching*. If the search option *Unaccented characters match accented characters and punctuation is ignored* is checked then a character will match all its variants (e.g., *e* matches *é*, *è*, *ê*, and *ë*) and all punctuation and whitespace are ignored. For example, if the *Unaccented characters match ...* option is checked then given the two book titles:

1. Big, Bothéred, and Bad
2. Big Bummer

then these searches find:

- `title:"er"` matches both («e» matches both «é» and «e»).
- `title:"g "` matches both because spaces are ignored.
- `title:"g, "` matches both because the comma is ignored.
- `title:"gb"` matches both because «,» is ignored in book 1 and spaces are ignored in book 2.

¹ https://en.wikipedia.org/wiki/Regular_expression

- `title:"g b"` matches both because comma and space are ignored.
- `title:"db"` matches #1 because the space in «and Bad» is ignored.
- `title:","` matches both (it actually matches all books) because commas are ignored.

If the *Unaccented characters match ...* option is not checked then character variants, punctuation, and spacing are all significant.

You can set only one of the search options *Case sensitive searching* and *Unaccented characters match accented characters and punctuation is ignored*.

«Equality» searches

Equality searches are indicated by prefixing the search string with an equals sign (=). For example, the query `tag:"=science"` will match *science*, but not *science fiction* or *hard science*. Character variants are significant: *é* doesn't match *e*.

Two variants of equality searches are used for hierarchical items (e.g., A.B.C): hierarchical prefix searches and hierarchical component searches. The first, indicated by a single period after the equals (=.) matches the initial parts of a hierarchical item. The second, indicated by two periods after the equals (=..) matches an internal name in the hierarchical item. Examples, using the tag `History.Military.WWII` as the value:

- `tags:"=.History" : True`. History is a prefix of the tag.
- `tags:"=.History.Military" : True`. History.Military is a prefix of the tag.
- `tags:"=.History.Military.WWII" : True`. History.Military.WWII is a prefix of the tag, albeit an improper one.
- `tags:"=.Military" : False`. Military is not a prefix of the tag.
- `tags:"=.WWII" : False`. WWII is not a prefix of the tag.
- `tags:"=..History" : True`. The hierarchy contains the value History.
- `tags:"=..Military" : True`. The hierarchy contains the value Military.
- `tags:"=..WWII" : True`. The hierarchy contains the value WWII.
- `tags:"=..Military.WWII" : False`. The .. search looks for single values.

«Regular expression» searches

Regular expression searches are indicated by prefixing the search string with a tilde (~). Any [Python-compatible regular expression](https://docs.python.org/library/re.html)² can be used. Backslashes used to escape special characters in regular expressions must be doubled because single backslashes will be removed during query parsing. For example, to match a literal parenthesis you must enter `\\ (` or alternatively use *super-quotes* (see below). Regular expression searches are «contains» searches unless the expression is anchored. Character variants are significant: *~e* doesn't match *é*.

«Character variant» searches

Character variant searches are indicated by prefixing the search string with a caret (^). This search is similar to the *contains* search (above) except that:

- letter case is always ignored.
- character variants always match each other.
- punctuation and whitespace are always significant.

The search options *Unaccented characters match accented characters and punctuation is ignored* and *Case sensitive searching* are ignored. They have no effect on this search's behavior.

² <https://docs.python.org/library/re.html>

The following compares this search to a contains search assuming the *Unaccented characters match...* option is checked (see above) given the same two book titles:

1. Big, Bothéred, and Bad
2. Big Bummer

then these character variant searches find:

- `title:"^er"` matches both («e» matches both «é» and «e»)
- `title:"^g"` matches both
- `title:"^g "` matches #2 because the space is significant
- `title:"^g, "` matches #1 because the comma is significant
- `title:"^gb"` matches nothing because space and comma are significant
- `title:"^g b"` matches #2 because the comma is significant
- `title:"^db"` matches nothing
- `title:"^, "` matches #1 (instead of all books) because the comma is significant

Search Expression Syntax

A *search expression* is a sequence of *search terms* optionally separated by the operators `and` and `or`. If two search terms occur without a separating operator, `and` is assumed. The `and` operator has priority over the `or` operator; for example the expression `a or b and c` is the same as `a or (b and c)`. You can use parenthesis to change the priority; for example `(a or b) and c` to make the `or` evaluate before the `and`. You can use the operator `not` to negate (invert) the result of evaluating a search expression. Examples:

- `not tag:foo` finds all books that don't contain the tag `foo`
- `not (author:Asimov or author:Weber)` finds all books not written by either Asimov or Weber.

The above examples show examples of *search terms*. A basic *search term* is a sequence of characters not including spaces, quotes (`"`), backslashes (`\`), or parentheses (`()`). It can be optionally preceded by a column name specifier: the *lookup name* of a column followed by a colon (`:`), for example `author:Asimov`. If a search term must contain a space then the entire term must be enclosed in quotes, as in `title:"The Ring"`. If the search term must contain quotes then they must be *escaped* with backslashes. For example, to search for a series named *The „Ball” and The „Chain”*, use:

```
series:"The \"Ball\" and The \"Chain\""
```


If you need an actual backslash, something that happens frequently in *regular expression* searches, use two of them (`\\`).

It is sometimes hard to get all the escapes right so the result is what you want, especially in *regular expression* and *template* searches. In these cases use the *super-quote*: `" " "sequence of characters" "`. Super-quoted characters are used unchanged: no escape processing is done.

More information

To search for a string that begins with an equals, tilde, or caret; prefix the string with a backslash.

Enclose search strings with quotes (`"`) if the string contains parenthesis or spaces. For example, to find books with the tag `Science Fiction` you must search for `tag:"=science fiction"`. If you search for `tag:=science fiction` you will find all books with the tag `science` and the word `fiction` in any metadata.

You can build advanced search queries easily using the *Advanced search dialog* accessed by clicking the button  on the left of the search box.

Available fields for searching are: `tag`, `title`, `author`, `publisher`, `series`, `series_index`, `rating`, `cover`, `comments`, `format`, `identifiers`, `date`, `pubdate`, `search`, `size`, `vl`

and custom columns. If a device is plugged in, the `ondevice` field becomes available, when searching the calibre library view. To find the search name (actually called the *lookup name*) for a custom column, hover your mouse over the column header in the library view.

Dates

Składnia wyszukiwania po datach jest następująca:

```
pubdate:>2000-1 Will find all books published after Jan, 2000
date:<=2000-1-3 Will find all books added to calibre before 3 Jan, 2000
pubdate:=2009 Will find all books published in 2009
```

If the date is ambiguous then the current locale is used for date comparison. For example, in an mm/dd/yyyy locale 2/1/2009 is interpreted as 1 Feb 2009. In a dd/mm/yyyy locale it is interpreted as 2 Jan 2009. Some special date strings are available. The string `today` translates to today's date, whatever it is. The strings `yesterday` and `thismonth` (or the translated equivalent in the current language) also work. In addition, the string `daysago` (also translated) can be used to compare to a date some number of days ago. For example:

```
date:>10daysago
date:<=45daysago
```

To avoid potential problems with translated strings when using a non-English version of calibre, the strings `_today`, `_yesterday`, `_thismonth`, and `_daysago` are always available. They are not translated.

Searching dates and numeric values with relational comparisons

Dates and numeric fields support the relational operators `=` (equals), `>` (greater than), `>=` (greater than or equal to), `<` (less than), `<=` (less than or equal to), and `!=` (not equal to). Rating fields are considered to be numeric. For example, the search `rating:>=3` will find all books rated 3 or higher.

You can search for books that have a format of a certain size like this:

- `size:>1.1M` will find books with a format larger than 1.1MB
- `size:<=1K` will find books with a format smaller than or equal to 1KB

You can search for the number of items in multiple-valued fields such as tags using the character `#` then using the same syntax as numeric fields. For example, to find all books with more than 4 tags use `tags:#>4`. To find all books with exactly 10 tags use `tags:#=10`.

Series indices

Series indices are searchable. For the standard series, the search name is `series_index`. For custom series columns, use the column search name followed by `_index`. For example, to search the indices for a custom series column named `#my_series`, you would use the search name `#my_series_index`. Series indices are numbers, so you can use the relational operators described above.

Saved searches

The special field `search` is used for *saved searches* (strona 17). If you save a search with the name „My spouse's books” you can enter `search:"My spouse's books"` in the Search bar to reuse the saved search. More about saving searches below.

Virtual libraries

The special field `v1` is used to search for books in a Virtual library. For example, `v1:Read` will find all the books in the *Read* Virtual library. The search `v1:Read` and `v1:"Science Fiction"` will find all the books that are in both the *Read* and *Science Fiction* virtual libraries. The value following `v1:` must be the name of a Virtual library. If the Virtual library name contains spaces then surround it with quotes.

Whether a field has a value

You can search for the absence or presence of a value for a field using „true” and „false”. For example:

- `cover:false` finds all books without a cover
- `series:true` finds all books that are in a series
- `series:false` finds all books that are not in a series
- `comments:false` finds all books with an empty comment
- `formats:false` finds all books with no book files (empty records)

Yes/no custom columns

Searching `Yes/no` custom columns for `false`, `empty`, or `blank` will find all books with undefined values in the column. Searching for `true` will find all books that do not have undefined values in the column. Searching for `yes` or `checked` will find all books with `Yes` in the column. Searching for `no` or `unchecked` will find all books with `No` in the column. Note that the words `yes`, `no`, `blank`, `empty`, `checked` and `unchecked` are translated; you can use either the current language's equivalent word or the English word. The words `true` and `false` and the special values `_yes`, `_no`, and `_empty` are not translated.

Identifiers

Identifiers (e.g., ISBN, DOI, LCCN, etc.) use an extended syntax. An identifier has the form `type:value`, as in `isbn:123456789`. The extended syntax permits you to specify independently the type and value to search for. Both the type and the value parts of the query can use any of the [search kinds](#) (strona 12). Examples:

- `identifiers:true` will find books with any identifier.
- `identifiers:false` will find books with no identifier.
- `identifiers:123` will search for books with any type having a value containing *123*.
- `identifiers:=123456789` will search for books with any type having a value equal to *123456789*.
- `identifiers:=isbn:` and `identifiers:isbn:true` will find books with a type equal to ISBN having any value
- `identifiers:=isbn:false` will find books with no type equal to ISBN.
- `identifiers:=isbn:123` will find books with a type equal to ISBN having a value containing *123*.
- `identifiers:=isbn:=123456789` will find books with a type equal to ISBN having a value equal to *123456789*.
- `identifiers:i:1` will find books with a type containing an *i* having a value containing a *1*.

Categories visible in the Tag browser

The search `in_tag_browser:true` finds all books that are in categories (tags, authors, etc.) currently shown in the *Tag browser*. This is useful if you set the two preferences *Preferences* → *Look & feel* → *Tag browser* → *Hide empty categories* and *Find shows all items that match*. With those two preferences set, doing a *Find* in the *Tag browser* shows only categories containing items matched by the *Find*. Then, the search `in_tag_browser:true` additionally finds books with these categories / items. You can easily run this search by pressing the key `Ctrl+Alt+Shift+F` or clicking the configure button in the *Tag browser* and choosing the *Show only books that have visible categories* entry.

Search using templates


You can search using a template in *Język szablonów calibre* (strona 161) instead of a metadata field. To do so you enter a template, a search type, and the value to search for. The syntax is:

```
template: (the template) #@#: (search type) : (the value)
```

The `template` is any valid calibre template language template. The `search type` must be one of `t` (text search), `d` (date search), `n` (numeric search), or `b` (set/not set (boolean)). The `value` is whatever you want, and can use the [search kinds](#) (strona 12) described above for the various search types. You must quote the entire search string if there are spaces anywhere in it.

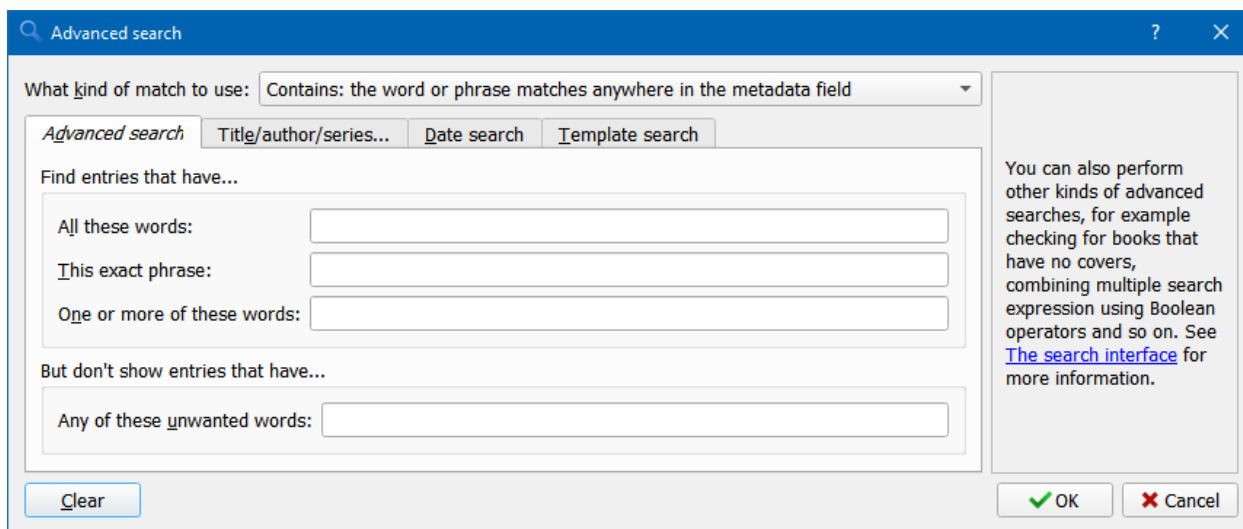
Przykłady:

- `template:"program: connected_device_name('main')#@#:t:kindle"` – is true when the kindle device is connected.
- `template:"program: select(formats_sizes(), 'EPUB')#@#:n:>1000000"` – finds books with EPUB files larger than 1 MB.
- `template:"program: select(formats_modtimes('iso'), 'EPUB')#@#:d:>10daysago"` – finds books with EPUB files newer than 10 days ago.
- `template:"""program: book_count('tags:^[^']* & $series & ''', 0) != 0#@#:n:1"""` – finds all books containing the series name in the tags. This example uses super-quoting because the template uses both single quotes (') and double quotes (") when constructing the search expression.

You can build template search queries easily using the *Advanced search dialog* accessed by clicking the button .

You can test templates on specific books using the calibre *Template tester*, which can be added to the toolbars or menus via *Preferences* → *Toolbars & menus*. It can also be assigned a keyboard shortcut via *Preferences* → *Shortcuts*.

Advanced search dialog



Rys. 1: *Advanced search dialog*

1.6 Zapisywanie wyszukiwań

calibre allows you to save a frequently used search under a special name and then reuse that search with a single click. To do this, create your search either by typing it in the Search bar or using the Tag browser. Then type the name you would like to give to the search in the Saved Searches box next to the Search bar. Click the plus icon next to the saved searches box to save the search.

Now you can access your saved search in the Tag browser under *Saved searches*. A single click will allow you to reuse any arbitrarily complex search easily, without needing to re-create it.

1.7 Searching the full text of all books



calibre can *optionally* index the full text of books in the library to allow for instant searching of words inside any book. To use this functionality click the *FT* button at the left edge of the search bar.

Then, enable indexing for the current library. Once indexing is complete you can search all the text in the full library. When you add new books to the library, they will be automatically indexed in the background. This search allows you to quickly find all books that contain a word or combination of words. You can even search for words that occur near other words, as shown in the examples in the search popup window. Note that this search tool will find only one occurrence of the search query in a particular book, not list every occurrence, for that it is best to search inside the book using the calibre *E-book viewer*.

You can re-index an individual book by right clicking on the *Book details panel* in calibre and choosing *Re-index this book for full text searching*.

1.8 Wirtualne biblioteki

A *Virtual library* is a way to pretend that your calibre library has only a few books instead of its full collection. This is an excellent way to partition your large collection of books into smaller, manageable chunks. To learn how to create and use Virtual libraries, see the tutorial: [Wirtualne biblioteki](#) (strona 247).

1.9 Temporarily marking books

You can temporarily mark arbitrary sets of books. Marked books will have a pin on them and can be found with the search `marked:true`. To mark a book press `Ctrl+M` or go to *Preferences* → *Toolbars & menus* and add the *Mark books* button to the main toolbar.

You can mark books with a specific text label by right clicking the *Mark books* button and choosing *Mark books with text label*. Books marked with text labels can later be found using the search `marked:"=the-text-you-entered"`.

1.10 Guessing metadata from file names

Normally, calibre reads metadata from inside the book file. However, it can be configured to read metadata from the file name instead, via *Preferences* → *Import/export* → *Adding books* → *Read metadata from file contents*.

You can also control how metadata is read from the filename using regular expressions (see [Wszystko, co chcielibyście wiedzieć o wyrażeniach regularnych w calibre](#) (strona 213)). In the *Adding books* section of the configuration dialog, you can specify a regular expression that calibre will use to try and guess metadata from the names of e-book files that you add to the library. The default regular expression is:

```
title - author
```

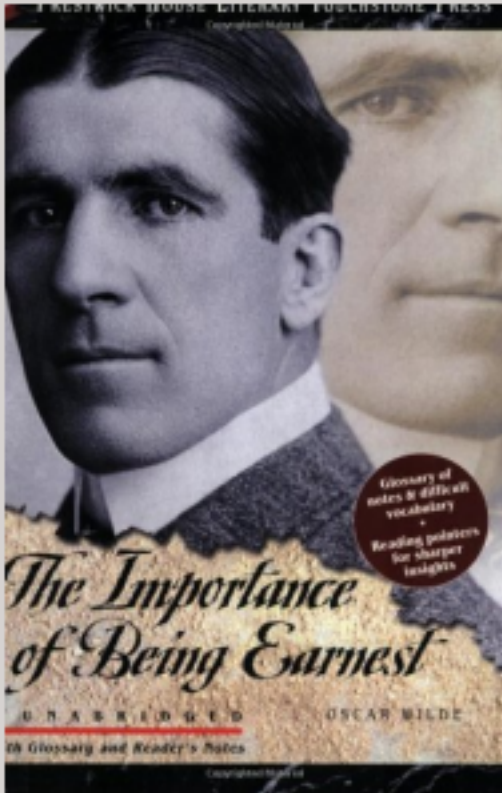
that is, it assumes that all characters up to the first – are the title of the book and subsequent characters are the author of the book. For example, the filename:

```
Foundation and Earth - Isaac Asimov.txt
```

will be interpreted to have the title: Foundation and Earth and author: Isaac Asimov

Wskazówka: If the filename does not contain the hyphen, the above regular expression will fail.

1.11 Szczegóły książki



Authors: [Oscar Wilde](#)

Formats: [EPUB](#)

Ids: [9781580495806](#)

Tags: [lit 101 homework](#)

Path: [Click to open](#)

SUMMARY:
This Prestwick House Literary Touchstone Edition includes a glossary and reader's notes to help the modern reader appreciate Wilde's wry wit and elaborate plot twists. Oscar Wilde's madcap farce about mistaken identities, secret engagements, and lovers' entanglements still delights readers

The Book details display shows the cover and all the metadata for the currently selected book. It can be hidden via the

Layout button in the lower right corner of the main calibre window. The author names shown in the Book details panel are click-able, they will by default take you to the Wikipedia page for the author. This can be customized by right clicking on the author name and selecting *Manage this author*.

Similarly, if you download metadata for the book, the Book details panel will automatically show you links pointing to the web pages for the book on Amazon, WorldCat, etc. from where the metadata was downloaded.

You can right click on individual e-book formats in the Book details panel to delete them, compare them to their original versions, save them to disk, open them with an external program, etc.

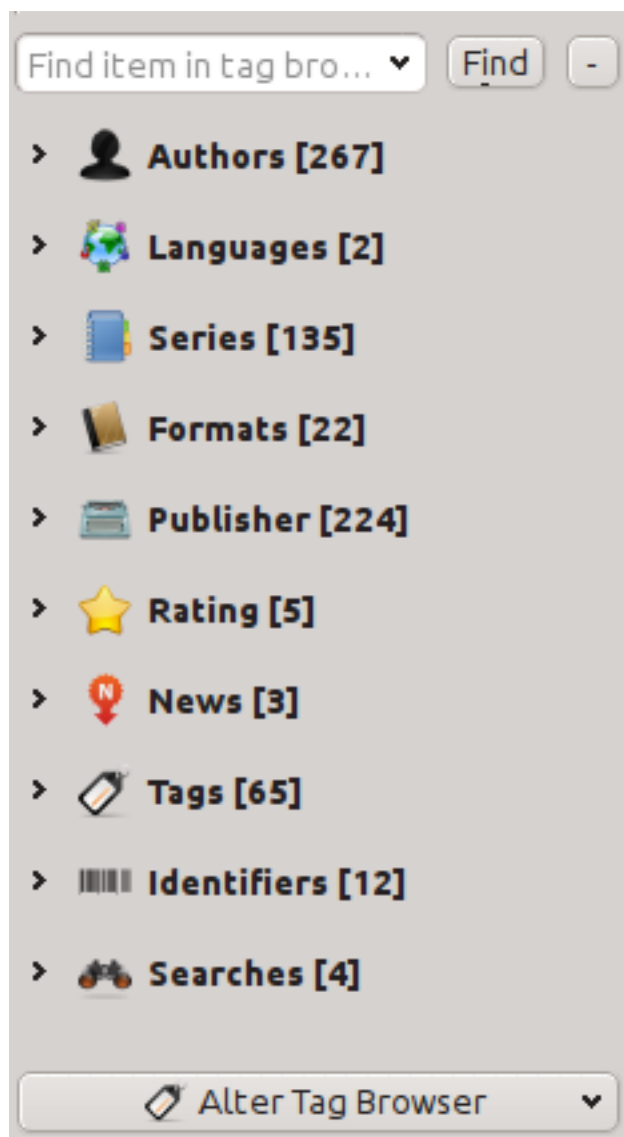
You can change the cover of the book by simply drag and dropping an image onto the Book details panel. If you wish to edit the cover image in an external program, simply right click on it and choose *Open with*.

You can also add e-book files to the current book by drag and dropping the files onto the Book details panel.

Double clicking the Book details panel will open it up in a separate popup window.

Finally, you can customize exactly what information is displayed in the Book details panel via *Preferences → Interface → Look & feel → Book details*.

1.12 Przeglądarka etykiet



The Tag browser allows you to easily browse your collection by Author/Tags/Series/etc. If you click on any item in the Tag browser, for example the author name Isaac Asimov, then the list of books to the right is restricted to showing books by that author. You can click on category names as well. For example, clicking on „Series” will show you all books in any series.

The first click on an item will restrict the list of books to those that contain or match the item. Continuing the above example, clicking on Isaac Asimov will show books by that author. Clicking again on the item will change what is shown, depending on whether the item has children (see sub-categories and hierarchical items below). Continuing the Isaac Asimov example, clicking again on Isaac Asimov will restrict the list of books to those not by Isaac Asimov. A third click will remove the restriction, showing all books. If you hold down the **Ctrl** or **Shift** keys and click on multiple items, then restrictions based on multiple items are created. For example you could hold **Ctrl** and click on the tags History and Europe for finding books on European history. The Tag browser works by constructing search expressions that are automatically entered into the Search bar. Looking at what the Tag browser generates is a good way to learn how to construct basic search expressions.

Items in the Tag browser have their icons partially colored. The amount of color depends on the average rating of the

books in that category. So for example if the books by Isaac Asimov have an average of four stars, the icon for Isaac Asimov in the Tag browser will be 4/5th colored. You can hover your mouse over the icon to see the average rating.

The outer-level items in the *Tag browser*, such as Authors and Series, are called categories. You can create your own categories, called *User categories*, which are useful for organizing items. For example, you can use the *User categories editor* (click the *Configure* button at the lower-left of the *Tag browser* and choose *Manage authors, tags, etc* → *User categories*) to create a User category called `Favorite Authors`, then put the items for your favorites into the category. User categories can have sub-categories. For example, the User category `Favorites.Authors` is a sub-category of `Favorites`. You might also have `Favorites.Series`, in which case there will be two sub-categories under `Favorites`. Sub-categories can be created by right-clicking on a User category, choosing *Add sub-category to...*, and entering the sub-category name; or by using the *User categories editor* by entering names like the `Favorites` example above.

You can search User categories in the same way as built-in categories, by clicking on them. There are four different searches cycled through by clicking:

1. „everything matching an item in the category” indicated by a single green plus sign.
2. „everything matching an item in the category or its sub-categories” indicated by two green plus signs.
3. „everything not matching an item in the category” shown by a single red minus sign.
4. „everything not matching an item in the category or its sub-categories” shown by two red minus signs.

It is also possible to create hierarchies inside some of the text categories such as tags, series, and custom columns. These hierarchies show with the small triangle, permitting the sub-items to be hidden. To use hierarchies of items in a category, you must first go to *Preferences* → *Interface* → *Look & feel* and enter the category name(s) into the „Categories with hierarchical items” field. Once this is done, items in that category that contain periods will be shown using the small triangle. For example, assume you create a custom column called „Genre” and indicate that it contains hierarchical items. Once done, items such as `Mystery.Thriller` and `Mystery.English` will display as `Mystery` with the small triangle next to it. Clicking on the triangle will show `Thriller` and `English` as sub-items. See *Managing subgroups of books, for example „genre”* (strona 153) for more information.

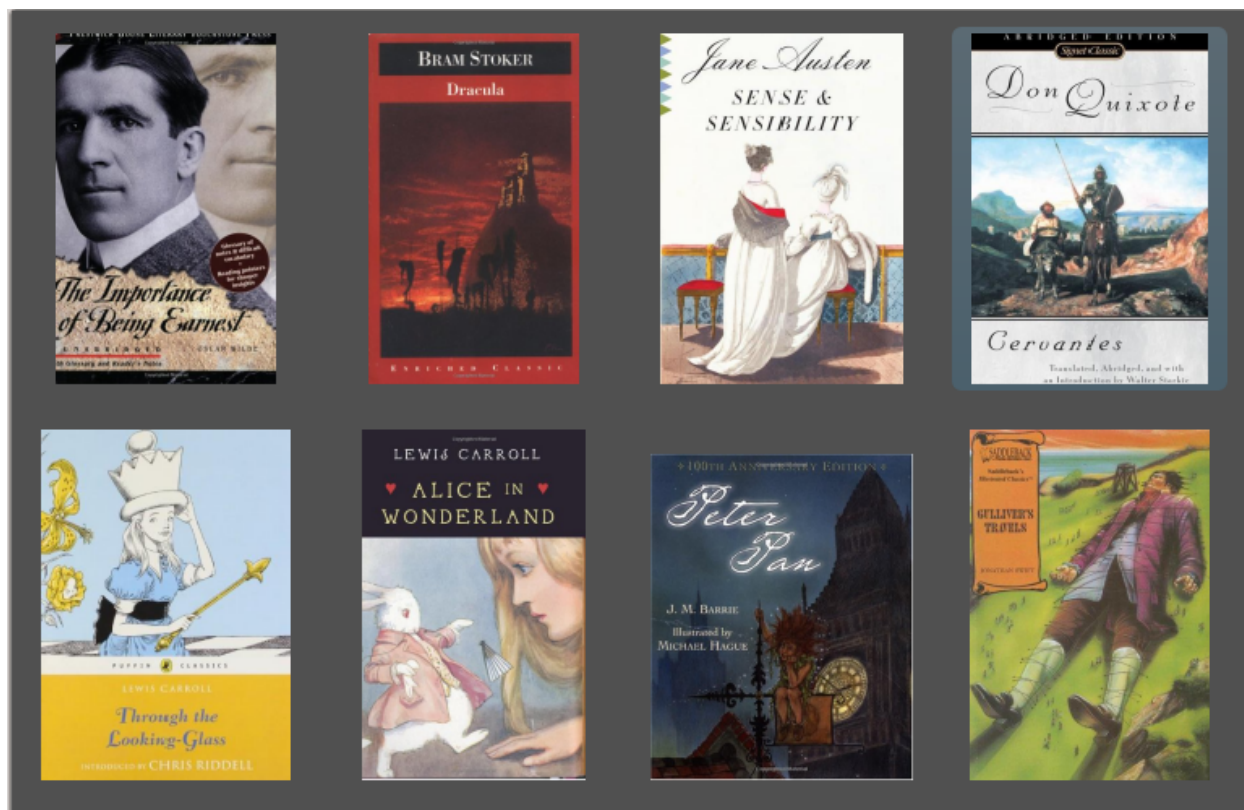
Hierarchical items (items with children) use the same four «click-on» searches as User categories. Items that do not have children use two of the searches: „everything matching” and „everything not matching”.

You can drag and drop items in the Tag browser onto User categories to add them to that category. If the source is a User category, holding the `Shift` key while dragging will move the item to the new category. You can also drag and drop books from the book list onto items in the Tag browser; dropping a book on an item causes that item to be automatically applied to the dropped books. For example, dragging a book onto `Isaac Asimov` will set the author of that book to `Isaac Asimov`. Dropping it onto the tag `History` will add the tag `History` to the book’s tags.

You can easily find any item in the Tag browser by clicking the search button at the lower-right corner. In addition, you can right click on any item and choose one of several operations. Some examples are to hide it, rename it, or open a „Manage x” dialog that allows you to manage items of that kind. For example, the *Manage authors* dialog allows you to rename authors and control how their names are sorted.

You can control how items are sorted in the Tag browser via the *Configure* button at the lower-left of the Tag browser. You can choose to sort by name, average rating or popularity (popularity is the number of books with an item in your library; for example, the popularity of `Isaac Asimov` is the number of books in your library by `Isaac Asimov`).

1.13 Widok okładek



You can have calibre display a grid of book covers instead of a list of books, if you prefer to browse your collection by covers instead. The *Cover grid* is activated by clicking the *Layout* button in the bottom right corner of the main calibre window. You can customize the cover sizes and the background of the *Cover grid* via *Preferences* → *Interface* → *Look & feel* → *Cover grid*. You can even have calibre display any specified field under the covers, such as title or authors or rating or a custom column of your own devising.

1.14 Przeglądarka okładek



In addition to the *Cover grid* described above, you can also have calibre display covers in the single row. This is activated via the *Layout* button in the lower right corner of the main window. In *Preferences* → *Interface* → *Look & feel* → *Cover browser* you can change the number of covers displayed, and even have the *Cover browser* display itself in a separate popup window.

1.15 Adding notes for authors, series, etc.

William Shakespeare



William Shakespeare (baptised 26 April 1564) was an English poet and playwright, widely regarded as the greatest writer in the English language and the world's pre-eminent dramatist. He is often called England's national poet and the "Bard of Avon" (or simply "The Bard"). His surviving works consist of 38 plays, 154 sonnets, two long narrative poems, and several other poems. His plays have been translated into every major living language, and are performed more often than those of any other playwright.

Shakespeare was born and raised in Stratford-upon-Avon. Scholars believe that he died on his fifty-second birthday, coinciding with St George's Day.

At the age of 18 he married Anne Hathaway, who bore him three children: Susanna, and twins Hamnet and Judith. Between 1585 and 1592 he began a successful career in London as an actor, writer, and part owner of the playing company the Lord Chamberlain's Men, later known as the King's Men.

 Edit
  Find books
  Copy URL
  Close

You can add notes for an author/series/tag/publisher/etc. to your calibre library. To do so right click on the author name in the *Tag browser* on the left or the *Book details* panel on the right and choose *Create note* or *Edit note*.

A simple popup window will allow you to enter your notes using basic formatting and supporting links and images. Once a note for an author is created, it can be viewed easily from the *Book details* panel by clicking the little pencil icon next to the author name.

You can search through all the notes in your library using the *Browse notes* tool by pressing `Ctrl+Shift+N` or adding it to the toolbar via *Preferences* → *Toolbars & menus*.

1.16 Podgląd

Sometimes you want to select a book and quickly get a list of books with the same value in some category (authors, tags, publisher, series, etc.) as the currently selected book, but without changing the current view of the library. You can do this with Quickview. Quickview opens either a second window or a panel in the book list showing the list of books matching the value of interest. For example, assume you want to see a list of all the books with the one or more of the authors of the currently-selected book. Click in the author cell you are interested in and press the «Q» key or click the *Quickview* icon in the *Layout* section of the calibre window. A window or panel will open with all the authors for that book on the left, and all the books by the selected author on the right.

Some example Quickview usages: quickly seeing what other books:

- have some tag(s) applied to the currently selected book,
- are in the same series as the current book
- have the same values in a custom column as the current book
- are written by one of the same authors of the current book
- share values in a custom column

There are two choices for where the Quickview information appears:

1. It can open „undocked”: on top of the calibre window and will stay open until you explicitly close it.
2. It can open „docked”: as a panel in the book list section of the calibre main window.

You can move the window from docked to undocked as desired using the „Dock/Undock” button.

The Quickview panel can be left open permanently, in which case it follows movements on the book list. For example, if you click in the calibre library view on a category column (tags, series, publisher, authors, etc.) for a book, the Quickview window contents will change to show you in the left-hand side panel the values in that category for the selected book (e.g., the tags for that book). The first item in that list will be selected, and Quickview will show you on the right-hand side panel all the books in your library that use that value. Click on an different value in the left-hand panel to see the books with that different value.

Double-click on a book in the Quickview window to select that book in the library view. This will also change the items display in the QuickView window (the left-hand panel) to show the items in the newly-selected book.

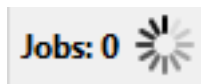
Shift- or **Ctrl-** double-click on a book in the Quickview window to open the edit metadata dialog on that book in the calibre window. The edited book will be Quickview'ed when you close the edit metadata dialog.

You can see if a column can be Quickview'ed by hovering your mouse over the column heading and looking at the tooltip for that heading. You can also know by right-clicking on the column heading to see if the „Quickview” option is shown in the menu, in which case choosing that Quickview option is equivalent to pressing «Q» in the current cell.

Options (in *Preferences* → *Look & feel* → *Quickview*):

- Respect (or not) the current Virtual library. If checked then Quickview shows only books in the current Virtual library. Default: respect Virtual libraries
- Change the Quickview window contents when the column is changed on the book list using the cursor keys. Default: don't follow changes made with cursor keys
- Change the column being „quickview'ed” when a cell in the Quickview window is double-clicked. Otherwise the book is changed but the column being examined is not. Default: change the column
- Change the column being „quickview'ed” to the current column when the return key is pressed in the Quickview panel. Otherwise the book is changed but the column being examined is not. Default: change the column
- Choose which columns are shown in the Quickview window/panel.

1.17 Zadania



The Jobs panel shows the number of currently running jobs. Jobs are tasks that run in a separate process. They include converting e-books and talking to your reader device. You can click on the jobs panel to access the list of jobs. Once a job has completed you can see a detailed log from that job by double-clicking it in the list. This is useful to debug jobs that may not have completed successfully.

1.18 Skróty klawiszowe

calibre has several keyboard shortcuts to save you time and mouse movement. These shortcuts are active in the book list view (when you're not editing the details of a particular book), and most of them affect the title you have selected. The calibre E-book viewer *has its own shortcuts* (strona 55) which can be customised in the viewer *Preferences*.

Informacja: Note: The calibre keyboard shortcuts do not require a modifier key (Command, Option, Control, etc.), unless specifically noted. You only need to press the letter key, e.g. E to edit.

Tabela 1: Keyboard shortcuts for the main calibre program

Keyboar shortcut	Działanie
F2 (Enter for macOS)	Edit the metadata of the currently selected field in the book list.
A	Dodaj książki
Shift+.	Add formats to the selected books
C	Convert selected books
D	Prześlij do urządzenia
Del	Usuń wybrane książki
E	Edit metadata of selected books
G	Zdobądź książki
I	Pokaż szczegóły książki
K	Edytuj spis treści
M	Merge selected records
Alt+M	Merge selected records, keeping originals
O	Otwórz folder zawierający
P	Dopracuj książki
S	Zapisz na dysku
T	Edytuj książkę
V	Wyświetl
Shift+`	View last read book
Alt+V/ Cmd+V for macOS	Wyświetl w określonym formacie

ciąg dalszy na następnej stronie

Tabela 1 – kontynuacja poprzedniej strony

Keyboar shortcut	Działanie
Alt+Sh	Toggle jobs list
Alt+Sh	Toggle Cover browser
Alt+Sh	Toggle Book details panel
Alt+Sh	Toggle Tag browser
Alt+Sh	Toggle Cover grid
Alt+A	Show books by the same author as the current book
Alt+T	Show books with the same tags as current book
Alt+P	Show books by the same publisher as current book
Alt+Sh	Show books in the same series as current book
/,	Focus the Search bar
Ctrl+F	
Shift+	Open the Advanced search dialog
Shift+	Toggle the Search bar
Esc	Wyczyść aktualne wyszukiwanie
Shift+	Przełącz na listę książek
Ctrl+E	Wyczyść wirtualną bibliotekę
Alt+Es	Wyczyść dodatkowe ograniczenia
Ctrl+*	Create a temporary Virtual library based on the current search
Ctrl+R	Select the next Virtual library tab
Ctrl+L	Select the previous Virtual library tab
N or F3	Find the next book that matches the current search (only works if search highlighting is turned on in search preferences)
Shift+ or Shift+	Find the previous book that matches the current search (only works if search highlighting is turned on in search preferences)
Ctrl+D	Pobierz metadane i okładki
Ctrl+R	Uruchom calibre ponownie
Ctrl+S	Restart calibre in debug mode
Shift+	Add empty books to calibre
Ctrl+M	Toggle mark/unmarked status on selected books
Ctrl+/	Open the popup to search the full text of all books in the library
or Ctrl+A	
Q	Open the Quick View popup for viewing books in related series/tags/etc.
Shift+	Focus the opened Quick View panel
Shift+	Perform a search in the Quick View panel
F5	Sortuj ponownie
Ctrl+Q	Zakończ calibre
X	Toggle auto scroll of the book list
Ctrl+A	Restrict the displayed books to only those books that are in a category currently displayed in the <i>Tag browser</i>
B	Browse annotations (highlights and bookmarks) made in the calibre viewer for all books in the library
Ctrl+S	Browse notes associated with authors/series/tags/etc.
Alt+Sh	Toggle the layout between wide and narrow views

Dodawanie ulubionej strony z wiadomościami

calibre zawiera potężne, elastyczne i łatwe w użyciu narzędzie, służące do pobierania wiadomości z internetu i przekształcaniu ich w ebook. Poniżej pokażemy na przykładach, jak pobrać wiadomości z różnych witryn.

Do zrozumienia, w jaki sposób korzystać z tego narzędzia posłużą kolejno następujące przykłady:

- *Całkowicie automatyczne pobieranie* (strona 32)
 - *Blog calibre* (strona 32)
 - *bbc.co.uk* (strona 33)
- *Dostrajanie procesu pobierania* (strona 33)
 - *Używanie wersji do druku witryny bbc.co.uk* (strona 33)
 - *Zamiana stylów* (strona 35)
 - *Szatkowanie wiadomości* (strona 35)
 - *Przykład z życia wzięty* (strona 36)
- *Wskazówki przydatne przy tworzeniu nowych źródeł* (strona 38)
- *Dalsza lektura* (strona 39)
- *Dokumentacja API* (strona 39)

2.1 Całkowicie automatyczne pobieranie

Jeśli pobierane wiadomości mają dostatecznie prostą strukturę, calibre może sobie z nimi poradzić całkowicie automatycznie, a wszystko, co musisz zrobić to podać adres witryny. calibre, zgromadziwszy wszystkie informacje, potrzebne do pobrania wiadomości, umieści je w źródle. Aby poinformować calibre o wiadomościach do pobrania, musisz stworzyć dla nich *źródło*. Oto kilka przykładów:

2.1.1 Blog calibre

Blog calibre zawiera wiele przydatnych informacji o funkcjach programu, podanych w przejrzysty i przystępny sposób. Aby pobrać wiadomości z tego bloga można użyć kanału RSS:

```
http://blog.calibre-ebook.com/feeds/posts/default
```

Adres kanału RSS można znaleźć na dole strony poniżej „Subscribe to”, należy wybrać *Posts* → *Atom*. Aby calibre pobrał wiadomości i przekształcił je w ebook, musisz kliknąć prawym klawiszem myszy przycisk *Pobierz wiadomości* i wybrać *Dodaj własne źródło wiadomości*, a następnie przycisk *Nowe źródło*. Pojawi się okno podobne do pokazanego poniżej.

Create a basic news recipe, by adding RSS feeds to it.
For some news sources, you will have to use the "Switch to advanced mode" button below to further customize the fetch process.

Recipe title:

Oldest article:

Max. number of articles per feed:

Feeds in recipe

▲


✖


▼

Add feed to recipe


Feed title:

Feed URL:

 Add feed

 Save

Switch to Advanced mode

 Cancel

Najpierw wprowadź „Calibre Blog” w polu: guilabel: „Tytuł przepisu”. Będzie to tytuł e-booka, który zostanie utworzony z artykułów w powyższych kanałach.

Następne dwa pola (*Najstarszy artykuł może być sprzed i Maksymalna liczba artykułów na strumień*) dają pewną kontrolę na tym jakie artykuły zostaną pobrane i raczej nie wymagają objaśnienia.

Aby dodać strumień (odpowiadające kanałom RSS), podaj nazwę strumienia, kanał RSS, z którego będą pobierane artykuły i kliknij przycisk *Dodaj strumień*. Po dodaniu strumienia po prostu kliknij przycisk *Zapisz* i gotowe! Można zamknąć okno.

Aby przetestować nowe *źródło*, kliknij przycisk *Pobierz wiadomości*, a następnie w podmenu *Własne* kliknij *Blog calibre*. Po kilku minutach nowy ebook, powstały z artykułów z bloga, pojawi się na liście książek biblioteki (jeśli jest podłączony czytnik to książka pojawi się na czytniku, a nie w bibliotece). Zaznacz go, kliknij przycisk *Wyświetl* i już możesz czytać!

Działa to tak dobrze przy tak małym nakładzie pracy, gdyż jest to kanał *full-content*, co oznacza, że cała treść artykułu jest w nim osadzona. W przypadku większości kanałów tego typu nie potrzeba nic więcej by przekształcić je w książki. Teraz przyjrzyjmy się źródłu wiadomości, które nie zawiera pełnego tekstu artykułów. W takim przypadku artykuł jest stroną w internecie, a źródło dostarcza jedynie odsyłacza do niego z krótkim podsumowaniem.

2.1.2 bbc.co.uk

Wypróbujmy następujące dwa źródła z *The BBC*:

1. Strona główna wiadomości: https://newsrss.bbc.co.uk/rss/newsonline_world_edition/front_page/rss.xml
2. Nauka/Natura: https://newsrss.bbc.co.uk/rss/newsonline_world_edition/science/nature/rss.xml

Wykonaj kroki opisane w powyższej procedurze, by zbudować źródło dla *The BBC* (przy użyciu podanych wyżej adresów). Otworzywszy pobraną książkę zobaczysz, że calibre wykonał kawał dobrej roboty, wydobywając zawartość każdego artykułu z witryny. Jednak proces ten nie jest doskonały. Czasem pozostają niepożądane fragmenty, takie jak menu i elementy nawigacji albo są usuwane części, które są potrzebne, np. nagłówki. Aby uzyskać doskonale wyodrębniony tekst artykułów trzeba dopracować proces pobierania, o czym traktuje następna część.

2.2 Dostrajanie procesu pobierania

Jeśli chcesz udoskonalić proces wyodrębniania tekstu albo pobrać artykuły ze szczególnie skomplikowanej witryny, możesz wykorzystać możliwości i elastyczność narzędzi źródeł. W tym celu, w oknie *Dodaj własne źródło wiadomości*, kliknij przycisk *Przełącz na tryb zaawansowany*.

Najprostszym i zwykle najbardziej efektywnym sposobem jest użycie wersji do druku pobieranych artykułów. Taka wersja zawiera zazwyczaj znacznie mniej dodatków i łatwiej ją dostosować do potrzeb e-booka. Użyjmy więc takiej wersji artykułów z *The BBC*.

2.2.1 Używanie wersji do druku witryny bbc.co.uk

Pierwszym krokiem jest sprawdzenie ebooka, pobranego uprzednio z bbc.co.uk (strona 33). Na końcu każdego artykułu jest mała notka, informująca, skąd został pobrany artykuł. Skopiuj ten adres do przeglądarki. Następnie, na stronie z artykułem poszukaj odsyłacza „Printable version”. Kliknij go, by zobaczyć wersję do druku. Jest znacznie bardziej przejrzysta. Teraz porównaj oba adresy. U mnie były następujące:

Adres artykułu

<https://news.bbc.co.uk/2/hi/science/nature/7312016.stm>

Adres wersji do druku

<https://newsvote.bbc.co.uk/mpapps/pagetools/print/news.bbc.co.uk/2/hi/science/nature/7312016.stm>

Aby otrzymać wersję do druku, adres każdego artykułu musi zaczynać się od:

newsvote.bbc.co.uk/mpapps/pagetools/print/

Now in the *Advanced mode* of the Custom news sources dialog, you should see something like (remember to select *The BBC* recipe before switching to advanced mode):

```
Recipe source code (python)

class AdvancedUserRecipe1206418393(BasicNewsRecipe):
    title          = u'The BBC'
    oldest_article = 7
    max_articles_per_feed = 100

    feeds          = [(u'News Front Page', u'http://newsrss.bbc.co.uk/rss/newsonlin
```

Zauważysz, że pola z okna w trybie podstawowym zostały przetłumaczone wprost na kod pythona. Teraz trzeba zmienić ten kod tak, by używał wersji do druku przy pobieraniu artykułów. Wszystko, czego potrzeba to dodać dwie linie:

```
def print_version(self, url):
    return url.replace('https://', 'https://newsvote.bbc.co.uk/mpapps/pagetools/print/')
```

To python, więc wcięcia są bardzo ważne. Po dodaniu tych dwóch linii powinno to wyglądać tak:

```
Recipe source code (python)

class AdvancedUserRecipe1206418393(BasicNewsRecipe):
    title          = u'The BBC'
    oldest_article = 7
    max_articles_per_feed = 100

    feeds          = [(u'News Front Page', u'http://newsrss.bbc.co.uk/rss/newsonlin

    def print_version(self, url):
        return url.replace('http://', 'http://newsvote.bbc.co.uk/mpapps/pagetools/p
```

In the above, `def print_version(self, url)` defines a *method* that is called by calibre for every article. `url` is the URL of the original article. What `print_version` does is take that url and replace it with the new URL that points to the print version of the article. To learn about [Python](https://www.python.org)³ see the [tutorial](https://docs.python.org/tutorial/)⁴.

Now, click the *Add/update recipe* button and your changes will be saved. Re-download the e-book. You should have a much improved e-book. One of the problems with the new version is that the fonts on the print version webpage are too small. This is automatically fixed when converting to an e-book, but even after the fixing process, the font size of the menus and navigation bar become too large relative to the article text. To fix this, we will do some more customization, in the next section.

³ <https://www.python.org>

⁴ <https://docs.python.org/tutorial/>

2.2.2 Zamiana stylów

W poprzedniej części okazało się, że czcionka w artykułach, pobranych z *The BBC* w wersji do druku. Jak w wielu innych witrynach, tak i w *The BBC* wielkość czcionki jest ustawiana za pomocą arkusza stylów *CSS*. Można zablokować pobieranie takich arkuszy dodając linię:

```
no_stylesheets = True
```

Teraz kod wygląda następująco:

```
Recipe source code (python)
class AdvancedUserRecipe1206419520(BasicNewsRecipe):
    title          = u'The BBC'
    oldest_article = 7
    max_articles_per_feed = 100
    no_stylesheets = True

    feeds          = [(u'News Front Page', u'http://newsrss.bbc.co.uk/rss/newsonli

    def print_version(self, url):
        return url.replace('http://', 'http://newsvote.bbc.co.uk/mpapps/pagetools/
```

Nowa wersja wygląda nieźle. Jeśli jesteś perfekcjonistą możesz przeczytać kolejną część, która traktuje o modyfikowaniu pobieranej treści.

2.2.3 Szatkowanie wiadomości

calibre może na wiele sposobów manipulować pobieraną treścią. Aby pokazać niektóre z nich, przyjrzyjmy się ponownie naszemu, znanemu już źródłu *The BBC* (strona 35). Przyjrzawszy się kodowi źródłowemu (:term:`HTML`) kilku artykułów (w wersji do druku) zauważymy, że mają one stopkę, która nie zawiera żadnych użytecznych informacji

```
<div class="footer">
...
</div>
```

Można ją usunąć, dodając:

```
remove_tags = [dict(name='div', attrs={'class':'footer'})]
```

do kodu źródła. Na koniec zamienimy *CSS*, który poprzednio został zablokowany, na naszą wersję, odpowiedniejszą do konwersji na ebook:

```
extra_css = '.headline {font-size: x-large;} \n .fact { padding-top: 10pt }'
```

Dzięki tym dodatkom nasza receptura stała się „jakością produkcji”.

to *źródło* dotyka jedynie czubka góry lodowej jeśli chodzi o możliwości calibre. Aby poznać je bliżej, w następnej części przyjrzymy się bliżej bardziej skomplikowanemu, autentycznemu przykładowi.

2.2.4 Przykład z życia wzięty.

Nieco bardziej skomplikowany przykład, wzięty z życia, który pokazuje dokładniej *API* BasicNewsRecipe to *źródło The new York Times*

```
import string, re
from calibre import strftime
from calibre.web.feeds.recipes import BasicNewsRecipe
from calibre.ebooks.BeautifulSoup import BeautifulSoup

class NYTimes(BasicNewsRecipe):

    title          = 'The New York Times'
    __author__     = 'Kovid Goyal'
    description     = 'Daily news from the New York Times'
    timefmt        = ' [%a, %d %b, %Y]'
    needs_subscription = True
    remove_tags_before = dict(id='article')
    remove_tags_after  = dict(id='article')
    remove_tags       = [dict(attrs={'class':['articleTools', 'post-tools', 'side_tool',
→'nextArticleLink clearfix']}),
                        dict(id=['footer', 'toolsRight', 'articleInline', 'navigation',
→'archive', 'side_search', 'blog_sidebar', 'side_tool', 'side_index']),
                        dict(name=['script', 'noscript', 'style'])]
    encoding        = 'cp1252'
    no_stylesheets  = True
    extra_css       = 'h1 {font: sans-serif large;}\n.byline {font:monospace;}'

    def get_browser(self):
        br = BasicNewsRecipe.get_browser(self)
        if self.username is not None and self.password is not None:
            br.open('https://www.nytimes.com/auth/login')
            br.select_form(name='login')
            br['USERID'] = self.username
            br['PASSWORD'] = self.password
            br.submit()
        return br

    def parse_index(self):
        soup = self.index_to_soup('https://www.nytimes.com/pages/todayspaper/index.
→html')

        def feed_title(div):
            return ''.join(div.findAll(text=True, recursive=False)).strip()

        articles = {}
        key = None
        ans = []
        for div in soup.findAll(True,
            attrs={'class':['section-headline', 'story', 'story headline']}):

            if ''.join(div['class']) == 'section-headline':
                key = string.capwords(feed_title(div))
                articles[key] = []
                ans.append(key)

            elif ''.join(div['class']) in ['story', 'story headline']:
                a = div.find('a', href=True)
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

        if not a:
            continue
        url = re.sub(r'\?.*', '', a['href'])
        url += '?pagewanted=all'
        title = self.tag_to_string(a, use_alt=True).strip()
        description = ''
        pubdate = strftime('%a, %d %b')
        summary = div.find(True, attrs={'class': 'summary'})
        if summary:
            description = self.tag_to_string(summary, use_alt=False)

        feed = key if key is not None else 'Uncategorized'
        if feed not in articles:
            articles[feed] = []
        if not 'podcasts' in url:
            articles[feed].append(
                dict(title=title, url=url, date=pubdate,
                    description=description,
                    content=''))
        ans = self.sort_index_by(ans, {'The Front Page':-1, 'Dining In, Dining Out':1,
→ 'Obituaries':2})
        ans = [(key, articles[key]) for key in ans if key in articles]
        return ans

    def preprocess_html(self, soup):
        refresh = soup.find('meta', {'http-equiv': 'refresh'})
        if refresh is None:
            return soup
        content = refresh.get('content').partition('=')[2]
        raw = self.browser.open('https://www.nytimes.com'+content).read()
        return BeautifulSoup(raw.decode('cp1252', 'replace'))

```

Przedstawione tu zostaną nowe funkcje. Po pierwsze:

```
timefmt = ' [%a, %d %b, %Y]'
```

Pozwala to ustawić format czasu na okładce tworzonych ebooków na: Dzień, Numer_dnia Miesiąc, Rok. Więcej o *timefmt* (strona 48).

Znajduje się tu również grupa poleceń, służących do oczyszczenia pobranego HTMLa:

```

remove_tags_before = dict(name='h1')
remove_tags_after  = dict(id='footer')
remove_tags = ...

```

Usuwać one wszystko przez pierwszym znacznikiem <h1> i wszystko po pierwszym znaczniku, który ma id footer. Więcej informacji na ten temat *remove_tags* (strona 47), *remove_tags_before* (strona 47), *remove_tags_after* (strona 47).

Następna ciekawostka:

```

needs_subscription = True
...
def get_browser(self):
    ...

```

`needs_subscription = True` tells calibre that this recipe needs a username and password in order to access the content. This causes, calibre to ask for a username and password whenever you try to use this recipe. The code in

`calibre.web.feeds.news.BasicNewsRecipe.get_browser()` (strona 40) actually does the login into the NYT website. Once logged in, calibre will use the same, logged in, browser instance to fetch all content. See [mechanize](#)⁵ to understand the code in `get_browser`.

The next new feature is the `calibre.web.feeds.news.BasicNewsRecipe.parse_index()` (strona 42) method. Its job is to go to <https://www.nytimes.com/pages/todayspaper/index.html> and fetch the list of articles that appear in *today's* paper. While more complex than simply using [RSS](#), the recipe creates an e-book that corresponds very closely to the days paper. `parse_index` makes heavy use of [BeautifulSoup](#)⁶ to parse the daily paper webpage. You can also use other, more modern parsers if you dislike BeautifulSoup. calibre comes with [lxml](#)⁷ and [html5lib](#)⁸, which are the recommended parsers. To use them, replace the call to `index_to_soup()` with the following:

```
raw = self.index_to_soup(url, raw=True)
# For html5lib
import html5lib
root = html5lib.parse(raw, namespaceHTMLElements=False, treebuilder='lxml')
# For the lxml html 4 parser
from lxml import html
root = html.fromstring(raw)
```

Na koniec jeszcze jedna nowa funkcja - `calibre.web.feeds.news.BasicNewsRecipe.preprocess_html()` (strona 43). Jest ona używana do wprowadzania zmian w każdej pobranej stronie HTML. Tu została użyta do pominięcia reklam, które witryna wyświetla przed każdym artykułem.

2.3 Wskazówki przydatne przy tworzeniu nowych źródeł

The best way to develop new recipes is to use the command line interface. Create the recipe using your favorite Python editor and save it to a file say `myrecipe.recipe`. The `.recipe` extension is required. You can download content using this recipe with the command:

```
ebook-convert myrecipe.recipe .epub --test -vv --debug-pipeline debug
```

The command **ebook-convert** will download all the webpages and save them to the EPUB file `myrecipe.epub`. The `-vv` option makes `ebook-convert` spit out a lot of information about what it is doing. The `ebook-convert-recipe-input --test` option makes it download only a couple of articles from at most two feeds. In addition, `ebook-convert` will put the downloaded HTML into the `debug/input` folder, where `debug` is the folder you specified in the `ebook-convert --debug-pipeline` (strona 335) option.

Once the download is complete, you can look at the downloaded [HTML](#) by opening the file `debug/input/index.html` in a browser. Once you're satisfied that the download and preprocessing is happening correctly, you can generate e-books in different formats as shown below:

```
ebook-convert myrecipe.recipe myrecipe.epub
ebook-convert myrecipe.recipe myrecipe.mobi
...
```

If you're satisfied with your recipe, and you feel there is enough demand to justify its inclusion into the set of built-in recipes, post your recipe in the [calibre recipes forum](#)⁹ to share it with other calibre users.

Informacja: W systemie macOS narzędzia wiersza poleceń znajdują się w pakiecie kalibru, na

⁵ <https://mechanize.readthedocs.io/en/latest/>

⁶ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

⁷ <https://lxml.de/>

⁸ <https://github.com/html5lib/html5lib-python>

⁹ <https://www.mobileread.com/forums/forumdisplay.php?f=228>

przykład, jeśli zainstalowałeś calibre w `/Applications`, narzędzia wiersza poleceń znajdują się w `:file:/Applications/calibre.app/Contents/MacOS/``.

Zobacz także:

ebook-convert (strona 326)

Interfejs wiersza poleceń do wszystkich konwersji e-booków.

2.4 Dalsza lektura

Aby dowiedzieć się więcej o pisaniu zaawansowanych źródeł przy użyciu narzędzi, dostępnych w `BasicNewsRecipe`, możesz skorzystać z poniższych źródeł:

API documentation (strona 39)

Dokumentacja klasy `BasicNewsRecipe` i wszystkich najważniejszych metod i właściwości.

*BasicNewsRecipe*¹⁰

Kod źródłowy `BasicNewsRecipe`

*Wbudowane źródła*¹¹

Kod źródeł dostarczanych z calibre

*The calibre recipes forum*¹²

Tu znajdziesz wiele osób, które wiedzą bardzo dużo o tworzeniu źródeł.

2.5 Dokumentacja API

2.5.1 API documentation for recipes

API, służące do pisania źródeł jest zdefiniowane przez

class `calibre.web.feeds.news.BasicNewsRecipe` (*options, log, progress_reporter*)

Podstawowa klasa, zawierająca logikę, niezbędną we wszystkich źródłach. Im bardziej przeciążysz funkcje tej klasy, tym bardziej dopasujesz je do źródła wiadomości. Wprowadzenie do tworzenia źródeł znajdziesz w

abort_article (*msg=None*)

Call this method inside any of the preprocess methods to abort the download for the current article. Useful to skip articles that contain inappropriate content, such as pure video articles.

abort_recipe_processing (*msg*)

Powoduje, że pobieranie treści zostaje przerwane, i wyświetla komunikaty zwrotne.

add_toc_thumbnail (*article, src*)

Call this from `populate_article_metadata` with the `src` attribute of an `` tag from the article that is appropriate for use as the thumbnail representing the article in the Table of Contents. Whether the thumbnail is actually used is device dependent (currently only used by the Kindles). Note that the referenced image must be one that was successfully downloaded, otherwise it will be ignored.

¹⁰ <https://github.com/kovidgoyal/calibre/blob/master/src/calibre/web/feeds/news.py>

¹¹ <https://github.com/kovidgoyal/calibre/tree/master/recipes>

¹² <https://www.mobileread.com/forums/forumdisplay.php?f=228>

classmethod `adeify_images` (*soup*)

If your recipe when converted to EPUB has problems with images when viewed in Adobe Digital Editions, call this method from within `postprocess_html()` (strona 42).

canonicalize_internal_url (*url*, *is_link=True*)

Return a set of canonical representations of *url*. The default implementation uses just the server hostname and path of the URL, ignoring any query parameters, fragments, etc. The canonical representations must be unique across all URLs for this news source. If they are not, then internal links may be resolved incorrectly.

Parametry

is_link – Is True if the URL is coming from an internal link in an HTML file. False if the URL is the URL used to download an article.

cleanup ()

Called after all articles have been download. Use it to do any cleanup like logging out of subscription sites, etc.

clone_browser (*br*)

Clone the browser *br*. Cloned browsers are used for multi-threaded downloads, since mechanize is not thread safe. The default cloning routines should capture most browser customization, but if you do something exotic in your recipe, you should override this method in your recipe and clone manually.

Cloned browser instances use the same, thread-safe CookieJar by default, unless you have customized cookie handling.

default_cover (*cover_file*)

Stwórz domyślną okładkę dla źródeł, które nie mają własnej.

download ()

Download and pre-process all articles from the feeds in this recipe. This method should be called only once on a particular Recipe instance. Calling it more than once will lead to undefined behavior. :return: Path to index.html

extract_readable_article (*html*, *url*)

Extracts main article content from «html», cleans up and returns as a (article_html, extracted_title) tuple. Based on the original readability algorithm by Arc90.

get_article_url (*article*)

Override in a subclass to customize extraction of the [URL](#) that points to the content for each article. Return the article URL. It is called with *article*, an object representing a parsed article from a feed. See [feedparser](#)¹³. By default it looks for the original link (for feeds syndicated via a service like FeedBurner or Pheedo) and if found, returns that or else returns [article.link](#)¹⁴.

get_browser (**args*, ***kwargs*)

Return a browser instance used to fetch documents from the web. By default it returns a [mechanize](#)¹⁵ browser instance that supports cookies, ignores robots.txt, handles refreshes and has a mozilla firefox user agent.

If your recipe requires that you login first, override this method in your subclass. For example, the following code is used in the New York Times recipe to login for full access:

```
def get_browser(self):
    br = BasicNewsRecipe.get_browser(self)
    if self.username is not None and self.password is not None:
        br.open('https://www.nytimes.com/auth/login')
        br.select_form(name='login')
        br['USERID'] = self.username
        br['PASSWORD'] = self.password
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

        br.submit()
    return br

```

get_cover_url()

Return a [URL](#) to the cover image for this issue or *None*. By default it returns the value of the member *self.cover_url* which is normally *None*. If you want your recipe to download a cover for the e-book override this method in your subclass, or set the member variable *self.cover_url* before this method is called.

get_extra_css()

By default returns *self.extra_css*. Override if you want to programmatically generate the extra_css.

get_feeds()

Return a list of [RSS](#) feeds to fetch for this profile. Each element of the list must be a 2-element tuple of the form (title, url). If title is *None* or an empty string, the title from the feed is used. This method is useful if your recipe needs to do some processing to figure out the list of feeds to download. If so, override in your subclass.

get_masthead_title()

Override in subclass to use something other than the recipe title

get_masthead_url()

Return a [URL](#) to the masthead image for this issue or *None*. By default it returns the value of the member *self.masthead_url* which is normally *None*. If you want your recipe to download a masthead for the e-book override this method in your subclass, or set the member variable *self.masthead_url* before this method is called. Masthead images are used in Kindle MOBI files.

get_obfuscated_article(url)

If you set *articles_are_obfuscated* this method is called with every article URL. It should return the path to a file on the filesystem that contains the article HTML. That file is processed by the recursive HTML fetching engine, so it can contain links to pages/images on the web. Alternately, you can return a dictionary of the form: {«data»: <HTML data>, «url»: <the resolved URL of the article>}. This avoids needing to create temporary files. The *url* key in the dictionary is useful if the effective URL of the article is different from the URL passed into this method, for example, because of redirects. It can be omitted if the URL is unchanged.

This method is typically useful for sites that try to make it difficult to access article content automatically.

get_url_specific_delay(url)

Return the delay in seconds before downloading this URL. If you want to programmatically determine the delay for the specified URL, override this method in your subclass, returning *self.delay* by default for URLs you do not want to affect.

Zwraca

A floating point number, the delay in seconds.

classmethod image_url_processor(baseurl, url)

Perform some processing on image urls (perhaps removing size restrictions for dynamically generated images, etc.) and return the preprocessed URL. Return *None* or an empty string to skip fetching the image.

index_to_soup(url_or_raw, raw=False, as_tree=False, save_raw=None)

Convenience method that takes an URL to the index page and returns a [BeautifulSoup](#)¹⁶ of it.

url_or_raw: Either a URL or the downloaded index page as a string

is_link_wanted(url, tag)

Return True if the link should be followed or False otherwise. By default, raises *NotImplementedError* which causes the downloader to ignore it.

Parametry

- **url** – The URL to be followed
- **tag** – The tag from which the URL was derived

parse_feeds()

Create a list of articles from the list of feeds returned by `BasicNewsRecipe.get_feeds()` (strona 41). Return a list of Feed objects.

parse_index()

This method should be implemented in recipes that parse a website instead of feeds to generate a list of articles. Typical uses are for news sources that have a „Print Edition” webpage that lists all the articles in the current print edition. If this function is implemented, it will be used in preference to `BasicNewsRecipe.parse_feeds()` (strona 42).

It must return a list. Each element of the list must be a 2-element tuple of the form ('feed title', list of articles).

Each list of articles must contain dictionaries of the form:

```
{
  'title'      : article title,
  'url'        : URL of print version,
  'date'       : The publication date of the article as a string,
  'description': A summary of the article
  'content'    : The full article (can be an empty string). Obsolete
                 do not use, instead save the content to a temporary
                 file and pass a file:///path/to/temp/file.html as
                 the URL.
}
```

For an example, see the recipe for downloading *The Atlantic*. In addition, you can add «author» for the author of the article.

If you want to abort processing for some reason and have calibre show the user a simple message instead of an error, call `abort_recipe_processing()` (strona 39).

populate_article_metadata(article, soup, first)

Called when each HTML page belonging to article is downloaded. Intended to be used to get article metadata like author/summary/etc. from the parsed HTML (soup).

Parametry

- **article** – A object of class `calibre.web.feeds.Article`. If you change the summary, remember to also change the `text_summary`
- **soup** – Parsed HTML belonging to this article
- **first** – True iff the parsed HTML is the first page of the article.

postprocess_book(oeb, opts, log)

Run any needed post processing on the parsed downloaded e-book.

Parametry

- **oeb** – An OEBBook object
- **opts** – Conversion options

postprocess_html (*soup*, *first_fetch*)

This method is called with the source of each downloaded *HTML* file, after it is parsed for links and images. It can be used to do arbitrarily powerful post-processing on the *HTML*. It should return *soup* after processing it.

Parametry

- **soup** – A BeautifulSoup¹⁷ instance containing the downloaded *HTML*.
- **first_fetch** – True if this is the first page of an article.

preprocess_html (*soup*)

This method is called with the source of each downloaded *HTML* file, before it is parsed for links and images. It is called after the cleanup as specified by `remove_tags` etc. It can be used to do arbitrarily powerful pre-processing on the *HTML*. It should return *soup* after processing it.

soup: A BeautifulSoup¹⁸ instance containing the downloaded *HTML*.

preprocess_image (*img_data*, *image_url*)

Perform some processing on downloaded image data. This is called on the raw data before any resizing is done. Must return the processed raw data. Return None to skip the image.

preprocess_raw_html (*raw_html*, *url*)

This method is called with the source of each downloaded *HTML* file, before it is parsed into an object tree. *raw_html* is a unicode string representing the raw HTML downloaded from the web. *url* is the URL from which the HTML was downloaded.

Note that this method acts *before* `preprocess_regexps`.

This method must return the processed *raw_html* as a unicode object.

classmethod print_version (*url*)

Take a *url* pointing to the webpage with article content and return the *URL* pointing to the print version of the article. By default does nothing. For example:

```
def print_version(self, url):
    return url + '?&pagewanted=print'
```

publication_date ()

Use this method to set the date when this issue was published. Defaults to the moment of download. Must return a `datetime.datetime` object.

skip_ad_pages (*soup*)

This method is called with the source of each downloaded *HTML* file, before any of the cleanup attributes like `remove_tags`, `keep_only_tags` are applied. Note that `preprocess_regexps` will have already been applied. It is meant to allow the recipe to skip ad pages. If the soup represents an ad page, return the HTML of the real page. Otherwise return None.

soup: A BeautifulSoup¹⁹ instance containing the downloaded *HTML*.

sort_index_by (*index*, *weights*)

Convenience method to sort the titles in *index* according to *weights*. *index* is sorted in place. Returns *index*.

index: A list of titles.

weights: A dictionary that maps weights to titles. If any titles in *index* are not in *weights*, they are assumed to have a weight of 0.

classmethod tag_to_string (*tag*, *use_alt=True*, *normalize_whitespace=True*)

Convenience method to take a BeautifulSoup²⁰ Tag and extract the text from it recursively, including any CDATA sections and alt tag attributes. Return a possibly empty Unicode string.

use_alt: If *True* try to use the alt attribute for tags that don't have any textual content

tag: BeautifulSoup²¹ Tag

articles_are_obfuscated = False

Set to True and implement *get_obfuscated_article()* (strona 41) to handle websites that try to make it difficult to scrape content.

auto_cleanup = False

Automatically extract all the text from downloaded article pages. Uses the algorithms from the readability project. Setting this to True, means that you do not have to worry about cleaning up the downloaded HTML manually (though manual cleanup will always be superior).

auto_cleanup_keep = None

Specify elements that the auto cleanup algorithm should never remove. The syntax is a XPath expression. For example:

```
auto_cleanup_keep = '//div[@id="article-image"]' will keep all divs with
                                                         id="article-image"
auto_cleanup_keep = '//*[[@class="important"]]' will keep all elements
                                                         with class="important"
auto_cleanup_keep = '//div[@id="article-image"]|//span[@class="important"]'
will keep all divs with id="article-image" and spans
with class="important"
```

center_navbar = True

If True the navigation bar is center aligned, otherwise it is left aligned

compress_news_images = False

Set this to False to ignore all scaling and compression parameters and pass images through unmodified. If True and the other compression parameters are left at their default values, JPEG images will be scaled to fit in the screen dimensions set by the output profile and compressed to size at most $(w * h)/16$ where $w \times h$ are the scaled image dimensions.

compress_news_images_auto_size = 16

The factor used when auto compressing JPEG images. If set to None, auto compression is disabled. Otherwise, the images will be reduced in size to $(w * h)/\text{compress_news_images_auto_size}$ bytes if possible by reducing the quality level, where $w \times h$ are the image dimensions in pixels. The minimum JPEG quality will be 5/100 so it is possible this constraint will not be met. This parameter can be overridden by the parameter *compress_news_images_max_size* which provides a fixed maximum size for images. Note that if you enable *scale_news_images_to_device* then the image will first be scaled and then its quality lowered until its size is less than $(w * h)/\text{factor}$ where w and h are now the *scaled* image dimensions. In other words, this compression happens after scaling.

compress_news_images_max_size = None

Set JPEG quality so images do not exceed the size given (in KBytes). If set, this parameter overrides auto compression via *compress_news_images_auto_size*. The minimum JPEG quality will be 5/100 so it is possible this constraint will not be met.

conversion_options = {}

Recipe specific options to control the conversion of the downloaded content into an e-book. These will override any user or plugin specified values, so only use if absolutely necessary. For example:


```
conversion_options = {
    'base_font_size' : 16,
    'linearize_tables' : True,
}
```

cover_margins = (0, 0, '#ffffff')

By default, the cover image returned by `get_cover_url()` will be used as the cover for the periodical. Overriding this in your recipe instructs calibre to render the downloaded cover into a frame whose width and height are expressed as a percentage of the downloaded cover. `cover_margins = (10, 15, «#ffffff»)` pads the cover with a white margin 10px on the left and right, 15px on the top and bottom. Color names are defined [here](#)²². Note that for some reason, white does not always work in Windows. Use `#ffffff` instead

delay = 0

The default delay between consecutive downloads in seconds. The argument may be a floating point number to indicate a more precise time. See `get_url_specific_delay()` (strona 41) to implement per URL delays.

description = ''

A couple of lines that describe the content this recipe downloads. This will be used primarily in a GUI that presents a list of recipes.

encoding = None

Specify an override encoding for sites that have an incorrect charset specification. The most common being specifying `latin1` and using `cp1252`. If `None`, try to detect the encoding. If it is a callable, the callable is called with two arguments: The recipe object and the source to be decoded. It must return the decoded source.

extra_css = None

Specify any extra [CSS](#) that should be added to downloaded [HTML](#) files. It will be inserted into `<style>` tags, just before the closing `</head>` tag thereby overriding all [CSS](#) except that which is declared using the style attribute on individual [HTML](#) tags. Note that if you want to programmatically generate the `extra_css` override the `get_extra_css()` (strona 41) method instead. For example:

```
extra_css = '.heading { font: serif x-large }'
```

feeds = None

List of feeds to download. Can be either `[url1, url2, ...]` or `[('title1', url1), ('title2', url2), ...]`

filter_regexps = []

List of regular expressions that determines which links to ignore. If empty it is ignored. Used only if `is_link_wanted` is not implemented. For example:

```
filter_regexps = [r'ads\.doubleclick\.net']
```

will remove all URLs that have `ads.doubleclick.net` in them.

Only one of `BasicNewsRecipe.match_regexps` (strona 46) or `BasicNewsRecipe.filter_regexps` (strona 45) should be defined.

handle_gzip = True

Set to `False` if you do not want to use gzipped transfers. Note that some old servers flake out with `gzip`

ignore_duplicate_articles = None

Ignore duplicates of articles that are present in more than one section. A duplicate article is an article that has the same title and/or URL. To ignore articles with the same title, set this to:

```
ignore_duplicate_articles = {'title'}
```

To use URLs instead, set it to:

```
ignore_duplicate_articles = {'url'}
```

To match on title or URL, set it to:

```
ignore_duplicate_articles = {'title', 'url'}
```

keep_only_tags = []

Keep only the specified tags and their children. For the format for specifying a tag see *BasicNewsRecipe.remove_tags* (strona 47). If this list is not empty, then the `<body>` tag will be emptied and re-filled with the tags that match the entries in this list. For example:

```
keep_only_tags = [dict(id=['content', 'heading'])]
```

will keep only tags that have an *id* attribute of „content” or „heading”.

language = 'und'

The language that the news is in. Must be an ISO-639 code either two or three characters long

masthead_url = None

By default, calibre will use a default image for the masthead (Kindle only). Override this in your recipe to provide a URL to use as a masthead.

match_regexps = []

List of regular expressions that determines which links to follow. If empty, it is ignored. Used only if `is_link_wanted` is not implemented. For example:

```
match_regexps = [r'page=[0-9]+']
```

will match all URLs that have *page=some number* in them.

Only one of *BasicNewsRecipe.match_regexps* (strona 46) or *BasicNewsRecipe.filter_regexps* (strona 45) should be defined.

max_articles_per_feed = 100

Maximum number of articles to download from each feed. This is primarily useful for feeds that don't have article dates. For most feeds, you should use *BasicNewsRecipe.oldest_article* (strona 46)

needs_subscription = False

If True the GUI will ask the user for a username and password to use while downloading. If set to „optional” the use of a username and password becomes optional

no_stylesheets = False

Convenient flag to disable loading of stylesheets for websites that have overly complex stylesheets unsuitable for conversion to e-book formats. If True stylesheets are not downloaded and processed

oldest_article = 7.0

Oldest article to download from this news source. In days.

preprocess_regexps = []

List of *regex* substitution rules to run on the downloaded *HTML*. Each element of the list should be a two element tuple. The first element of the tuple should be a compiled regular expression and the second a callable that takes a single match object and returns a string to replace the match. For example:

```
preprocess_regexps = [
    (re.compile(r'<!--Article ends here-->.*</body>', re.DOTALL|re.IGNORECASE),
     lambda match: '</body>'),
]
```

will remove everything from `<!--Article ends here-->` to `</body>`.

publication_type = 'unknown'

Publication type Set to newspaper, magazine or blog. If set to None, no publication type metadata will be written to the opf file.

recipe_disabled = None

Set to a non empty string to disable this recipe. The string will be used as the disabled message

recursions = 0

Number of levels of links to follow on article webpages

remove_attributes = []

List of attributes to remove from all tags. For example:

```
remove_attributes = ['style', 'font']
```

remove_empty_feeds = False

If True empty feeds are removed from the output. This option has no effect if `parse_index` is overridden in the sub class. It is meant only for recipes that return a list of feeds using `feeds` or `get_feeds()` (strona 41). It is also used if you use the `ignore_duplicate_articles` option.

remove_javascript = True

Convenient flag to strip all JavaScript tags from the downloaded HTML

remove_tags = []

List of tags to be removed. Specified tags are removed from downloaded HTML. A tag is specified as a dictionary of the form:

```
{
    name      : 'tag name',    #e.g. 'div'
    attrs     : a dictionary,  #e.g. {'class': 'advertisement'}
}
```

All keys are optional. For a full explanation of the search criteria, see [Beautiful Soup](#)²³ A common example:

```
remove_tags = [dict(name='div', class_='advert')]
```

This will remove all `<div class="advert">` tags and all their children from the downloaded [HTML](#).

remove_tags_after = None

Remove all tags that occur after the specified tag. For the format for specifying a tag see [BasicNewsRecipe.remove_tags](#) (strona 47). For example:

```
remove_tags_after = [dict(id='content')]
```

will remove all tags after the first element with `id="content"`.

remove_tags_before = None

Remove all tags that occur before the specified tag. For the format for specifying a tag see [BasicNewsRecipe.remove_tags](#) (strona 47). For example:

```
remove_tags_before = dict(id='content')
```

will remove all tags before the first element with *id*="content".

requires_version = (0, 6, 0)

Minimum calibre version needed to use this recipe

resolve_internal_links = False

If set to True then links in downloaded articles that point to other downloaded articles are changed to point to the downloaded copy of the article rather than its original web URL. If you set this to True, you might also need to implement *canonicalize_internal_url()* (strona 40) to work with the URL scheme of your particular website.

reverse_article_order = False

Odwróć porządek artykułów w każdym strumieniu

scale_news_images = None

Maximum dimensions (w,h) to scale images to. If *scale_news_images_to_device* is True this is set to the device screen dimensions set by the output profile unless there is no profile set, in which case it is left at whatever value it has been assigned (default None).

scale_news_images_to_device = True

Rescale images to fit in the device screen dimensions set by the output profile. Ignored if no output profile is set.

simultaneous_downloads = 5

Number of simultaneous downloads. Set to 1 if the server is picky. Automatically reduced to 1 if *BasicNewsRecipe.delay* (strona 45) > 0

summary_length = 500

Max number of characters in the short description

```
template_css = '\n .article_date {\n color: gray; font-family: monospace;\n }\n\n .article_description {\n text-indent: 0pt;\n }\n\n a.article {\n font-weight: bold; text-align:left;\n }\n\n a.feed {\n font-weight: bold;\n }\n\n .calibre_navbar {\n font-family:monospace;\n }\n '
```

The CSS that is used to style the templates, i.e., the navigation bars and the Tables of Contents. Rather than overriding this variable, you should use *extra_css* in your recipe to customize look and feel.

timefmt = ' [%a, %d %b %Y] '

The format string for the date shown on the first page. By default: Day_Name, Day_Number Month_Name Year

timeout = 120.0

Timeout for fetching files from server in seconds

title = 'Nieznane źródło informacji'

The title to use for the e-book

use_embedded_content = None

Normally we try to guess if a feed has full articles embedded in it based on the length of the embedded content. If *None*, then the default guessing is used. If *True* then the we always assume the feeds has embedded content and if *False* we always assume the feed does not have embedded content.

-
- ¹³ <https://pythonhosted.org/feedparser/>
 - ¹⁴ <https://pythonhosted.org/feedparser/reference-entry-link.html>
 - ¹⁵ <https://mechanize.readthedocs.io/en/latest/>
 - ¹⁶ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
 - ¹⁷ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
 - ¹⁸ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
 - ¹⁹ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
 - ²⁰ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
 - ²¹ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
 - ²² <https://www.imagemagick.org/script/color.php>
 - ²³ <https://www.crummy.com/software/BeautifulSoup/bs4/doc/#searching-the-tree>

The E-book viewer

calibre includes a built-in E-book viewer that can view all the major e-book formats. The E-book viewer is highly customizable and has many advanced features.

- *Starting the E-book viewer* (strona 52)
- *Navigating around an e-book* (strona 52)
- *Highlighting text* (strona 53)
- *Czytać na głos* (strona 53)
- *Searching the text* (strona 54)
- *Following links using only the keyboard* (strona 54)
- *Konfigurowanie wyglądu czytnika* (strona 54)
- *Wyszukiwanie w słowniku* (strona 55)
- *Kopiowanie tekstu i grafiki* (strona 55)
- *Zooming in on images* (strona 55)
- *Skróty klawiszowe* (strona 55)
- *Non re-flowable content* (strona 57)
- *Designing your book to work well with the calibre viewer* (strona 58)

3.1 Starting the E-book viewer

You can view any of the books in your calibre library by selecting the book and pressing the *View* button. This will open up the book in the E-book viewer. You can also launch the E-book viewer by itself from the Start menu in Windows. On macOS, you can pin it to the dock and launch it from there. On Linux you can use its launcher in the desktop menus or run the command **ebook-viewer**.

3.2 Navigating around an e-book

You can „turn pages” in a book by either:

- Clicking in the left or right margin or the page with the mouse
- Pressing the `spacebar`, `page up`, `page down` or arrow keys
- On a touchscreen tapping on the text or swiping left and right

You can access the viewer controls by either:

- Right clicking on the text
- Pressing the `Esc` or `Menu` keys
- On a touchscreen by tapping the top 1/3rd of the screen

The viewer has two modes, „paged” and „flow”. In paged mode the book content is presented as pages, similar to a paper book. In flow mode the text is presented continuously, like in a web browser. You can switch between them using the viewer *Preferences* under *Page layout* or by pressing the `Ctrl+M` key.

3.2.1 Zakładki

When you are in the middle of a book and close the E-book viewer, it will remember where you stopped reading and return there the next time you open the book. You can also set bookmarks in the book by using the *Bookmarks* button in the E-book viewer controls or pressing `Ctrl+B`. When viewing EPUB format books, these bookmarks are actually saved in the EPUB file itself. You can add bookmarks, then send the file to a friend. When they open the file, they will be able to see your bookmarks. You can turn off this behavior in the *Miscellaneous* section of the viewer preferences.

3.2.2 Spis treści

If the book you are reading defines a Table of Contents, you can access it by pressing the *Table of Contents* button. This will bring up a list of sections in the book. You can click on any of them to jump to that portion of the book.

3.2.3 Używanie lokacji

E-books, unlike paper books, have no concept of pages. You can refer to precise locations in e-books using the *Go to → Location* functionality in the viewer controls.

You can use this location information to unambiguously refer to parts of the books when discussing it with friends or referring to it in other works. You can enter these locations under *Go to → Location* in the viewer controls.

There is a URL you can copy to the clipboard and paste into other programs or documents. Clicking on this URL will open the book in the calibre E-book viewer at the current location.

If you click on links inside the e-book to take you to different parts of the book, such as an endnote, you can use the *Back* and *Forward* buttons in the top left corner of the viewer controls. These buttons behave just like those in a web browser.

3.2.4 Tryb odwołań

calibre also has a very handy *Reference mode*. You can turn it on by clicking the *Reference mode* button in the viewer controls. Once you do this, every paragraph will have a unique number displayed at the start, made up of the section and paragraph numbers.

You can use this number to unambiguously refer to parts of the books when discussing it with friends or referring to it in other works. You can enter these numbers in the *Go to function* to navigate to a particular reference location.

3.3 Highlighting text

When you select text in the viewer, a little popup bar appears next to the selection. You can click the highlight button in that bar to create a highlight. You can add notes and change the color of the highlight. On a touch screen, long tap a word to select it and show the popup bar. Once in highlight mode you can change what text is selected, using touch screen friendly selection handles. Drag the handles to the top or bottom margins to scroll while selecting. You can also `Shift+click` or `right click` to extend the selection, particularly useful for multi-page selections.

You can use the *Highlights* button in the viewer controls to show a separate panel with a list of all highlights in the book, sorted by chapter.

You can browse *all highlights* in your entire calibre library by right clicking the *View* button and choosing *Browse annotations*.

Finally, if you use the calibre Content server's in browser viewer, you can have the viewer sync its annotations with the browser viewer by going to *Preferences* → *Miscellaneous* in the viewer preferences and entering the username of the Content server viewer to sync with. Use the special value `*` to sync with anonymous users.

3.4 Czytać na głos

The viewer can read book text aloud. To use it you can simply click the *Read aloud* button in the viewer controls to start reading book text aloud. The word being currently read is highlighted. Speech is synthesized from the text using your operating system services for text-to-speech. You can change the voice being used by clicking the gear icon in the bar that is displayed while *Read aloud* is active.

You can also read aloud highlighted passages by adding the *Read aloud* button to the selection bar in the viewer preferences under *Selection behavior*.

Informacja: Support for text-to-speech in browsers is very incomplete and bug-ridden so how well *Read aloud* will work in the in-browser viewer is dependent on how well the underlying browser supports text-to-speech. In particular, highlighting of current word does not work, and changing speed or voice will cause reading to start again from the beginning.

Informacja: On Linux, *Read aloud* requires [Speech Dispatcher](https://freebsoft.org/speechd)²⁴ to be installed and working.

²⁴ <https://freebsoft.org/speechd>

Informacja: On Windows, not all installed voices may be visible to the SAPI sub-system that is used for text-to-speech. There are [instructions to make all voices visible](#)²⁵.

3.5 Searching the text

The viewer has very powerful search capabilities. Press the `Ctrl+F` key or access the viewer controls and click search. The simplest form of searching is to just search for whatever text you enter in the text box. The different forms of searching are chosen by the search mode box below the search input. Available modes are:

1. *Contains* - The simplest default mode. The text entered in the search box is searched for anywhere. All punctuation, accents and spaces are ignored. For example, the search: `Pena` will match all of the following: `penal`, `pen a`, `pen.a` and `Peña`. If you select the *Case sensitive* box then accents, spaces and punctuation are no longer ignored.
2. *Whole words* - Searches for whole words. So for example, the search `pena` will match the word `Peña` but not the word `Penal`. As with *Contains* searches above, accents and punctuation are ignored unless the *Case sensitive* box is checked.
3. *Nearby words* - Searches for whole words that are near each other. So for example, the search `calibre cool` will match places where the words `calibre` and `cool` occur within sixty characters of each other. To change the number of characters add the new number to the end of the list of words. For instance, `calibre cool awesome 120` will match places where the three words occur within 120 characters of each other. Note that punctuation and accents are *not* ignored for these searches.
4. *Regex* - Interprets the search text as a *regular expression*. To learn more about using regular expressions, see [the tutorial](#) (strona 213).

3.6 Following links using only the keyboard

The E-book viewer has a *Hints mode* that allows you to click links in the text without using the mouse. Press the `Alt+F` key and all links in the current screen will be highlighted with a number or letter over them. Press the letter on your keyboard to click the link. Pressing the `Esc` key will abort the *Hints mode* without selecting any link.

If more than thirty five links are on-screen then some of them will have multiple letters, in which case type the first and second, or the first and press `Enter` to activate. You can also use the `Backspace` key to undo a mistake in typing.

3.7 Konfigurowanie wyglądu czytnika

You can change font sizes on the fly by using *Font size* in the viewer controls or `Ctrl++` or `Ctrl+-` or holding the `Ctrl` key and using the mouse wheel.

Colors can be changed in the *Colors* section of the viewer preferences.

You can change the number of pages displayed on the screen as well as page margins in *Page layout* in the viewer preferences.

You can display custom headers and footers such as time left to read, current chapter title, book position, etc. via the *Headers and footers* section of the viewer preferences.

²⁵ <https://www.mobileread.com/forums/showpost.php?p=4084051&postcount=108>

More advanced customization can be achieved by the *Styles* settings. Here you can specify a background image to display under the text and also a stylesheet you can set that will be applied to every book. Using it you can do things like change paragraph styles, text justification, etc. For examples of custom stylesheets used by calibre's users, see [the forums](#)²⁶.

3.8 Wyszukiwanie w słowniku

You can look up the meaning of words in the current book by double clicking or long tapping the word you want to lookup and then clicking the lookup button that looks like a library.

3.9 Kopiowanie tekstu i grafiki

You can select text and images by dragging the content with your mouse and then right clicking and selecting *Copy* to copy to the clipboard. The copied material can be pasted into another application as plain text and images.

3.10 Zooming in on images

You can zoom in to show an image at full size in a separate window by either double clicking or long tapping on it. You can also right click on it and choose *View image*.

3.11 Skróty klawiszowe

The viewer has extensive keyboard shortcuts, like the rest of calibre. They can be customised in the viewer *Preferences*. The default shortcuts are listed below:

Tabela 1: Keyboard shortcuts for the calibre viewer

Keyboar shortcut	Działanie
Home, Ctrl+A	Scroll to the start of the current file in a multi file book
Ctrl+A	
Ctrl+H	Przewiń do początku książki
Ctrl+E	Przewiń do końca książki
End, Ctrl+A	Scroll to the end of the current file in a multi file book
Ctrl+A	
ArrowU	Przewijaj do tyłu, płynnie w trybie przepływu i po pełnym ekranie w trybie stronicowania
ArrowD	Przewijaj do przodu, płynnie w trybie przepływu i przez pełny ekran w trybie stronicowania
ArrowL	Przewiń w lewo trochę w trybie przepływu i stronę w trybie stronicowania
ArrowR	Przewiń w prawo trochę w trybie przepływu i stronę w trybie stronicowania

ciąg dalszy na następnej stronie

²⁶ <https://www.mobileread.com/forums/showthread.php?t=51500>

Tabela 1 – kontynuacja poprzedniej strony

Keyboard shortcut	Działanie
PageUp	Przewiń do tyłu o pełne ekrany
Shift+PageDo	Przewiń do przodu o pełny ekran
Spaceb	
Ctrl+P	Przewiń do poprzedniej sekcji
Ctrl+P	Przewiń do następnej sekcji
Alt+Ar	Wstecz
Alt+Ar	Powtórz
Ctrl+T	Toggle Table of Contents
Ctrl+S	Czytać na głos
Alt+P	Change settings quickly by creating and switching to <i>profiles</i>
Alt+f	Podążaj za linkami za pomocą klawiatury
Ctrl+C	Kopiuj do schowka
Alt+C	Skopiuj bieżącą lokalizację do schowka
Ctrl+S	Skopiuj bieżącą lokalizację jako calibre:// URL do schowka
/,	Szukaj
Ctrl+f	
Cmd+f	
F3,	Znajdź następny
Enter	
Shift+	Znajdź poprzedni
Shift+	
Ctrl+P	Zwiększ rozmiar czcionki
Meta+P	
Ctrl+M	Zmniejsz rozmiar czcionki
Meta+M	
Ctrl+0	Przywróć domyślny rozmiar czcionki
Ctrl+]	Zwiększ liczbę stron na ekranie
Ctrl+[Zmniejsz liczbę stron na ekranie
Ctrl+A	Ustaw liczbę stron na ekranie automatycznie
F11,	Przełącz na tryb pełnoekranowy
Ctrl+S	
Ctrl+M	Przełącz tekst między trybem stron a trybem ciągłym
Ctrl+W	Przełącz pasek przewijania
Ctrl+X	Przełącz tryb odniesienia
Ctrl+B	Pokaż/ukryj zakładki
Ctrl+A	Nowa zakładka
Ctrl+N	Pokaż metadane książki
Ctrl+E	
Ctrl+A	Załaduj ponownie
Ctrl+A	
Ctrl+S	Zmień bieżący wybór w przód o słowo

ciąg dalszy na następnej stronie

Tabela 1 – kontynuacja poprzedniej strony

Keyboard shortcut	Działanie
Ctrl+S	Zmień bieżący wybór wstecz o słowo
Shift+	Zmień bieżący wybór do przodu o znak
Shift+	Zmień bieżący wybór wstecz o znak
Shift+	Zmień bieżący wybór do przodu o linię
Shift+	Rozszerz bieżący wybór do początku linii
Shift+	Rozszerz bieżący wybór do końca wiersza
Ctrl+A	Zaznacz wszystko
Shift+	Zmień bieżący wybór wstecz o linię
Ctrl+S	Zmień bieżący wybór do przodu o akapit
Ctrl+S	Zmień bieżące zaznaczenie wstecz o akapit
Esc, MenuKe	Pokaż elementy sterujące przeglądarki e-booków
Ctrl+C	Pokaż preferencje przeglądarki e-booków
Ctrl+E	
Meta+E	
Meta+C	
Ctrl+G	Idź do określonej lokalizacji lub pozycji
, :	
Ctrl+S	Przełącz automatyczne przewijanie
Alt+Ar	Szybsze autoprzewijanie
Alt+Ar	Wolniejsze autoprzewijanie
Ctrl+I	Pokaż/Ukryj Inspektora
Ctrl+L	Pokaż/ukryj panel wyszukiwania słów
Ctrl+Q	Zakończ
(Cmd+Q on macOS)	
Ctrl+P	Drukuj do PDF
Ctrl+F	Przełącz pasek narzędzi
Ctrl+H	Przełącz panel podświetlenia
Ctrl+D	Edytuj tą książkę

3.12 Non re-flowable content

Some books have very wide content that cannot be broken up at page boundaries. For example tables or `<pre>` tags. In such cases, you should switch the viewer to *flow mode* by pressing `Ctrl+M` to read this content. Alternately, you can also add the following CSS to the *Styles* section of the viewer preferences to force the viewer to break up lines of text in `<pre>` tags:

```
code, pre { white-space: pre-wrap }
```

3.13 Designing your book to work well with the calibre viewer

The calibre viewer will set the `is-calibre-viewer` class on the root element. So you can write CSS rules that apply only for it. Additionally, the viewer will set the following classes on the `body` element:

`body.calibre-viewer-dark-colors`

Set when using a dark color scheme

`body.calibre-viewer-light-colors`

Set when using a light color scheme

`body.calibre-viewer-paginated`

Set when in paged mode

`body.calibre-viewer-scrolling`

Set when in flow (non-paginated) mode

`body.calibre-footnote-container`

Set when displaying a popup footnote

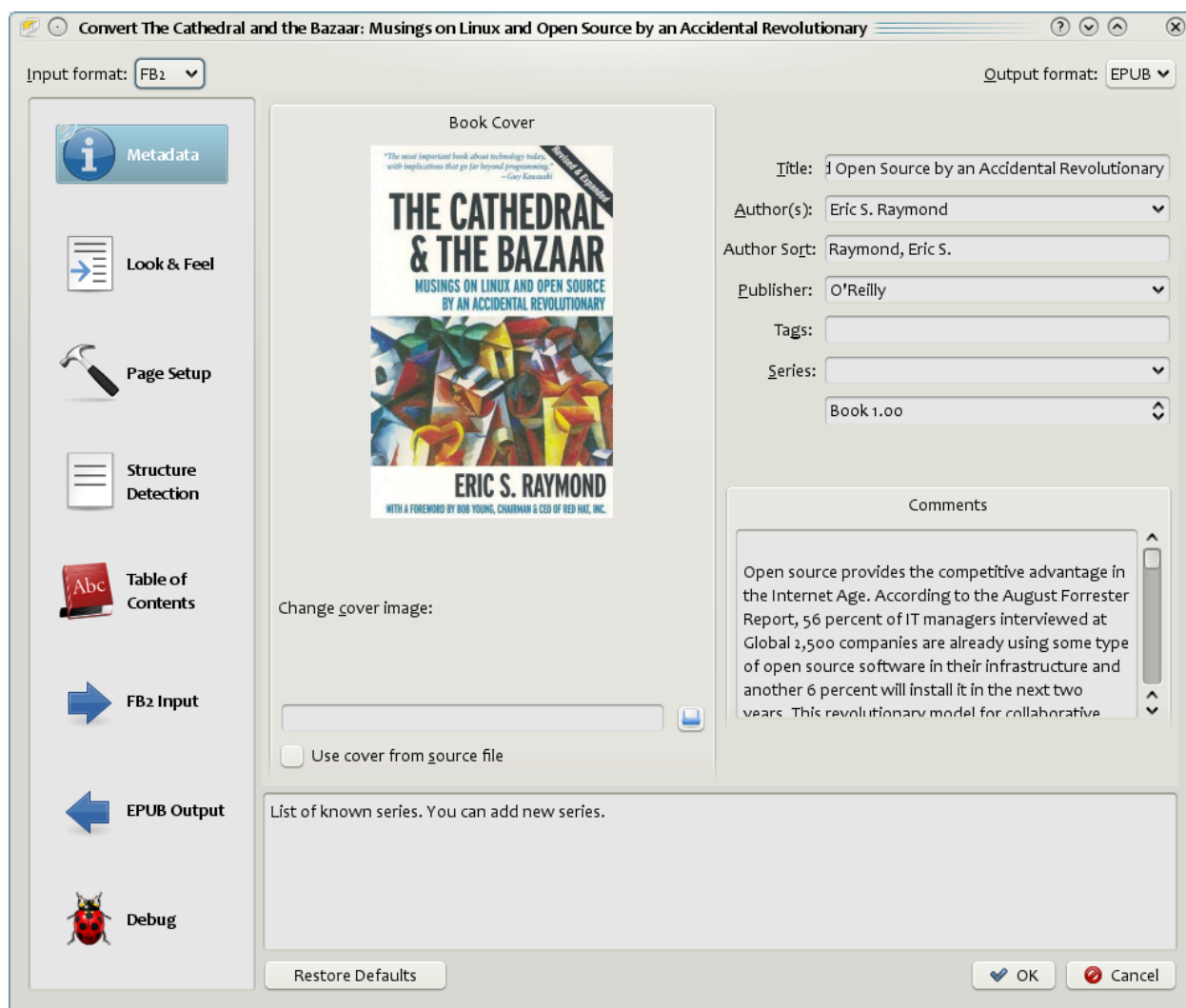
Finally, you can use the calibre color scheme colors via [CSS variables](https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties)²⁷. The calibre viewer defines the following variables: `--calibre-viewer-background-color`, `--calibre-viewer-foreground-color` and optionally `--calibre-viewer-link-color` in color themes that define a link color.

²⁷ https://developer.mozilla.org/en-US/docs/Web/CSS/Using_CSS_custom_properties

Konwersja książek

calibre has a conversion system that is designed to be very easy to use. Normally, you just add a book to calibre, click convert and calibre will try hard to generate output that is as close as possible to the input. However, calibre accepts a very large number of input formats, not all of which are as suitable as others for conversion to e-books. In the case of such input formats, or if you just want greater control over the conversion system, calibre has a lot of options to fine tune the conversion process. Note however that calibre's conversion system is not a substitute for a full blown e-book editor. To edit e-books, I recommend first converting them to EPUB or AZW3 using calibre and then using the *Edit book* feature to get them into perfect shape. You can then use the edited e-book as input for conversion into other formats in calibre.

This document will refer mainly to the conversion settings as found in the conversion dialog, pictured below. All these settings are also available via command line interface to conversion, documented at generated/en/ebook-convert. In calibre, you can obtain help on any individual setting by holding your mouse over it, a tooltip will appear describing the setting.

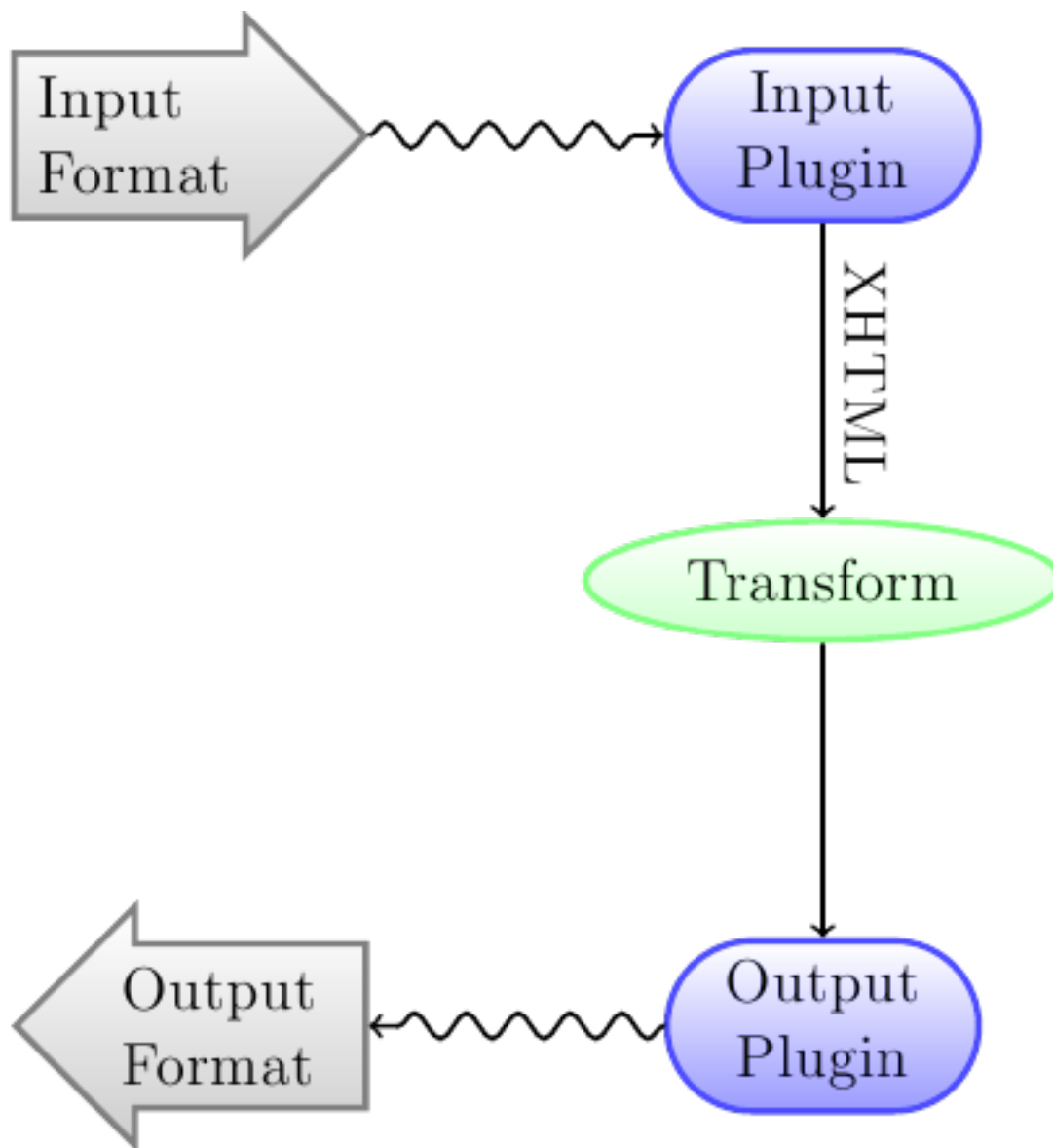


Spis treści

- *Wstęp* (strona 61)
- *Wygląd* (strona 62)
- *Ustawienia strony* (strona 65)
- *Przetwarzanie heurystyczne* (strona 65)
- *Wyszukaj i zastąp* (strona 67)
- *Wykrywanie struktury* (strona 67)
- *Spis treści* (strona 68)
- *Using images as chapter titles when converting HTML input documents* (strona 70)
- *Using tag attributes to supply the text for entries in the Table of Contents* (strona 70)
- *How options are set/saved for conversion* (strona 71)
- *Format specific tips* (strona 71)

4.1 Wstęp

Pierwszym krokiem do zrozumienia systemu konwersji jest to, że jest zaprojektowany jak rurociąg. Na schemacie prezentuje się to następująco:



The input format is first converted to XHTML by the appropriate *Input plugin*. This HTML is then *transformed*. In the last step, the processed XHTML is converted to the specified output format by the appropriate *Output plugin*. The results of the conversion can vary greatly, based on the input format. Some formats convert much better than others. A list of the best source formats for conversion is available [here](#) (strona 130).

The transforms that act on the XHTML output are where all the work happens. There are various transforms, for example, to insert book metadata as a page at the start of the book, to detect chapter headings and automatically create a Table of Contents, to proportionally adjust font sizes, et cetera. It is important to remember that all the transforms act on the XHTML output by the *Input plugin*, not on the input file itself. So, for example, if you ask calibre to convert an RTF file to EPUB, it will first be converted to XHTML internally, the various transforms will be applied to the XHTML and then the *Output plugin* will create the EPUB file, automatically generating all metadata, Table of Contents, et cetera.

You can see this process in action by using the debug option



. Just specify the path to a folder for the

debug output. During conversion, calibre will place the XHTML generated by the various stages of the conversion pipeline in different sub-folders. The four sub-folders are:

Tabela 1: Etapy konwersji

Folder	Opis
źródłowy	To zawiera dane wyjściowe w formacie HTML, tworzone przez wtyczkę wejściową. Użyj ich do usunięcia błędów we wtyczce wejściowej.
przeanalizowane	Wynik obróbki wstępnej i konwersji do formatu XHTML przez wtyczkę wejściową. Użyj do usunięcia błędów wykrywania struktury.
struktura	Post structure detection, but before CSS flattening and font size conversion. Use to debug font size conversion and CSS transforms.
przetworzone	Just before the e-book is passed to the Output plugin. Use to debug the Output plugin.

If you want to edit the input document a little before having calibre convert it, the best thing to do is edit the files in the `input` sub-folder, then zip it up, and use the ZIP file as the input format for subsequent conversions. To do this use the *Edit meta information* dialog to add the ZIP file as a format for the book and then, in the top left corner of the conversion dialog, select ZIP as the input format.

This document will deal mainly with the various transforms that operate on the intermediate XHTML and how to control them. At the end are some tips specific to each input/output format.

4.2 Wygląd

Spis treści

- *Czcionki* (strona 63)
- *Tekst* (strona 64)
- *Układ* (strona 64)
- *Styling* (strona 64)
- *Transformuj style* (strona 65)
- *Transform HTML* (strona 65)

This group of options controls various aspects of the look and feel of the converted e-book.

4.2.1 Czcionki

One of the nicest features of the e-reading experience is the ability to easily adjust font sizes to suit individual needs and lighting conditions. calibre has sophisticated algorithms to ensure that all the books it outputs have a consistent font sizes, no matter what font sizes are specified in the input document.

The base font size of a document is the most common font size in that document, i.e., the size of the bulk of text in that document. When you specify a *Base font size*, calibre automatically rescales all font sizes in the document proportionately, so that the most common font size becomes the specified base font size and other font sizes are rescaled appropriately. By choosing a larger base font size, you can make the fonts in the document larger and vice versa. When you set the base font size, for best results, you should also set the font size key.

Normally, calibre will automatically choose a base font size appropriate to the output profile you have chosen (see [Ustawienia strony](#) (strona 65)). However, you can override this here in case the default is not suitable for you.

The *Font size key* option lets you control how non-base font sizes are rescaled. The font rescaling algorithm works using a font size key, which is simply a comma-separated list of font sizes. The font size key tells calibre how many „steps” bigger or smaller a given font size should be compared to the base font size. The idea is that there should be a limited number of font sizes in a document. For example, one size for the body text, a couple of sizes for different levels of headings and a couple of sizes for super/sub scripts and footnotes. The font size key allows calibre to compartmentalize the font sizes in the input documents into separate „bins” corresponding to the different logical font sizes.

Let’s illustrate with an example. Suppose the source document we are converting was produced by someone with excellent eyesight and has a base font size of 8pt. That means the bulk of the text in the document is sized at 8pts, while headings are somewhat larger (say 10 and 12pt) and footnotes somewhat smaller at 6pt. Now if we use the following settings:

```
Base font size : 12pt
Font size key  : 7, 8, 10, 12, 14, 16, 18, 20
```

The output document will have a base font size of 12pt, headings of 14 and 16pt and footnotes of 8pt. Now suppose we want to make the largest heading size stand out more and make the footnotes a little larger as well. To achieve this, the font key should be changed to:

```
New font size key : 7, 9, 12, 14, 18, 20, 22
```

The largest headings will now become 18pt, while the footnotes will become 9pt. You can play with these settings to try and figure out what would be optimum for you by using the font rescaling wizard, which can be accessed by clicking the little button next to the *Font size key* setting.

All the font size rescaling in the conversion can also be disabled here, if you would like to preserve the font sizes in the input document.

A related setting is *Line height*. Line height controls the vertical height of lines. By default, (a line height of 0), no manipulation of line heights is performed. If you specify a non-default value, line heights will be set in all locations that don’t specify their own line heights. However, this is something of a blunt weapon and should be used sparingly. If you want to adjust the line heights for some section of the input, it’s better to use the [Extra CSS](#) (strona 64).

In this section you can also tell calibre to embed any referenced fonts into the book. This will allow the fonts to work on reader devices even if they are not available on the device.

4.2.2 Tekst

Text can be either justified or not. Justified text has extra spaces between words to give a smooth right margin. Some people prefer justified text, others do not. Normally, calibre will preserve the justification in the original document. If you want to override it you can use the *Text justification* option in this section.

You can also tell calibre to *Smarten punctuation* which will replace plain quotes, dashes and ellipses with their typographically correct alternatives. Note that this algorithm is not perfect so it is worth reviewing the results. The reverse, namely, *Unsmarten punctuation* is also available.

Finally, there is *Input character encoding*. Older documents sometimes don't specify their character encoding. When converted, this can result in non-English characters or special characters like smart quotes being corrupted. calibre tries to auto-detect the character encoding of the source document, but it does not always succeed. You can force it to assume a particular character encoding by using this setting. *cp1252* is a common encoding for documents produced using Windows software. You should also read *Jak przekonwertować pliki, zawierające znaki diakrytyczne lub cudzysłowy typograficzne?* (strona 130) for more on encoding issues.

4.2.3 Układ

Normally, paragraphs in XHTML are rendered with a blank line between them and no leading text indent. calibre has a couple of options to control this. *Remove spacing between paragraphs* forcefully ensure that all paragraphs have no inter paragraph spacing. It also sets the text indent to 1.5em (can be changed) to mark the start of every paragraph. *Insert blank line* does the opposite, guaranteeing that there is exactly one blank line between each pair of paragraphs. Both these options are very comprehensive, removing spacing, or inserting it for *all* paragraphs (technically <p> and <div> tags). This is so that you can just set the option and be sure that it performs as advertised, irrespective of how messy the input file is. The one exception is when the input file uses hard line breaks to implement inter-paragraph spacing.

If you want to remove the spacing between all paragraphs, except a select few, don't use these options. Instead add the following CSS code to *Extra CSS* (strona 64):

```
p, div { margin: 0pt; border: 0pt; text-indent: 1.5em }
.spacious { margin-bottom: 1em; text-indent: 0pt; }
```

Then, in your source document, mark the paragraphs that need spacing with *class="spacious"*. If your input document is not in HTML, use the Debug option, described in the Introduction to get HTML (use the *input* sub-folder).

Another useful options is *Linearize tables*. Some badly designed documents use tables to control the layout of text on the page. When converted these documents often have text that runs off the page and other artifacts. This option will extract the content from the tables and present it in a linear fashion. Note that this option linearizes *all* tables, so only use it if you are sure the input document does not use tables for legitimate purposes, like presenting tabular information.

4.2.4 Styling

The *Extra CSS* option allows you to specify arbitrary CSS that will be applied to all HTML files in the input. This CSS is applied with very high priority and so should override most CSS present in the **input document** itself. You can use this setting to fine tune the presentation/layout of your document. For example, if you want all paragraphs of class *endnote* to be right aligned, just add:

```
.endnote { text-align: right }
```

or if you want to change the indentation of all paragraphs:

```
p { text-indent: 5mm; }
```

Extra CSS is a very powerful option, but you do need an understanding of how CSS works to use it to its full potential. You can use the debug pipeline option described above to see what CSS is present in your input document.

A simpler option is to use *Filter style information*. This allows you to remove all CSS properties of the specified types from the document. For example, you can use it to remove all colors or fonts.

4.2.5 Transformuj style

This is the most powerful styling related facility. You can use it to define rules that change styles based on various conditions. For example you can use it to change all green colors to blue, or remove all bold styling from the text or color all headings a certain color, etc.

4.2.6 Transform HTML

Similar to transform styles, but allows you to make changes to the HTML content of the book. You can replace one tag with another, add classes or other attributes to tags based on their content, etc.

4.3 Ustawienia strony

The *Page setup* options are for controlling screen layout, like margins and screen sizes. There are options to setup page margins, which will be used by the output plugin, if the selected output format supports page margins. In addition, you should choose an Input profile and an output profile. Both sets of profiles basically deal with how to interpret measurements in the input/output documents, screen sizes and default font rescaling keys.

If you know that the file you are converting was intended to be used on a particular device/software platform, choose the corresponding input profile, otherwise just choose the default input profile. If you know the files you are producing are meant for a particular device type, choose the corresponding output profile. Otherwise, choose one of the Generic output profiles. If you are converting to MOBI or AZW3 then you will almost always want to choose one of the Kindle output profiles. Otherwise, your best bet for modern E-book reading devices is to choose the *Generic e-ink HD* output profile.

The output profile also controls the screen size. This will cause, for example, images to be auto-resized to be fit to the screen in some output formats. So choose a profile of a device that has a screen size similar to your device.

4.4 Przetwarzanie heurystyczne

Heuristic processing provides a variety of functions which can be used to try and detect and correct common problems in poorly formatted input documents. Use these functions if your input document suffers from poor formatting. Because these functions rely on common patterns, be aware that in some cases an option may lead to worse results, so use with care. As an example, several of these options will remove all non-breaking-space entities, or may include false positive matches relating to the function.

Enable heuristic processing

This option activates calibre's *Heuristic processing* stage of the conversion pipeline. This must be enabled in order for various sub-functions to be applied

Unwrap lines

Enabling this option will cause calibre to attempt to detect and correct hard line breaks that exist within a document using punctuation clues and line length. calibre will first attempt to detect whether hard line breaks exist, if they do not appear to exist calibre will not attempt to unwrap lines. The line-unwrap factor can be reduced if you want to «force» calibre to unwrap lines.

Line-unwrap factor

This option controls the algorithm calibre uses to remove hard line breaks. For example, if the value of this option is 0.4, that means calibre will remove hard line breaks from the end of lines whose lengths are less than the length of 40% of all lines in the document. If your document only has a few line breaks which need correction, then this value should be reduced to somewhere between 0.1 and 0.2.

Detect and markup unformatted chapter headings and sub headings

If your document does not have chapter headings and titles formatted differently from the rest of the text, calibre can use this option to attempt to detect them and surround them with heading tags. <h2> tags are used for chapter headings; <h3> tags are used for any titles that are detected.

This function will not create a TOC, but in many cases it will cause calibre's default chapter detection settings to correctly detect chapters and build a TOC. Adjust the XPath under Structure detection if a TOC is not automatically created. If there are no other headings used in the document then setting „//h:h2” under Structure detection would be the easiest way to create a TOC for the document.

The inserted headings are not formatted, to apply formatting use the *Extra CSS* option under the Look and Feel conversion settings. For example, to center heading tags, use the following:

```
h2, h3 { text-align: center }
```

Renumber sequences of <h1> or <h2> tags

Some publishers format chapter headings using multiple <h1> or <h2> tags sequentially. calibre's default conversion settings will cause such titles to be split into two pieces. This option will re-number the heading tags to prevent splitting.

Delete blank lines between paragraphs

This option will cause calibre to analyze blank lines included within the document. If every paragraph is interleaved with a blank line, then calibre will remove all those blank paragraphs. Sequences of multiple blank lines will be considered scene breaks and retained as a single paragraph. This option differs from the *Remove paragraph spacing* option under *Look and Feel* in that it actually modifies the HTML content, while the other option modifies the document styles. This option can also remove paragraphs which were inserted using calibre's *Insert blank line* option.

Ensure scene breaks are consistently formatted

With this option calibre will attempt to detect common scene-break markers and ensure that they are center aligned. «Soft» scene break markers, i.e. scene breaks only defined by extra white space, are styled to ensure that they will not be displayed in conjunction with page breaks.

Replace scene breaks

If this option is configured then calibre will replace scene break markers it finds with the replacement text specified by the user. Please note that some ornamental characters may not be supported across all reading devices.

In general you should avoid using HTML tags, calibre will discard any tags and use pre-defined markup. <hr /> tags, i.e. horizontal rules, and tags are exceptions. Horizontal rules can optionally be specified with styles, if you choose to add your own style be sure to include the «width» setting, otherwise the style information will be discarded. Image tags can be used, but calibre does not provide the ability to add the image during conversion, this must be done after the fact using the «Edit book» feature.

Example image tag (place the image within an «Images» folder inside the EPUB after conversion):

```

```

Example horizontal rule with styles:

```
<hr style="width:20%;padding-top: 1px;border-top: 2px ridge black;border-bottom: 2px groove black;" />
```

Remove unnecessary hyphens

calibre will analyze all hyphenated content in the document when this option is enabled. The document itself is

used as a dictionary for analysis. This allows calibre to accurately remove hyphens for any words in the document in any language, along with made-up and obscure scientific words. The primary drawback is words appearing only a single time in the document will not be changed. Analysis happens in two passes, the first pass analyzes line endings. Lines are only unwrapped if the word exists with or without a hyphen in the document. The second pass analyzes all hyphenated words throughout the document, hyphens are removed if the word exists elsewhere in the document without a match.

Italicize common words and patterns

When enabled, calibre will look for common words and patterns that denote italics and italicize them. Examples are common text conventions such as ~word~ or phrases that should generally be italicized, e.g. latin phrases like «etc.» or «et cetera».

Replace entity indents with CSS indents

Some documents use a convention of defining text indents using non-breaking space entities. When this option is enabled calibre will attempt to detect this sort of formatting and convert them to a 3% text indent using CSS.

4.5 Wyszukaj i zastąp

These options are useful primarily for conversion of PDF documents or OCR conversions, though they can also be used to fix many document specific problems. As an example, some conversions can leave behind page headers and footers in the text. These options use regular expressions to try and detect headers, footers, or other arbitrary text and remove or replace them. Remember that they operate on the intermediate XHTML produced by the conversion pipeline. There is a wizard to help you customize the regular expressions for your document. Click the magic wand beside the expression box, and click the «Test» button after composing your search expression. Successful matches will be highlighted in Yellow.

The search works by using a Python regular expression. All matched text is simply removed from the document or replaced using the replacement pattern. The replacement pattern is optional, if left blank then text matching the search pattern will be deleted from the document. You can learn more about regular expressions and their syntax at [Wszystko, co chciałbyście wiedzieć o wyrażeniach regularnych w calibre](#) (strona 213).

4.6 Wykrywanie struktury

Structure detection involves calibre trying its best to detect structural elements in the input document, when they are not properly specified. For example, chapters, page breaks, headers, footers, etc. As you can imagine, this process varies widely from book to book. Fortunately, calibre has very powerful options to control this. With power comes complexity, but if once you take the time to learn the complexity, you will find it well worth the effort.

4.6.1 Chapters and page breaks

calibre has two sets of options for *chapter detection* and *inserting page breaks*. This can sometimes be slightly confusing, as by default, calibre will insert page breaks before detected chapters as well as the locations detected by the page breaks option. The reason for this is that there are often locations where page breaks should be inserted that are not chapter boundaries. Also, detected chapters can be optionally inserted into the auto generated Table of Contents.

calibre uses *XPath*, a powerful language to allow the user to specify chapter boundaries/page breaks. XPath can seem a little daunting to use at first, fortunately, there is a [XPath tutorial](#) (strona 159) in the User Manual. Remember that Structure detection operates on the intermediate XHTML produced by the conversion pipeline. Use the debug option described in the [Wstęp](#) (strona 61) to figure out the appropriate settings for your book. There is also a button for a XPath wizard to help with the generation of simple XPath expressions.

By default, calibre uses the following expression for detecting chapters:

```
/* [ (name()='h1' or name()='h2') and re:test(., 'chapter|book|section|part\s+', 'i
↪') ) or @class = 'chapter' ]
```

This expression is rather complex, because it tries to handle a number of common cases simultaneously. What it means is that calibre will assume chapters start at either `<h1>` or `<h2>` tags that have any of the words (*chapter*, *book*, *section* or *part*) in them or that have the `class="chapter"` attribute.

A related option is *Chapter mark*, which allows you to control what calibre does when it detects a chapter. By default, it will insert a page break before the chapter. You can have it insert a ruled line instead of, or in addition to the page break. You can also have it do nothing.

The default setting for detecting page breaks is:

```
/* [name()='h1' or name()='h2']
```

which means that calibre will insert page breaks before every `<h1>` and `<h2>` tag by default.

Informacja: The default expressions may change depending on the input format you are converting.

4.6.2 Różne

W tej sekcji jest jeszcze trochę innych opcji.

Insert metadata as page at start of book

One of the great things about calibre is that it allows you to maintain very complete metadata about all of your books, for example, a rating, tags, comments, etc. This option will create a single page with all this metadata and insert it into the converted e-book, typically just after the cover. Think of it as a way to create your own customised book jacket.

Remove first image

Sometimes, the source document you are converting includes the cover as part of the book, instead of as a separate cover. If you also specify a cover in calibre, then the converted book will have two covers. This option will simply remove the first image from the source document, thereby ensuring that the converted book has only one cover, the one specified in calibre.

4.7 Spis treści

When the input document has a Table of Contents in its metadata, calibre will just use that. However, a number of older formats either do not support a metadata based Table of Contents, or individual documents do not have one. In these cases, the options in this section can help you automatically generate a Table of Contents in the converted e-book, based on the actual content in the input document.

Informacja: Using these options can be a little challenging to get exactly right. If you prefer creating/editing the Table of Contents by hand, convert to the EPUB or AZW3 formats and select the checkbox at the bottom of the Table of Contents section of the conversion dialog that says *Manually fine-tune the Table of Contents after conversion*. This will launch the ToC Editor tool after the conversion. It allows you to create entries in the Table of Contents by simply clicking the place in the book where you want the entry to point. You can also use the ToC Editor by itself, without doing a conversion. Go to *Preferences* → *Interface* → *Toolbars* and add the *ToC Editor* to the main toolbar. Then just select the book you want to edit and click the *ToC Editor* button.

The first option is *Force use of auto-generated Table of Contents*. By checking this option you can have calibre override any Table of Contents found in the metadata of the input document with the auto generated one.

The default way that the creation of the auto generated Table of Contents works is that, calibre will first try to add any detected chapters to the generated table of contents. You can learn how to customize the detection of chapters in the [Wykrywanie struktury](#) (strona 67) section above. If you do not want to include detected chapters in the generated table of contents, check the *Do not add detected chapters* option.

If less than the *Chapter threshold* number of chapters were detected, calibre will then add any hyperlinks it finds in the input document to the Table of Contents. This often works well: many input documents include a hyperlinked Table of Contents right at the start. The *Number of links* option can be used to control this behavior. If set to zero, no links are added. If set to a number greater than zero, at most that number of links is added.

calibre will automatically filter duplicates from the generated Table of Contents. However, if there are some additional undesirable entries, you can filter them using the *TOC Filter* option. This is a regular expression that will match the title of entries in the generated table of contents. Whenever a match is found, it will be removed. For example, to remove all entries titles „Next” or „Previous” use:

```
Next|Previous
```

The *Level 1,2,3 TOC* options allow you to create a sophisticated multi-level Table of Contents. They are XPath expressions that match tags in the intermediate XHTML produced by the conversion pipeline. See the [Wstęp](#) (strona 61) for how to get access to this XHTML. Also read the [Samouczek XPath](#) (strona 159), to learn how to construct XPath expressions. Next to each option is a button that launches a wizard to help with the creation of basic XPath expressions. The following simple example illustrates how to use these options.

Suppose you have an input document that results in XHTML that look like this:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Sample document</title>
  </head>
  <body>
    <h1>Chapter 1</h1>
    ...
    <h2>Section 1.1</h2>
    ...
    <h2>Section 1.2</h2>
    ...
    <h1>Chapter 2</h1>
    ...
    <h2>Section 2.1</h2>
    ...
  </body>
</html>
```

Then, we set the options as:

```
Level 1 TOC : //h:h1
Level 2 TOC : //h:h2
```

This will result in an automatically generated two level Table of Contents that looks like:

```
Chapter 1
  Section 1.1
  Section 1.2
Chapter 2
  Section 2.1
```

Ostrzeżenie: Not all output formats support a multi level Table of Contents. You should first try with EPUB output. If that works, then try your format of choice.

4.8 Using images as chapter titles when converting HTML input documents

Suppose you want to use an image as your chapter title, but still want calibre to be able to automatically generate a Table of Contents for you from the chapter titles. Use the following HTML markup to achieve this:

```
<html>
  <body>
    <h2>Chapter 1</h2>
    <p>chapter 1 text...</p>
    <h2 title="Chapter 2"></h2>
    <p>chapter 2 text...</p>
  </body>
</html>
```

Set the *Level 1 TOC* setting to `//h:h2`. Then, for chapter two, calibre will take the title from the value of the `title` attribute on the `<h2>` tag, since the tag has no text.

4.9 Using tag attributes to supply the text for entries in the Table of Contents

If you have particularly long chapter titles and want shortened versions in the Table of Contents, you can use the `title` attribute to achieve this, for example:

```
<html>
  <body>
    <h2 title="Chapter 1">Chapter 1: Some very long title</h2>
    <p>chapter 1 text...</p>
    <h2 title="Chapter 2">Chapter 2: Some other very long title</h2>
    <p>chapter 2 text...</p>
  </body>
</html>
```

Set the *Level 1 TOC* setting to `//h:h2/@title`. Then calibre will take the title from the value of the `title` attribute on the `<h2>` tags, instead of using the text inside the tag. Note the trailing `/@title` on the XPath expression, you can use this form to tell calibre to get the text from any attribute you like.

4.10 How options are set/saved for conversion

There are two places where conversion options can be set in calibre. The first is in Preferences->Conversion. These settings are the defaults for the conversion options. Whenever you try to convert a new book, the settings set here will be used by default.

You can also change settings in the conversion dialog for each book conversion. When you convert a book, calibre remembers the settings you used for that book, so that if you convert it again, the saved settings for the individual book will take precedence over the defaults set in *Preferences*. You can restore the individual settings to defaults by using the *Restore defaults* button in the individual book conversion dialog. You can remove the saved settings for a group of books by selecting all the books and then clicking the *Edit metadata* button to bring up the bulk metadata edit dialog, near the bottom of the dialog is an option to remove stored conversion settings.

When you bulk convert a set of books, settings are taken in the following order (last one wins):

- From the defaults set in Preferences->Conversion
- From the saved conversion settings for each book being converted (if any). This can be turned off by the option in the top left corner of the Bulk conversion dialog.
- From the settings set in the Bulk conversion dialog

Note that the final settings for each book in a Bulk conversion will be saved and re-used if the book is converted again. Since the highest priority in Bulk Conversion is given to the settings in the Bulk conversion dialog, these will override any book specific settings. So you should only bulk convert books together that need similar settings. The exceptions are metadata and input format specific settings. Since the Bulk conversion dialog does not have settings for these two categories, they will be taken from book specific settings (if any) or the defaults.

Informacja: You can see the actual settings used during any conversion by clicking the rotating icon in the lower right corner and then double clicking the individual conversion job. This will bring up a conversion log that will contain the actual settings used, near the top.

4.11 Format specific tips

Here you will find tips specific to the conversion of particular formats. Options specific to particular format, whether input or output are available in the conversion dialog under their own section, for example *TXT input* or *EPUB output*.

4.11.1 Convert Microsoft Word documents

calibre can automatically convert `.docx` files created by Microsoft Word 2007 and newer. Just add the file to calibre and click convert.

Informacja: There is a [demo .docx file](https://calibre-ebook.com/downloads/demos/demo.docx)²⁸ that demonstrates the capabilities of the calibre conversion engine. Just download it and convert it to EPUB or AZW3 to see what calibre can do.

calibre will automatically generate a Table of Contents based on headings if you mark your headings with the *Heading 1*, *Heading 2*, etc. styles in Microsoft Word. Open the output e-book in the calibre E-book viewer and click the *Table of Contents* button to view the generated Table of Contents.

²⁸ <https://calibre-ebook.com/downloads/demos/demo.docx>

Older .doc files

For older .doc files, you can save the document as HTML with Microsoft Word and then convert the resulting HTML file with calibre. When saving as HTML, be sure to use the „Save as Web Page, Filtered” option as this will produce clean HTML that will convert well. Note that Word produces really messy HTML, converting it can take a long time, so be patient. If you have a newer version of Word available, you can directly save it as .docx as well.

Another alternative is to use the free LibreOffice. Open your .doc file in LibreOffice and save it as .docx, which can be directly converted in calibre.

4.11.2 Convert TXT documents

TXT documents have no well defined way to specify formatting like bold, italics, etc, or document structure like paragraphs, headings, sections and so on, but there are a variety of conventions commonly used. By default calibre attempts automatic detection of the correct formatting and markup based on those conventions.

TXT input supports a number of options to differentiate how paragraphs are detected.

Paragraph style: Auto

Analyzes the text file and attempts to automatically determine how paragraphs are defined. This option will generally work fine, if you achieve undesirable results try one of the manual options.

Paragraph style: Block

Assumes one or more blank lines are a paragraph boundary:

```
This is the first.  
  
This is the  
second paragraph.
```

Paragraph style: Single

Assumes that every line is a paragraph:

```
This is the first.  
This is the second.  
This is the third.
```

Paragraph style: Print

Assumes that every paragraph starts with an indent (either a tab or 2+ spaces). Paragraphs end when the next line that starts with an indent is reached:

```
    This is the  
first.  
    This is the second.  
  
    This is the  
third.
```

Paragraph style: Unformatted

Assumes that the document has no formatting, but does use hard line breaks. Punctuation and median line length are used to attempt to re-create paragraphs.

Formatting style: Auto

Attempts to detect the type of formatting markup being used. If no markup is used then heuristic formatting will be applied.

Formatting style: Heuristic

Analyzes the document for common chapter headings, scene breaks, and italicized words and applies the appropriate HTML markup during conversion.

Formatting style: Markdown

calibre also supports running TXT input through a transformation preprocessor known as Markdown. Markdown allows for basic formatting to be added to TXT documents, such as bold, italics, section headings, tables, lists, a Table of Contents, etc. Marking chapter headings with a leading # and setting the chapter XPath detection expression to „//h:h1” is the easiest way to have a proper table of contents generated from a TXT document. You can learn more about the Markdown syntax at [daringfireball](https://daringfireball.net/projects/markdown/syntax)²⁹.

Formatting style: None

Applies no special formatting to the text, the document is converted to HTML with no other changes.

4.11.3 Convert PDF documents

PDF documents are one of the worst formats to convert from. They are a fixed page size and text placement format. Meaning, it is very difficult to determine where one paragraph ends and another begins. calibre will try to unwrap paragraphs using a configurable, *Line un-wrapping factor*. This is a scale used to determine the length at which a line should be unwrapped. Valid values are a decimal between 0 and 1. The default is 0.45, just under the median line length. Lower this value to include more text in the unwrapping. Increase to include less. You can adjust this value in the conversion settings under *PDF Input*.

Also, they often have headers and footers as part of the document that will become included with the text. Use the *Search and replace* panel to remove headers and footers to mitigate this issue. If the headers and footers are not removed from the text it can throw off the paragraph unwrapping. To learn how to use the header and footer removal options, read *Wszystko, co chcielibyście wiedzieć o wyrażeniach regularnych w calibre* (strona 213).

Some limitations of PDF input are:

- Complex, multi-column, and image based documents are not supported.
- Extraction of vector images and tables from within the document is also not supported.
- Some PDFs use special glyphs to represent ll or ff or fi, etc. Conversion of these may or may not work depending on just how they are represented internally in the PDF.
- Links and Tables of Contents are not supported
- PDFs that use embedded non-Unicode fonts to represent non-English characters will result in garbled output for those characters
- Some PDFs are made up of photographs of the page with OCR'd text behind them. In such cases calibre uses the OCR'd text, which can be very different from what you see when you view the PDF file
- PDFs that are used to display complex text, like right to left languages and math typesetting will not convert correctly

To re-iterate **PDF is a really, really bad** format to use as input. If you absolutely must use PDF, then be prepared for an output ranging anywhere from decent to unusable, depending on the input PDF.

²⁹ <https://daringfireball.net/projects/markdown/syntax>

4.11.4 Comic book collections

A comic book collection is a .cbc file. A .cbc file is a ZIP file that contains other CBZ/CBR files. In addition the .cbc file must contain a simple text file called comics.txt, encoded in UTF-8. The comics.txt file must contain a list of the comics files inside the .cbc file, in the form filename:title, as shown below:

```
one.cbz:Chapter One
two.cbz:Chapter Two
three.cbz:Chapter Three
```

The .cbc file will then contain:

```
comics.txt
one.cbz
two.cbz
three.cbz
```

calibre will automatically convert this .cbc file into a e-book with a Table of Contents pointing to each entry in comics.txt.

4.11.5 EPUB advanced formatting demo

Various advanced formatting for EPUB files is demonstrated in this [demo file](#)³⁰. The file was created from hand coded HTML using calibre and is meant to be used as a template for your own EPUB creation efforts.

The source HTML it was created from is available [demo.zip](#)³¹. The settings used to create the EPUB from the ZIP file are:

```
ebook-convert demo.zip .epub -vv --authors "Kovid Goyal" --language en --level1-toc '/*[@class="title"]' --disable-font-rescaling --page-breaks-before / --no-default-epub-cover
```

Note that because this file explores the potential of EPUB, most of the advanced formatting is not going to work on readers less capable than calibre's built-in EPUB viewer.

4.11.6 Convert ODT documents

calibre can directly convert ODT (OpenDocument Text) files. You should use styles to format your document and minimize the use of direct formatting. When inserting images into your document you need to anchor them to the paragraph, images anchored to a page will all end up in the front of the conversion.

To enable automatic detection of chapters, you need to mark them with the built-in styles called *Heading 1*, *Heading 2*, ..., *Heading 6* (*Heading 1* equates to the HTML tag <h1>, *Heading 2* to <h2>, etc). When you convert in calibre you can enter which style you used into the *Detect chapters at* box. Example:

- If you mark Chapters with style *Heading 2*, you have to set the «Detect chapters at» box to //h:h2
- For a nested TOC with Sections marked with *Heading 2* and the Chapters marked with *Heading 3* you need to enter //h:h2|//h:h3. On the Convert - TOC page set the *Level 1 TOC* box to //h:h2 and the *Level 2 TOC* box to //h:h3.

Well-known document properties (Title, Keywords, Description, Creator) are recognized and calibre will use the first image (not too small, and with good aspect-ratio) as the cover image.

³⁰ <https://calibre-ebook.com/downloads/demos/demo.epub>

³¹ <https://calibre-ebook.com/downloads/demos/demo.zip>

There is also an advanced property conversion mode, which is activated by setting the custom property `opf.metadata` («Yes or No» type) to Yes in your ODT document (File->Properties->Custom Properties). If this property is detected by calibre, the following custom properties are recognized (`opf.authors` overrides document creator):

```
opf.titlesort
opf.authors
opf.authorsort
opf.publisher
opf.pubdate
opf.isbn
opf.language
opf.series
opf.seriesindex
```

In addition to this, you can specify the picture to use as the cover by naming it `opf.cover` (right click, Picture->Options->Name) in the ODT. If no picture with this name is found, the «smart» method is used. As the cover detection might result in double covers in certain output formats, the process will remove the paragraph (only if the only content is the cover!) from the document. But this works only with the named picture!

To disable cover detection you can set the custom property `opf.nocover` («Yes or No» type) to Yes in advanced mode.

4.11.7 Converting to PDF

The first, most important, setting to decide on when converting to PDF is the page size. By default, calibre uses a page size of „U.S. Letter”. You can change this to another standard page size or a completely custom size in the *PDF Output* section of the conversion dialog. If you are generating a PDF to be used on a specific device, you can turn on the option to use the page size from the *output profile* instead. So if your output profile is set to Kindle, calibre will create a PDF with page size suitable for viewing on the small Kindle screen.

Nagłówki i stopki

You can insert arbitrary headers and footers on each page of the PDF by specifying header and footer templates. Templates are just snippets of HTML code that get rendered in the header and footer locations. For example, to display page numbers centered at the bottom of every page, in green, use the following footer template:

```
<footer><div style="margin: auto; color: green">_PAGENUM_</div></footer>
```

calibre will automatically replace `_PAGENUM_` with the current page number. You can even put different content on even and odd pages, for example the following header template will show the title on odd pages and the author on even pages:

```
<header style="justify-content: flex-end">
  <div class="even-page">_AUTHOR_</div>
  <div class="odd-page"><i>_TITLE_</i></div>
</header>
```

calibre will automatically replace `_TITLE_` and `_AUTHOR_` with the title and author of the document being converted. Setting `justify-content` to `flex-end` will cause the text to be right aligned.

You can also display text at the left and right edges and change the font size, as demonstrated with this header template:

```
<header style="justify-content: space-between; font-size: smaller">
  <div>_TITLE_</div>
  <div>_AUTHOR_</div>
</header>
```

This will display the title at the left and the author at the right, in a font size smaller than the main text.

You can also use the current section in templates, as shown below:

```
<header><div>_SECTION_</div></header>
```

SECTION is replaced by whatever the name of the current section is. These names are taken from the metadata Table of Contents in the document (the PDF Outline). If the document has no table of contents then it will be replaced by empty text. If a single PDF page has multiple sections, the first section on the page will be used. Similarly, there is a variable named _TOP_LEVEL_SECTION_ that can be used to get the name of the current top-level section.

You can even use JavaScript inside the header and footer templates, for example, the following template will cause page numbers to start at 4 instead of 1:

```
<footer>
  <div></div>
  <script>document.currentScript.parentNode.querySelector("div").innerHTML = "" + (_
  ↪ PAGENUM_ + 3)</script>
</footer>
```

In addition there are some more variables you can use in the headers and footers, documented below:

- _TOTAL_PAGES_ - total number of pages in the PDF file, useful for implementing a progress counter, for example.
- _TOP_LEVEL_SECTION_PAGES_ - total number of pages in the current top level section
- _TOP_LEVEL_SECTION_PAGENUM_ - the page number of the current page within the current top level section

Informacja: When adding headers and footers make sure you set the page top and bottom margins to large enough values, under the *PDF Output* section of the conversion dialog.

Printable Table of Contents

You can also insert a printable Table of Contents at the end of the PDF that lists the page numbers for every section. This is very useful if you intend to print out the PDF to paper. If you wish to use the PDF on an electronic device, then the PDF Outline provides this functionality and is generated by default.

You can customize the look of the generated Table of contents by using the Extra CSS conversion setting under the Look & feel part of the conversion dialog. The default CSS used is listed below, simply copy it and make whatever changes you like.

```
.calibre-pdf-toc table { width: 100%% }

.calibre-pdf-toc table tr td:last-of-type { text-align: right }

.calibre-pdf-toc .level-0 {
  font-size: larger;
}

.calibre-pdf-toc .level-1 td:first-of-type { padding-left: 1.4em }
.calibre-pdf-toc .level-2 td:first-of-type { padding-left: 2.8em }
```


Custom page margins for individual HTML files

If you are converting an EPUB or AZW3 file with multiple individual HTML files inside it and you want to change the page margins for a particular HTML file you can add the following style block to the HTML file using the calibre E-book editor:

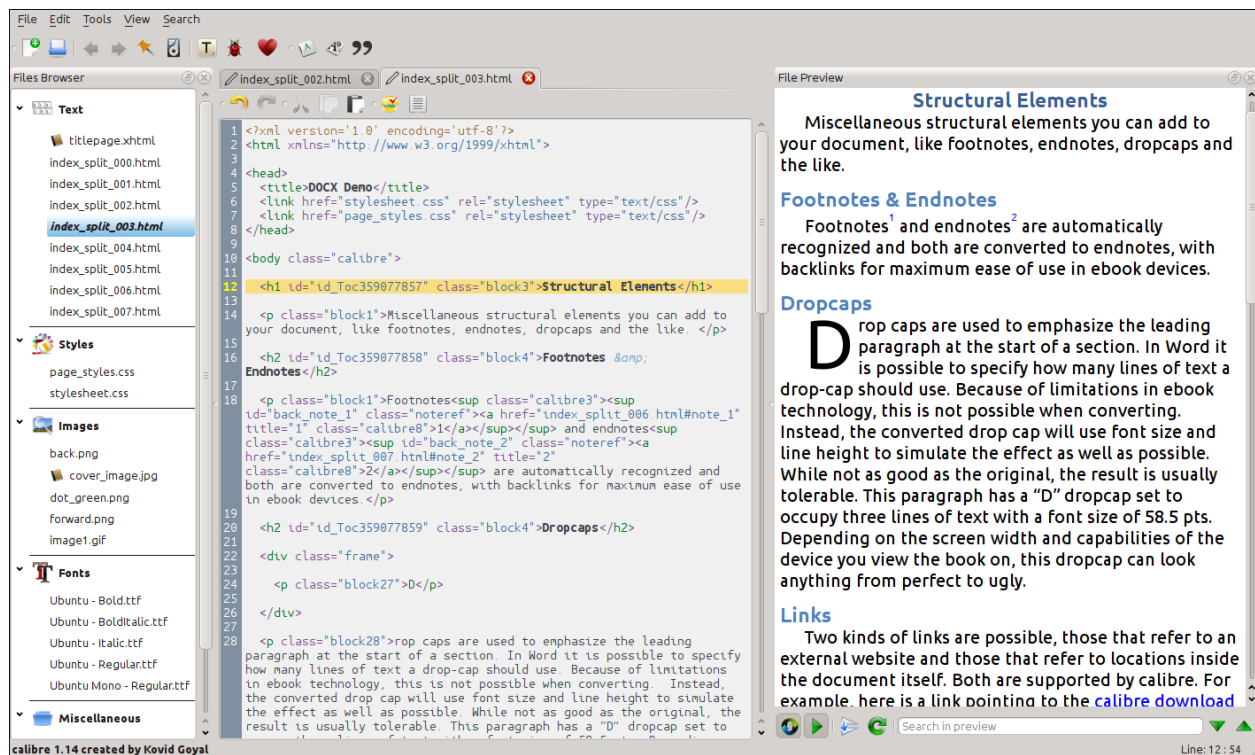
```
<style>
@page {
    margin-left: 10pt;
    margin-right: 10pt;
    margin-top: 10pt;
    margin-bottom: 10pt;
}
</style>
```

Then, in the PDF output section of the conversion dialog, turn on the option to *Use page margins from the document being converted*. Now all pages generated from this HTML file will have 10pt margins.

Edycja e-booków

calibre has an integrated e-book editor that can be used to edit books in the EPUB and AZW3 (Kindle) formats. The editor shows you the HTML and CSS that is used internally inside the book files, with a live preview that updates as you make changes. It also contains various automated tools to perform common cleanup and fixing tasks.

You can use this editor by right clicking on any book in calibre and selecting *Edit book*.



Spis treści

- *Basic workflow* (strona 81)
- *The File browser* (strona 83)
 - *Zmiana nazwy plików* (strona 84)
 - *Merging files* (strona 84)
 - *Changing text file order* (strona 85)
 - *Marking the cover* (strona 85)
 - *Deleting files* (strona 85)
 - *Exporting files* (strona 85)
 - *Adding new images/fonts/etc. or creating new blank files* (strona 85)
 - *Replacing files* (strona 86)
 - *Linking stylesheets to HTML files efficiently* (strona 86)
- *Wyszukaj i zastąp* (strona 86)
 - *Zapisane wyszukiwania* (strona 86)
 - *Function mode* (strona 87)
 - *Search ignoring HTML tags* (strona 87)
- *Automated tools* (strona 87)
 - *Editing the Table of Contents* (strona 87)
 - *Checking the book* (strona 88)
 - *Adding a cover* (strona 89)
 - *Osadzanie użytych czcionek* (strona 89)
 - *Redukowanie osadzonych czcionek* (strona 89)
 - *Inteligentna interpunkcja* (strona 90)
 - *Transforming CSS properties* (strona 90)
 - *Usuwanie nieużywanych stylów* (strona 90)
 - *Fixing HTML* (strona 90)
 - *Beautifying files* (strona 90)
 - *Inserting an inline Table of Contents* (strona 91)
 - *Setting Semantics* (strona 91)
 - *Filtering style information* (strona 91)
 - *Upgrading the book's internals* (strona 91)
- *Punkty kontrolne* (strona 91)
- *The Live preview panel* (strona 93)
 - *Splitting HTML files* (strona 94)
- *The Live CSS panel* (strona 95)

- *Miscellaneous tools* (strona 96)
 - *The Table of Contents view* (strona 96)
 - *Checking the spelling of words in the book* (strona 96)
 - *Inserting special characters* (strona 98)
 - *The code inspector view* (strona 99)
 - *Checking external links* (strona 99)
 - *Downloading external resources* (strona 99)
 - *Arranging files into folders by type* (strona 99)
 - *Importing files in other e-book formats as EPUB* (strona 99)
 - *The Reports tool* (strona 110)
- *Special features in the code editor* (strona 110)
 - *Syntax highlighting* (strona 110)
 - *Context sensitive help* (strona 111)
 - *Auto-complete* (strona 111)
 - *Wstawki* (strona 111)

5.1 Basic workflow

Informacja: A video tour of the calibre E-book editor is available [here](https://calibre-ebook.com/pl/demo#tutorials)³².

When you first open a book with the Edit book tool, you will be presented with a list of files on the left. These are the individual HTML files, stylesheets, images, etc. that make up the content of the book. Simply double click on a file to start editing it. Note that if you want to do anything more sophisticated than making a few small tweaks, you will need to know [HTML Tutorial](http://html.net/tutorials/html/)³³ and [CSS Tutorial](http://html.net/tutorials/css/)³⁴.

As you make changes to the HTML or CSS in the editor, the changes will be previewed, live, in the preview panel to the right. When you are happy with how the changes you have made look, click the *Save* button or use *File → Save* to save your changes into the e-book.

One useful feature is *Checkpoints*. Before you embark on some ambitious set of edits, you can create a checkpoint. The checkpoint will preserve the current state of your book, then if in the future you decide you don't like the changes you have made to you can go back to the state when you created the checkpoint. To create a checkpoint, use *Edit → Create checkpoint*. Checkpoints will also be automatically created for you whenever you run any automated tool like global search and replace. The checkpointing functionality is in addition to the normal undo/redo mechanism when editing individual files. Checkpoints are needed for when changes are spread over multiple files in the book.

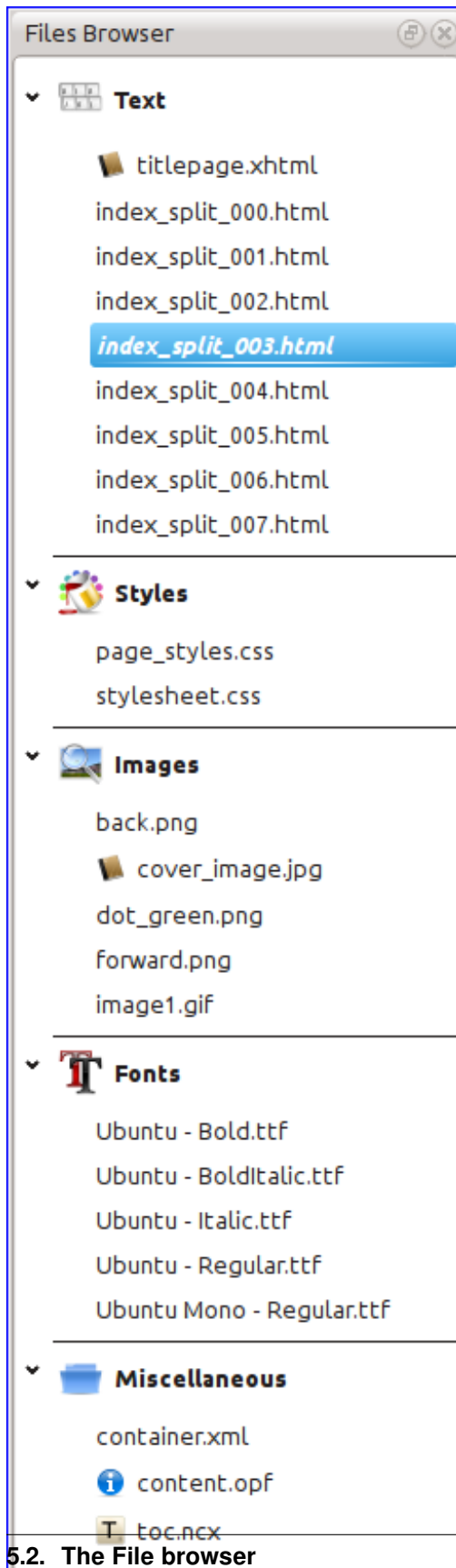
That is the basic work flow for editing books – Open a file, make changes, preview and save. The rest of this manual will discuss the various tools and features present to allow you to perform specific tasks efficiently.

³² <https://calibre-ebook.com/pl/demo#tutorials>

³³ <http://html.net/tutorials/html/>

³⁴ <http://html.net/tutorials/css/>

5.2 The File browser



The *File browser* gives you an overview of the various files inside the book you are editing. The files are arranged by category, with text (HTML) files at the top, followed by stylesheet (CSS) files, images and so on. Simply double click on a file to start editing it. Editing is supported for HTML, CSS and image files. The order of text files is the same order that they would be displayed in, if you were reading the book. All other files are arranged alphabetically.

By hovering your mouse over an entry, you can see its size, and also, at the bottom of the screen, the full path to the file inside the book. Note that files inside e-books are compressed, so the size of the final book is not the sum of the individual file sizes.

Many files have special meaning, in the book. These will typically have an icon next to their names, indicating the special meaning. For example, in the picture to the left, you can see that the files *cover_image.jpg* and *titlepage.xhtml* have the icon of a cover next to them, this indicates they are the book cover image and titlepage. Similarly, the *content.opf* file has a metadata icon next to it, indicating the book metadata is present in it and the *toc.ncx* file has a T icon next to it, indicating it is the Table of Contents.

You can perform many actions on individual files, by right clicking them.

5.2.1 Zmiana nazwy plików

You can rename an individual file by right clicking it and selecting *Rename*. Renaming a file automatically updates all links and references to it throughout the book. So all you have to do is provide the new name, calibre will take care of the rest.

You can also bulk rename many files at once. This is useful if you want the files to have some simple name pattern. For example you might want to rename all the HTML files to have names *Chapter-1.html*, *Chapter-2.html* and so on. Select the files you want bulk renamed by holding down the *Shift* or *Ctrl* key and clicking the files. Then right click and select *Bulk rename*. Enter a prefix and what number you would like the automatic numbering to start at, click *OK* and you are done. The bulk rename dialog also lets you rename files by the order they appear in the book instead of the order you selected them in, useful, for instance to rename all images by the order they appear.

Finally, you can bulk change the file extension for all selected files. Select multiple files, as above, and right click and choose *Change the file extension for the selected files*.

5.2.2 Merging files

Sometimes, you may want to merge two HTML files or two CSS files together. It can sometimes be useful to have everything in a single file. Be wary, though, putting a lot of content into a single file will cause performance problems when viewing the book in a typical e-book reader.

To merge multiple files together, select them by holding the *Ctrl* key and clicking on them (make sure you only select files of one type, either all HTML files or all CSS files and so on). Then right click and select *merge*. That's all, calibre will merge the files, automatically taking care of migrating all links and references to the merged files. Note that merging files can sometimes cause text styling to change, since the individual files could have used different stylesheets.

You can also select text files and then drag and drop the text files onto another text file to merge the dropped text files into the target text file.

5.2.3 Changing text file order

You can re-arrange the order in which text (HTML) files are opened when reading the book by simply dragging and dropping them in the *File browser* or clicking on the file to move and then pressing the `Ctrl+Shift` modifiers with the Up, Down, Home or End keys. For the technically inclined, this is called re-ordering the book spine.

Note that you have to drop the items *between* other items, not on top of them, this can be a little fiddly until you get used to it. Dropping on top of another file will cause the files to be merged.

5.2.4 Marking the cover

E-books typically have a cover image. This image is indicated in the *File browser* by the icon of a brown book next to the image name. If you want to designate some other image as the cover, you can do so by right clicking on the file and choosing *Mark as cover*.

In addition, EPUB files has the concept of a *titlepage*. A title page is a HTML file that acts as the title page/cover for the book. You can mark an HTML file as the titlepage when editing EPUBs by right-clicking. Be careful that the file you mark contains only the cover information. If it contains other content, such as the first chapter, then that content will be lost if the user ever converts the EPUB file in calibre to another format. This is because when converting, calibre assumes that the marked title page contains only the cover and no other content.

5.2.5 Deleting files

You can delete files by either right clicking on them or by selecting them and pressing the Delete key. Deleting a file removes all references to the file from the OPF file, saving you that chore. However, references in other places are not removed, you can use the Check Book tool to easily find and remove/replace them.

5.2.6 Exporting files

You can export a file from inside the book to somewhere else on your computer. This is useful if you want to work on the file in isolation, with specialised tools. To do this, simply right click on the file and choose *Export*.

Once you are done working on the exported file, you can re-import it into the book, by right clicking on the file again and choosing *Replace with file...* which will allow you to replace the file in the book with the previously exported file.

You can also copy files between multiple editor instances. Select the files you want to copy in the *File browser*, then right click and choose, *Copy selected files to another editor instance*. Then, in the other editor instance, right click in the *File browser* and choose *Paste file from other editor instance*.

5.2.7 Adding new images/fonts/etc. or creating new blank files

You can add a new image, font, stylesheet, etc. from your computer into the book by clicking *File → New file*. This lets you either import a file by clicking the *Import resource file* button or create a new blank HTML file or stylesheet by simply entering the file name into the box for the new file.

You can also import multiple files into the book at once using *File->Import files into book*.

5.2.8 Replacing files

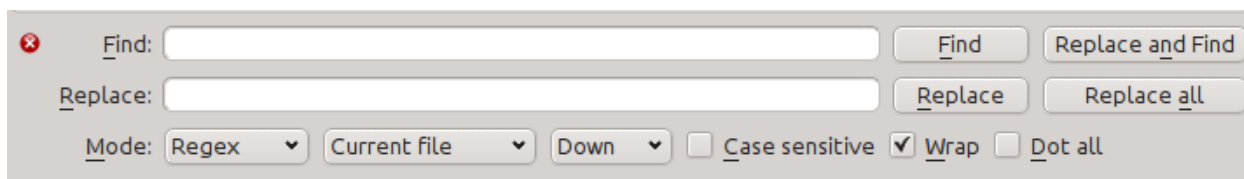
You can easily replace existing files in the book, by right clicking on the file and choosing replace. This will automatically update all links and references, in case the replacement file has a different name than the file being replaced.

5.2.9 Linking stylesheets to HTML files efficiently

As a convenience, you can select multiple HTML files in the File browser, right click and choose Link stylesheets to have calibre automatically insert the <link> tags for those stylesheets into all the selected HTML files.

5.3 Wyszukaj i zastąp

Edit book has a very powerful search and replace interface that allows you to search and replace text in the current file, across all files and even in a marked region of the current file. You can search using a normal search or using regular expressions. To learn how to use regular expressions for advanced searching, see *Wszystko, co chcielibyście wiedzieć o wyrażeniach regularnych w calibre* (strona 213).



Start the search and replace via the *Search → Find/replace* menu entry (you must be editing an HTML or CSS file).

Type the text you want to find into the Find box and its replacement into the Replace box. You can then click the appropriate buttons to Find the next match, replace the current match and replace all matches.

Using the drop downs at the bottom of the box, you can have the search operate over the current file, all text files, all style files or all files. You can also choose the search mode to be a normal (string) search or a regular expression search.

You can count all the matches for a search expression via *Search → Count all*. The count will run over whatever files/regions you have selected in the dropdown box.

You can also go to a specific line in the currently open editor via *Search → Go to line*.

Informacja: Remember, to harness the full power of search and replace, you will need to use regular expressions. See *Wszystko, co chcielibyście wiedzieć o wyrażeniach regularnych w calibre* (strona 213).

5.3.1 Zapisane wyszukiwania

You can save frequently used search/replace expressions (including function mode expressions) and reuse them multiple times. To save a search simply right click in the Find box and select *Save current search*.

You can bring up the saved searches via *Search → Saved searches*. This will present you with a list of search and replace expressions that you can apply. You can even select multiple entries in the list by holding down the **Ctrl** key while clicking so as to run multiple search and replace expressions in a single operation.

5.3.2 Function mode

Function mode allows you to write arbitrarily powerful Python functions that are run on every Find/replace. You can do pretty much any text manipulation you like in function mode. For more information, see *Tryb funkcji Wyszukaj i zamień w Edytorze* (strona 99).

5.3.3 Search ignoring HTML tags

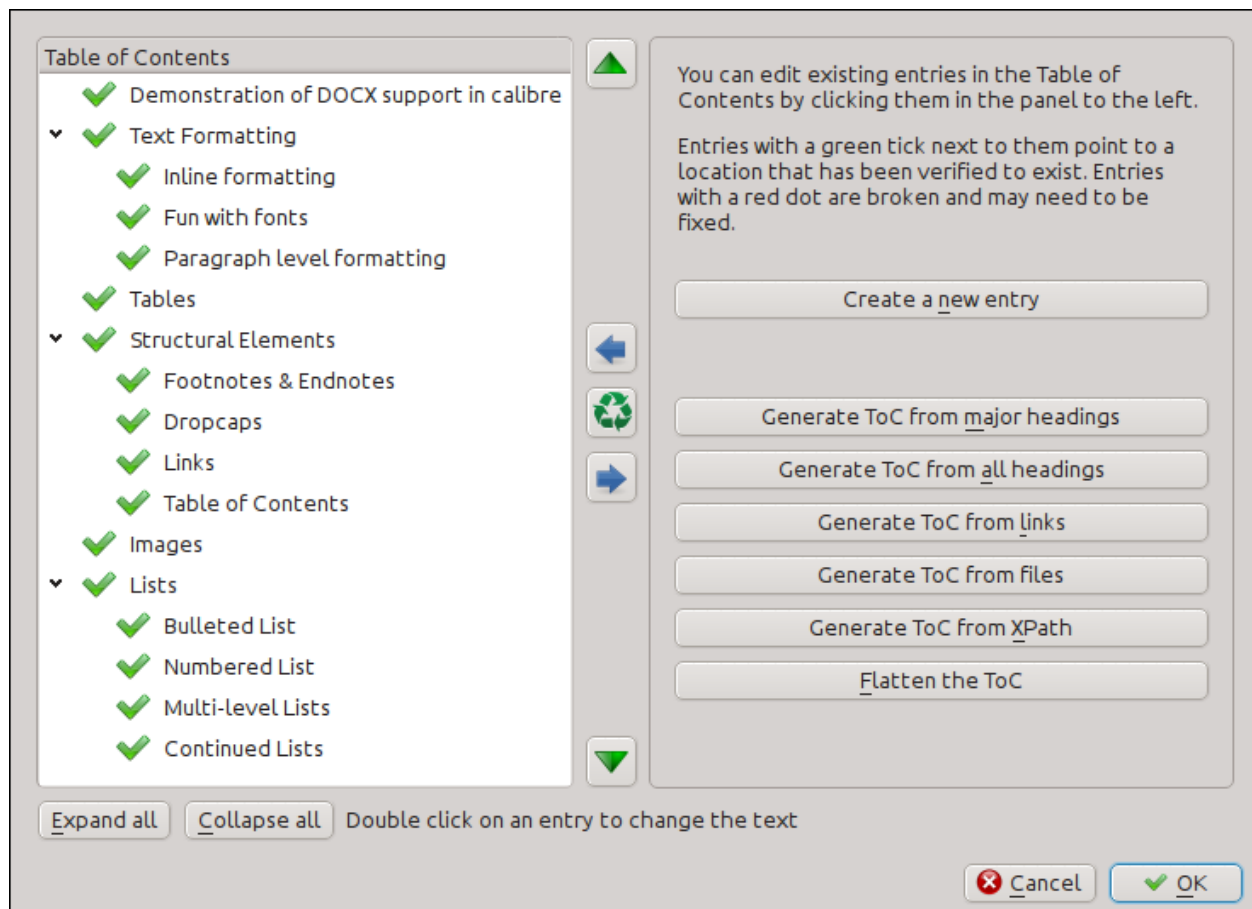
There is also a dedicated tool for searching for text, ignoring any HTML tags in between. For example, if the book has the HTML `Emphasis on a <i>word</i>`, you can search for `on a word` and it will be found even though there is an `<i>` tag in the middle. Use this tool via the *Search → Search ignoring HTML markup* menu item.

5.4 Automated tools

Edit book has various tools to help with common tasks. These are accessed via the *Tools* menu.

5.4.1 Editing the Table of Contents

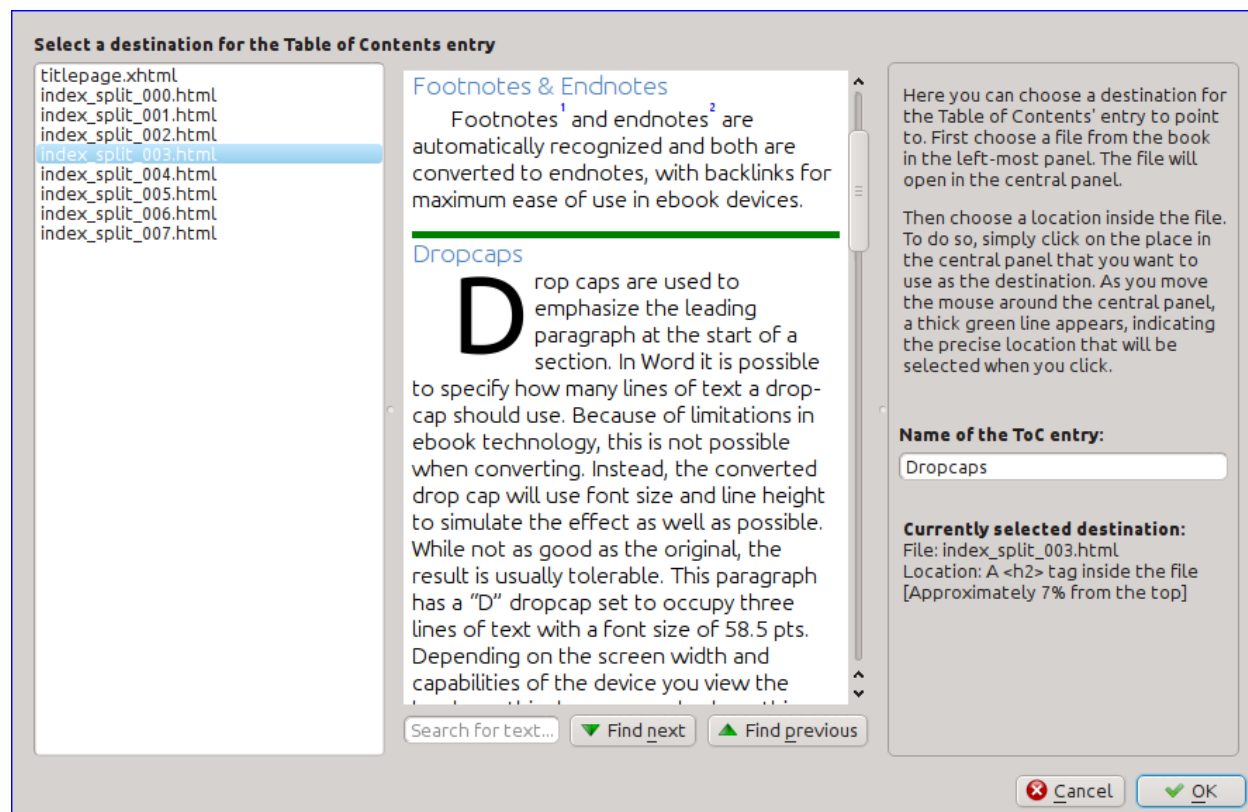
There is a dedicated tool to ease editing of the Table of Contents. Launch it with *Tools → Table of Contents → Edit Table of Contents*.



The Edit Table of Contents tool shows you the current Table of Contents (if any) on the left. Simply double click on any entry to change its text. You can also re-arrange entries by drag and drop or by using the buttons to the right.

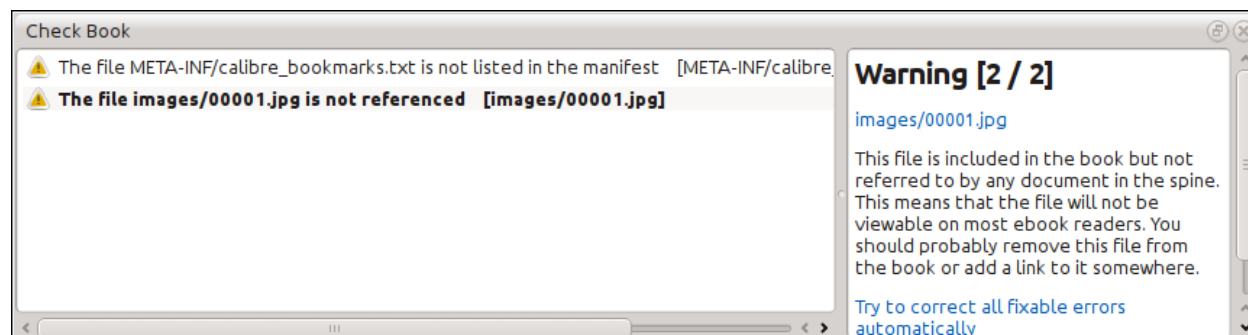
For books that do not have a pre-existing Table of Contents, the tool gives you various options to auto-generate a Table of Contents from the text. You can generate from the headings in the document, from links, from individual files and so on.

You can edit individual entries by clicking on them and then clicking the *Change the location this entry points to* button. This will open up a mini-preview of the book, simply move the mouse cursor over the book view panel, and click where you want the entry to point to. A thick green line will show you the location. Click OK once you are happy with the location.



5.4.2 Checking the book

The *Check book* tool searches your book for problems that could prevent it working as intended on actual reader devices. Activate it via *Tools* → *Check book*.



Any problems found are reported in a nice, easy to use list. Clicking any entry in the list shows you some help about that error as well as giving you the option to auto-fix that error, if the error can be fixed automatically. You can also double click the error to open the location of the error in an editor, so you can fix it yourself.

Some of the checks performed are:

- Malformed HTML markup. Any HTML markup that does not parse as well-formed XML is reported. Correcting it will ensure that your markup works as intended in all contexts. calibre can also auto-fix these errors, but auto-fixing can sometimes have unexpected effects, so use with care. As always, a checkpoint is created before auto-fixing so you can easily revert all changes. Auto-fixing works by parsing the markup using the HTML5 algorithm, which is highly fault tolerant and then converting to well formed XML.
- Malformed or unknown CSS styles. Any CSS that is not valid or that has properties not defined in the CSS 2.1 standard (plus a few from CSS 3) are reported. CSS is checked in all stylesheets, inline style attributes and `<style>` tags in HTML files.
- Broken links. Links that point to files inside the book that are missing are reported.
- Unreferenced files. Files in the book that are not referenced by any other file or are not in the spine are reported.
- Various common problems in OPF files such as duplicate spine or manifest items, broken idrefs or meta cover tags, missing required sections and so on.
- Various compatibility checks for known problems that can cause the book to malfunction on reader devices.

5.4.3 Adding a cover

You can easily add a cover to the book via *Tools* → *Add cover*. This allows you to either choose an existing image in the book as the cover or import a new image into the book and make it the cover. When editing EPUB files, the HTML wrapper for the cover is automatically generated. If an existing cover in the book is found, it is replaced. The tool also automatically takes care of correctly marking the cover files as covers in the OPF.

5.4.4 Osadzanie użytych czcionek

Accessed via *Tools* → *Embed reference fonts*, this tool finds all fonts referenced in the book and if they are not already embedded, searches your computer for them and embeds them into the book, if found. Please make sure that you have the necessary copyrights for embedding commercially licensed fonts, before doing this.

5.4.5 Redukowanie osadzonych czcionek

Accessed via *Tools* → *Subset embedded fonts*, this tool reduces all the fonts in the book to only contain glyphs for the text actually present in the book. This commonly reduces the size of the font files by ~ 50%. However, be aware that once the fonts are subset, if you add new text whose characters are not previously present in the subset font, the font will not work for the new text. So do this only as the last step in your workflow.

5.4.6 Inteligentna interpunkcja

Convert plain text dashes, ellipsis, quotes, multiple hyphens, etc. into their typographically correct equivalents. Note that the algorithm can sometimes generate incorrect results, especially when single quotes at the start of contractions are involved. Accessed via *Tools* → *Smarten punctuation*.

5.4.7 Transforming CSS properties

Create rules to transform the styling of the book. For example, create a rule to convert all red text to green or to double the font size of all text in the book or make text of a certain font family italic, etc.

Creating the rules is simple, the rules follow a natural language format, that looks like:

- If the property *color* is *red* change it to *green*
- If the property *font-size* is *any value* multiply the value by 2

Accessed via *Tools* → *Transform styles*.

5.4.8 Usuwanie nieużywanych stylów

Remove all unused CSS rules from stylesheets and <style> tags. Some books created from production templates can have a large number of extra CSS rules that don't match any actual content. These extra rules can slow down readers that need to process them all. Accessed via *Tools* → *Remove unused CSS*.

5.4.9 Fixing HTML

This tool simply converts HTML that cannot be parsed as XML into well-formed XML. It is very common in e-books to have non-well-formed XML, so this tool simply automates the process of fixing such HTML. The tool works by parsing the HTML using the HTML5 algorithm (the algorithm used in all modern browsers) and then converting the result into XML. Be aware that auto-fixing can sometimes have counter-intuitive results. If you prefer, you can use the Check Book tool discussed above to find and manually correct problems in the HTML. Accessed via *Tools* → *Fix HTML*.

5.4.10 Beautifying files

This tool is used to auto-format all HTML and CSS files so that they „look pretty”. The code is auto-indented so that it lines up nicely, blank lines are inserted where appropriate and so on. Note that beautifying also auto-fixes broken HTML/CSS. Therefore, if you don't want any auto-fixing to be performed, first use the Check Book tool to correct all problems and only then run beautify. Accessed via *Tools* → *Beautify all files*.

Informacja: In HTML any text can have significant whitespace, via the CSS white-space directive. Therefore, beautification could potentially change the rendering of the HTML. To avoid this as far as possible, the beautify algorithm only beautifies block level tags that contain other block level tags. So, for example, text inside a <p> tag will not have its whitespace changed. But a <body> tag that contains only other <p> and <div> tags will be beautified. This can sometimes mean that a particular file will not be affected by beautify as it has no suitable block level tags. In such cases you can try different beautification tools, that are less careful, for example: [HTML Tidy](https://infohound.net/tidy/)³⁵.

³⁵ <https://infohound.net/tidy/>

5.4.11 Inserting an inline Table of Contents

Normally in e-books, the Table of Contents is separate from the main text and is typically accessed via a special Table of Contents button/menu in the e-book reading device. You can also have calibre automatically generate an *inline* Table of Contents that becomes part of the text of the book. It is generated based on the currently defined Table of Contents.

If you use this tool multiple times, each invocation will cause the previously created inline Table of Contents to be replaced. The tool can be accessed via *Tools* → *Table of Contents* → *Insert inline Table of Contents*.

5.4.12 Setting Semantics

This tool is used to set *semantics* in EPUB files. Semantics are simply, links in the OPF file that identify certain locations in the book as having special meaning. You can use them to identify the foreword, dedication, cover, table of contents, etc. Simply choose the type of semantic information you want to specify and then select the location in the book the link should point to. This tool can be accessed via *Tools* → *Set semantics*.

5.4.13 Filtering style information

This tool can be used to easily remove specified CSS style properties from the entire book. You can tell it what properties you want removed, for example, `color`, `background-color`, `line-height` and it will remove them from everywhere they occur — stylesheets, `<style>` tags and inline `style` attributes. After removing the style information, a summary of all the changes made is displayed so you can see exactly what was changed. The tool can be accessed via *Tools* → *Filter style information*.

5.4.14 Upgrading the book's internals

This tool can be used to upgrade the book's internals, if possible. For instance it will upgrade EPUB 2 books to EPUB 3 books. The tool can be accessed via *Upgrade book internals*.

5.5 Punkty kontrolne

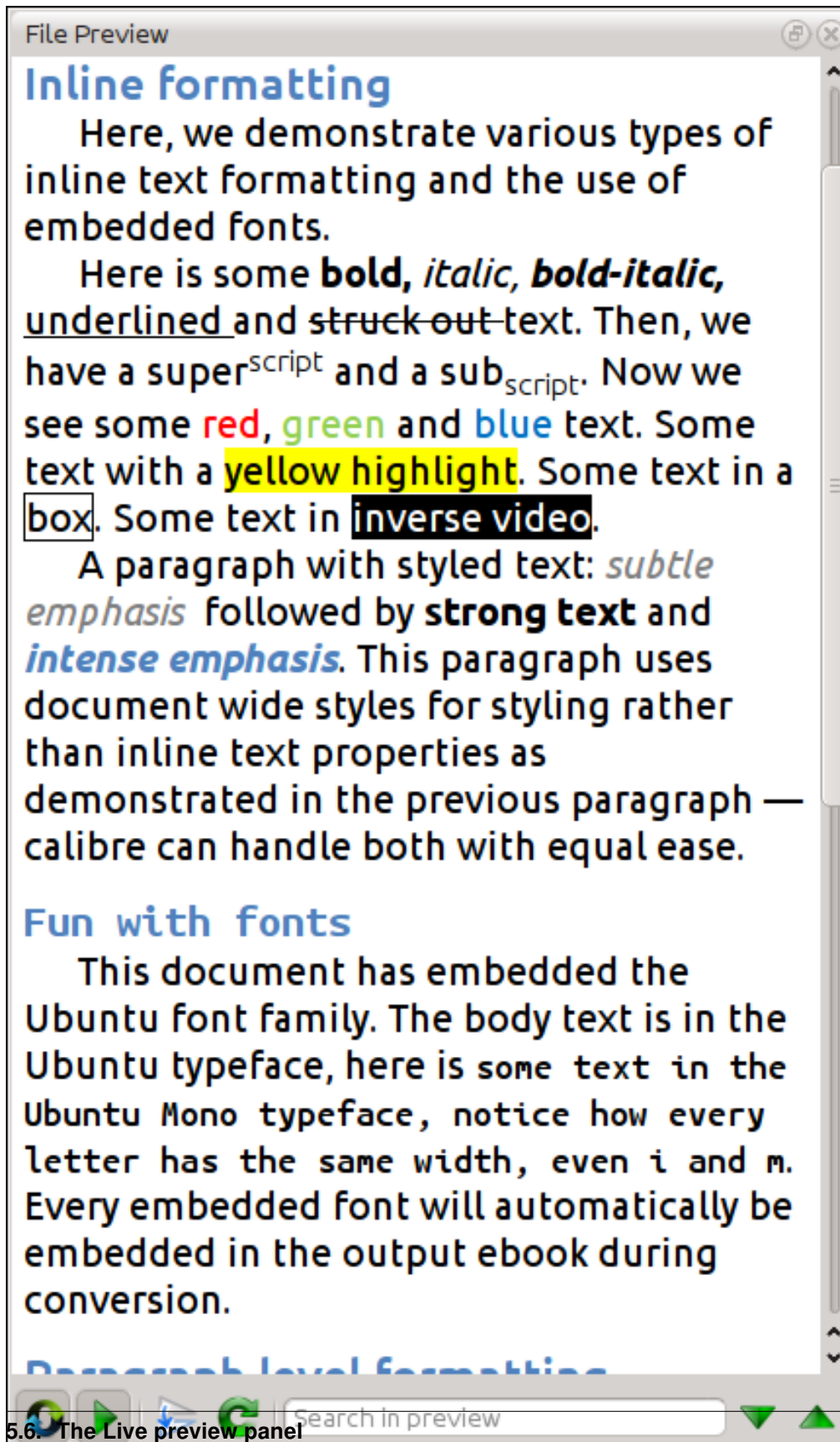
Checkpoints are a way to mark the current state of the book as „special”. You can then go on to do whatever changes you want to the book and if you don't like the results, return to the checkpointed state. Checkpoints are automatically created every time you run any of the automated tools described in the previous section.

You can create a checkpoint via *Edit* → *Create checkpoint*. And go back to a previous checkpoint with *Edit* → *Revert to ...*

The check pointing functionality is in addition to the normal Undo/redo mechanism when editing individual files. Checkpoints are needed for when changes are spread over multiple files in the book or when you wish to be able to revert a large group of related changes as a whole.

You can see a list of available checkpoints via *View* → *Checkpoints*. You can compare the current state of the book to a specified checkpoint using the *Porównanie e-booków* (strona 121) tool – by selecting the checkpoint of interest and clicking the *Compare* button. The *Revert to* button restores the book to the selected checkpoint, undoing all changes since that checkpoint was created.

5.6 The Live preview panel



The *File preview* gives you an overview of the various files inside The live preview panel shows you the changes you are making live (with a second or two of delay). As you edit HTML or CSS files, the preview panel is updated automatically to reflect your changes. As you move the cursor around in the editor, the preview panel will track its location, showing you the corresponding location in the book. Clicking in the preview panel, will cause the cursor in the editor to be positioned over the element you clicked. If you click a link pointing to another file in the book, that file will be opened in the edit and the preview panel, automatically.

You can turn off the automatic syncing of position and live preview of changes – by buttons under the preview panel. The live update of the preview panel only happens when you are not actively typing in the editor, so as not to be distracting or slow you down, waiting for the preview to render.

The preview panel shows you how the text will look when viewed. However, the preview panel is not a substitute for actually testing your book on an actual reader device. It is both more, and less capable than an actual reader. It will tolerate errors and sloppy markup much better than most reader devices. It will also not show you page margins, page breaks and embedded fonts that use font name aliasing. Use the preview panel while you are working on the book, but once you are done, review it in an actual reader device or software emulator.

Informacja: The preview panel does not support embedded fonts if the name of the font inside the font file does not match the name in the CSS @font-face rule. You can use the Check Book tool to quickly find and fix any such problem fonts.

5.6.1 Splitting HTML files

One, perhaps non-obvious, use of the preview panel is to split long HTML files. While viewing the file you want to split,

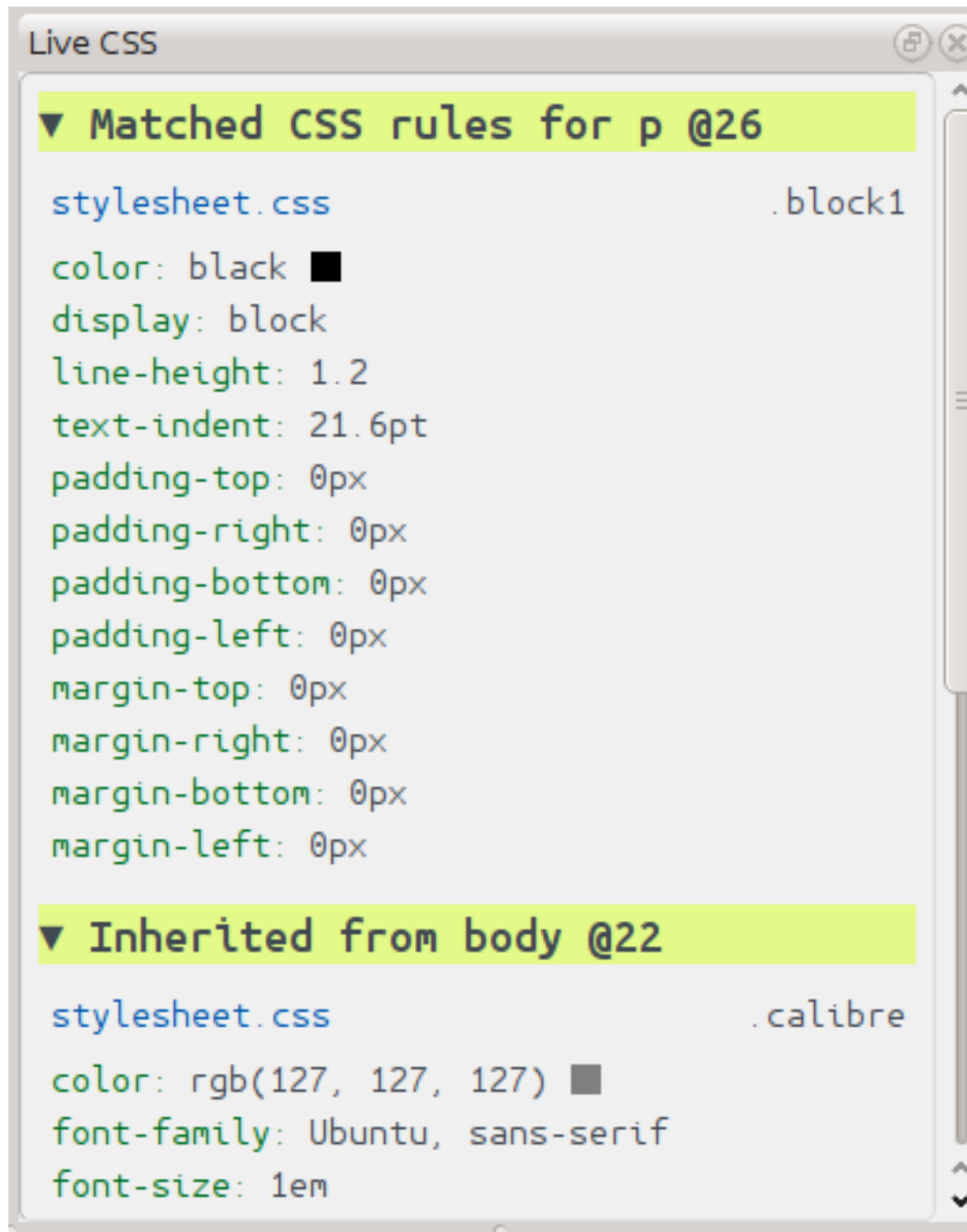


click the *Split mode* button under the preview panel. Then simply move your mouse to the place where you want to split the file and click. A thick green line will show you exactly where the split will happen as you move your mouse. Once you have found the location you want, simply click and the split will be performed.

Splitting the file will automatically update all links and references that pointed into the bottom half of the file and will open the newly split file in an editor.

You can also split a single HTML file at multiple locations automatically, by right clicking inside the file in the editor and choosing *Split at multiple locations*. This will allow you to easily split a large file at all heading tags or all tags having a certain class and so on.

5.7 The Live CSS panel



The *Live CSS* panel shows you all the style rules that apply to the tag you are currently editing. The name of tag, along with its line number in the editor are displayed, followed by a list of matching style rules.

It is a great way to quickly see which style rules apply to any tag. The view also has clickable links (in blue), which take you directly to the location where the style was defined, in case you wish to make any changes to the style rules. Style rules that apply directly to the tag, as well as rules that are inherited from parent tags are shown.

The panel also shows you what the finally calculated styles for the tag are. Properties in the list that are superseded by higher priority rules are shown with a line through them.

You can enable the Live CSS panel via *View* → *Live CSS*.

5.8 Miscellaneous tools

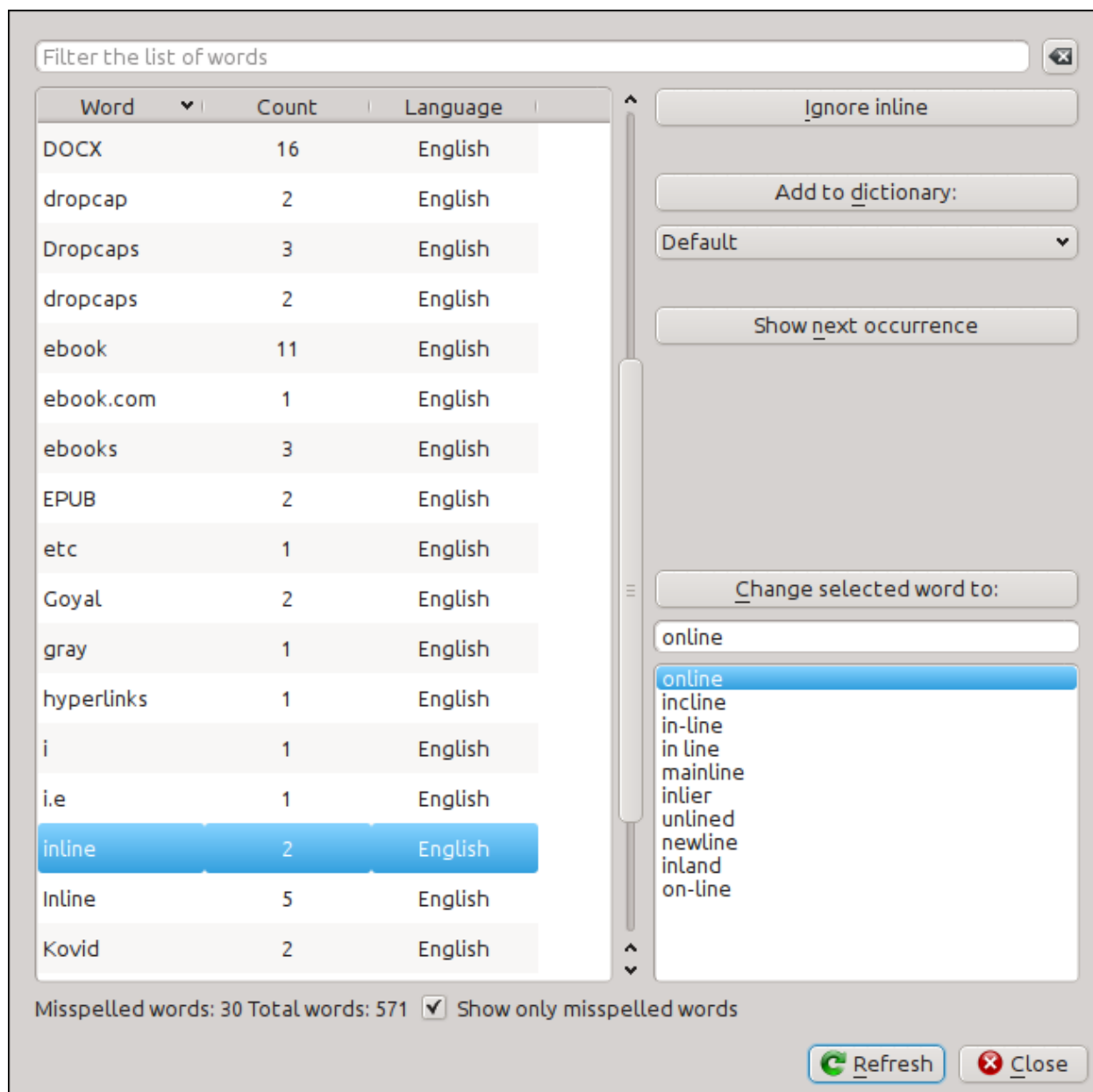
There are a few more tools that can be useful while you edit the book.

5.8.1 The Table of Contents view

The Table of Contents view shows you the current table of contents in the book. Double clicking on any entry opens the place that entry points to in an editor. You can right click to edit the Table of Contents, refresh the view or expand/collapse all items. Access this view via *View → Table of Contents*.

5.8.2 Checking the spelling of words in the book

You can run a spelling checker via *Tools → Check spelling*.



Words are shown with the number of times they occur in the book and the language the word belongs to. Language information is taken from the books metadata and from `lang` attributes in the HTML files. This allows the spell checker to work well even with books that contain text in multiple languages. For example, in the following HTML extract, the word `color` will be checked using American English and the word `colour` using British English:

```
<div lang="en_US">color <span lang="en_GB">colour</span></div>
```

Informacja: You can double click a word to highlight the next occurrence of that word in the editor. This is useful if you wish to manually edit the word, or see what context it is in.

To change a word, simply double click one of the suggested alternative spellings on the right, or type in your own corrected spelling and click the *Change selected word to* button. This will replace all occurrences of the word in the book. You can also right click on a word in the main word list to change the word conveniently from the right click menu.

You can have the spelling checker ignore a word for the current session by clicking the *Ignore* button. You can also add a word to the user dictionary by clicking the *Add to dictionary* button. The spelling checker supports multiple user dictionaries, so you can select the dictionary you want the word added to.

You can also have the spelling checker display all the words in your book, not just the incorrectly spelled ones. This is useful to see what words are most common in your book and to run a simple search and replace on individual words.

Informacja: If you make any changes to the book by editing files while the spell check tool is open, you should click the *Refresh* button in the Spell check tool. If you do not do this and continue to use the Spell check tool, you could lose the changes you have made in the editor.

Informacja: To exclude an individual file from being spell checked when running the spell check tool, you can use the *Exclude files* button or add the following comment just under the opening tag in the file:

```
<!-- calibre-no-spell-check -->
```

Dodanie nowych słowników

The spelling checker comes with builtin dictionaries for the English and Spanish languages. You can install your own dictionaries via *Preferences* → *Editor* → *Manage spelling dictionaries*. The spell checker can use dictionaries from the LibreOffice program (in the .oxf format). You can download these dictionaries from [The LibreOffice Extensions repository](#)³⁶.

5.8.3 Inserting special characters

You can insert characters that are difficult to type by using the *Edit* → *Insert special character* tool. This shows you all Unicode characters, simply click on the character you want to type. If you hold `Ctrl` while clicking, the window will close itself after inserting the selected character. This tool can be used to insert special characters into the main text or into any other area of the user interface, such as the Search and replace tool.

Because there are a lot of characters, you can define your own *Favorite* characters, that will be shown first. Simply right click on a character to mark it as favorite. You can also right click on a character in favorites to remove it from favorites. Finally, you can re-arrange the order of characters in favorites by clicking the *Re-arrange favorites* button and then drag and dropping the characters in favorites around.

You can also directly type in special characters using the keyboard. To do this, you type the Unicode code for the character (in hexadecimal) and then press the `Alt+X` key which will convert the previously typed code into the corresponding character. For example, to type `ÿ` you would type `ff` and then `Alt+X`. To type a non-breaking space you would use `a0` and then `Alt+X`, to type the horizontal ellipsis you would use `2026` and `Alt+X` and so on.

Finally, you can type in special characters by using HTML named entities. For example, typing ` ` will be replaced by a non breaking space when you type the semi-colon. The replacement happens only when typing the semi-colon.

³⁶ <https://extensions.libreoffice.org/?Tags%5B%5D=50>

5.8.4 The code inspector view

This view shows you the HTML coding and CSS that applies to the current element of interest. You open it by right clicking a location in the preview panel and choosing *Inspect*. It allows you to see the HTML coding for that element and more importantly, the CSS styles that apply to it. You can even dynamically edit the styles and see what effect your changes have instantly. Note that editing the styles does not actually make changes to the book contents, it only allows for quick experimentation. The ability to live edit inside the Inspector is under development.

5.8.5 Checking external links

You can use this tool to check all links in your book that point to external websites. The tool will try to visit every externally linked website, and if the visit fails, it will report all broken links in a convenient format for you to fix.

5.8.6 Downloading external resources

You can use this tool to automatically download any images/stylesheets/etc. in the book that are not bundled with the book (i.e. they have URLs pointing to a location on the internet). The tool will find all such resources and automatically download them, add them to the book and replace all references to them to use the downloaded files.

5.8.7 Arranging files into folders by type

Often when editing EPUB files that you get from somewhere, you will find that the files inside the EPUB are arranged haphazardly, in different sub-folders. This tool allows you to automatically move all files into sub-folders based on their types. Access it via *Tools* → *Arrange into folders*. Note that this tool only changes how the files are arranged inside the EPUB, it does not change how they are displayed in the File browser.

5.8.8 Importing files in other e-book formats as EPUB

The editor includes the ability to import files in some other e-book formats directly as a new EPUB, without going through a full conversion. This is particularly useful to directly create EPUB files from your own hand-edited HTML files. You can do this via *File* → *Import an HTML or DOCX file as a new book*.

Tryb funkcji Wyszukaj i zamień w Edytorze

The *Search & replace* tool in the editor support a *function mode*. In this mode, you can combine regular expressions (see *Wszystko, co chcielibyście wiedzieć o wyrażeniach regularnych w calibre* (strona 213)) with arbitrarily powerful Python functions to do all sorts of advanced text processing.

In the standard *regex* mode for search and replace, you specify both a regular expression to search for as well as a template that is used to replace all found matches. In function mode, instead of using a fixed template, you specify an arbitrary function, in the *Python programming language*³⁷. This allows you to do lots of things that are not possible with simple templates.

Techniki wykorzystania trybu funkcji oraz używana do tego składnia zostaną przedstawione w serii przykładów, pokazując jak tworzyć funkcje, by wykonywać coraz bardziej złożone zadania.

³⁷ <https://docs.python.org>

Automatyczne poprawianie wielkości liter w nagłówkach dokumentu

Tu wykorzystamy jedną z wbudowanych funkcji edytora do automatycznej zmiany wielkości liter we wszystkich znacznikach tytułów tak, by każda pierwsza litera wyrazu była wielka:

Find expression: `<([Hh][1-6])[^>]*>.+?</\1>`

Dla funkcji wybierz po prostu wbudowaną funkcję *Title-case ignore tags*). To zamieni tytuły z `<h1>jakiś TYTUŁ</h1>` na `<h1>Jakiś Tytuł</h1>`. Działa to nawet wtedy, gdy wewnątrz znaczników nagłówków są jakieś inne znaczniki.

Twoja pierwsza własna funkcja - poprawianie myślników

Prawdziwy potencjał trybu funkcji ujawnia się wraz z możliwością pisania własnych funkcji do dowolnego przetwarzania tekstu. Inteligentna Interpunkcja w edytorze pozostawia pojedyncze myślniki, możesz jej więc użyć do zamiany ich na znak pauzy.

To create a new function, simply click the *Create/edit* button to create a new function and copy the Python code from below.

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
            **kwargs):
    return match.group().replace('--', '-').replace('-', '—')
```

Every *Search & replace* custom function must have a unique name and consist of a Python function named `replace`, that accepts all the arguments shown above. For the moment, we won't worry about all the different arguments to `replace()` function. Just focus on the `match` argument. It represents a match when running a search and replace. Its full documentation is available [here](https://docs.python.org/library/re.html#match-objects)³⁸. `match.group()` simply returns all the matched text and all we do is replace hyphens in that text with em-dashes, first replacing double hyphens and then single hyphens.

Użyj tej funkcji z wyrażeniem regularnym:

`>[^<>]+<`

Zamieni ona wszystkie myślniki na pauzy, ale tylko we właściwym tekście, nie wewnątrz definicji znaczników.

³⁸ <https://docs.python.org/library/re.html#match-objects>

Potęga trybu funkcji - użycie słownika do poprawienia błędnie podzielonych słów.

Often, e-books created from scans of printed books contain mis-hyphenated words – words that were split at the end of the line on the printed page. We will write a simple function to automatically find and fix such words.

```
import regex
from calibre import replace_entities
from calibre import prepare_string_for_xml

def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
↳ **kwargs):

    def replace_word(wmatch):
        # Try to remove the hyphen and replace the words if the resulting
        # hyphen free word is recognized by the dictionary
        without_hyphen = wmatch.group(1) + wmatch.group(2)
        if dictionaries.recognized(without_hyphen):
            return without_hyphen
        return wmatch.group()

    # Search for words split by a hyphen
    text = replace_entities(match.group()[1:-1]) # Handle HTML entities like &
    corrected = regex.sub(r'(\w+)\s*-\s*(\w+)', replace_word, text, flags=regex.
↳ VERSION1 | regex.UNICODE)
    return '>%s<' % prepare_string_for_xml(corrected) # Put back required entities
```

Użyj tej funkcji z tym samym poszukiwanym wyrażeniem jak poprzednio, mianowicie:

```
>[^<>]+<
```

W magiczny sposób poprawi to wszystkie błędnie podzielone słowa w tekście. Trik polega na użyciu jednego z argumentów funkcji, `dictionaries`. Stanowi on odniesienie do słowników, których używa edytor do sprawdzania pisowni. Funkcja odszukuje słowo z myślnikiem w środku, usuwa ów myślnik i sprawdza, czy tak połączony wyraz jest w słowniku. Jeśli tak to oryginalny wyraz jest zastępowany nowym, połączonym.

Należy pamiętać, że działanie tej funkcji jest ograniczone do książek w jednym języku, bowiem domyślnie `dictionaries.recognized()` używa głównego języka książki.

Autonumerowanie rozdziałów

Teraz coś innego. Przypuśćmy, że plik HTML ma wiele rozdziałów, każdy rozpoczynający się znacznikiem `<h2>` w postaci `<h2>Jakiś tekst</h2>`. Można napisać funkcję, która ponumeruje te rozdziały kolejnymi numerami, by wyglądały następująco: `<h2>1. Jakiś tekst</h2>`.

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
↳ **kwargs):
    section_number = '%d. ' % number
    return match.group(1) + section_number + match.group(2)

# Ensure that when running over multiple files, the files are processed
# in the order in which they appear in the book
replace.file_order = 'spine'
```

Użyj jej, wyszukując wyrażenie:

```
(?s) (<h2[^\<>]*>) (.+?</h2>)
```

Ustaw kursor na górze pliku i kliknij *Zamień wszystkie*

Ta funkcja wykorzystuje inny przydatny argument funkcji `replace()`: `number`. Przy operacji *Zamień wszystkie* liczba jest automatycznie inkrementowana przy każdym dopasowaniu.

Kolejną nowością jest użycie `replace.file_order` – ustawienie tej opcji na `spine` oznacza, że jeśli przeszukiwanych jest wiele plików to będą one przetwarzane w takiej kolejności, w jakiej występują w książce. Szczegóły znajdziesz w *Wybierz kolejność plików, jeśli przetwarzanych jest wiele plików HTML* (strona 105).

Automatyczne tworzenie spisu treści

Na koniec spróbujmy czegoś bardziej ambitnego. Przypuśćmy, że w książce tytuły są w znacznikach `h1` i `h2`, które wyglądają następująco: `<h1 id="jakiesid">Jakiś tekst</h1>`. Wygenerujemy automatycznie spis treści w HTML, bazując na tych znacznikach. Stwórz funkcję jak poniżej:

```
from calibre import replace_entities
from calibre.ebooks.oeb.polish.toc import TOC, toc_to_html
from calibre.gui2.tweak_book import current_container
from calibre.ebooks.oeb.base import xml2str

def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
    **kwargs):
    if match is None:
        # All matches found, output the resulting Table of Contents.
        # The argument metadata is the metadata of the book being edited
        if 'toc' in data:
            toc = data['toc']
            root = TOC()
            for (file_name, tag_name, anchor, text) in toc:
                parent = root.children[-1] if tag_name == 'h2' and root.children else
            root
                parent.add(text, file_name, anchor)
            toc = toc_to_html(root, current_container(), 'toc.html', 'Table of
            Contents for ' + metadata.title, metadata.language)
            print(xml2str(toc))
        else:
            print('No headings to build ToC from found')
    else:
        # Add an entry corresponding to this match to the Table of Contents
        if 'toc' not in data:
            # The entries are stored in the data object, which will persist
            # for all invocations of this function during a 'Replace All' operation
            data['toc'] = []
            tag_name, anchor, text = match.group(1), replace_entities(match.group(2)),
            replace_entities(match.group(3))
            data['toc'].append((file_name, tag_name, anchor, text))
            return match.group() # We don't want to make any actual changes, so return
            the original matched text

# Ensure that we are called once after the last match is found so we can
# output the ToC
replace.call_after_last_match = True
# Ensure that when running over multiple files, this function is called,
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
# the files are processed in the order in which they appear in the book
replace.file_order = 'spine'
```

Użyj takiego szukanego wyrażenia:

```
<(h[12]) [^<>]* id=['"]([^\"]+)['] [^<>]*>([<>]+)
```

Run the search on *All text files* and at the end of the search, a window will popup with „Debug output from your function” which will have the HTML Table of Contents, ready to be pasted into `toc.html`.

The function above is heavily commented, so it should be easy to follow. The key new feature is the use of another useful extra argument to the `replace()` function, the `data` object. The `data` object is a Python *dictionary* that persists between all successive invocations of `replace()` during a single *Replace All* operation.

Kolejna rzecz to właściwość `call_after_last_match` – ustawienie jej na `True` oznacza, że edytor wywoła funkcję `replace()` dodatkowo jeden raz po odnalezieniu wszystkich dopasowań. W tym wywołaniu obiekt dopasowania jest równy `None`.

To tylko demonstracja możliwości trybu funkcji, jeśli rzeczywiście chcesz wygenerować spis treści z nagłówków, lepiej będzie posłużyć się dedykowanym temu narzędziem *Narzędzia* → *Spis treści*.

API trybu funkcji

All function mode functions must be Python functions named `replace`, with the following signature:

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
    ↪ **kwargs):
    return a_string
```

Kiedy uruchamiane jest wyszukiwanie/zamiana, dla każdego odnalezionego dopasowania, dla którego zostanie wywołana funkcja `replace()`, musi ona zwrócić tekst zastępujący owo dopasowanie. Jeśli zamiana nie zostanie wykonana, powinna zwrócić `match.group()`, który jest oryginalnym tekstem. Różne argumenty funkcji `replace()` zostały opisane poniżej.

Argument `match`

The `match` argument represents the currently found match. It is a [Python Match object](#)³⁹. Its most useful method is `group()` which can be used to get the matched text corresponding to individual capture groups in the search regular expression.

Argument `number`

Argument `number` jest numerem aktualnego dopasowania. Kiedy uruchamiasz *Zamień wszystko*, każde kolejne dopasowanie będzie wywoływać `replace()` z następnym numerem. Pierwsze dopasowanie ma numer 1.

³⁹ <https://docs.python.org/library/re.html#match-objects>

Argument `file_name`

To jest nazwa pliku, w którym odnaleziono bieżące dopasowanie. Przy wyszukiwaniu w zaznaczeniu `file_name` jest puste. `file_name` ma postać kanonicznej względnej ścieżki, od katalogu głównego książki, do oddzielania katalogów służy `/`.

Argument `metadata`

Ten argument reprezentuje metadane aktualnej książki, takie jak tytuł, autor, język itp. Jest to obiekt klasy `calibre.ebooks.metadata.book.base.Metadata` (strona 209). Najbardziej przydatne właściwości to `title`, `authors` (lista autorów) i `language` (kod języka).

Argument `dictionaries`

This represents the collection of dictionaries used for spell checking the current book. Its most useful method is `dictionaries.recognized(word)` which will return `True` if the passed in word is recognized by the dictionary for the current book's language.

Argument `data`

This a simple Python dictionary. When you run *Replace all*, every successive match will cause `replace()` to be called with the same dictionary as data. You can thus use it to store arbitrary data between invocations of `replace()` during a *Replace all* operation.

Argument `functions`

Ten argument daje ci dostęp do wszystkich innych zdefiniowanych przez siebie funkcji. Przydaje się to do wielokrotnego użycia tego samego kodu. Możesz zdefiniować jakąś funkcję w jednym miejscu i wykorzystać ją w innych twoich funkcjach. Na przykład przypuśćmy, że stworzyłeś taką funkcję o nazwie `My function`:

```
def utility():
    # do something

def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
    **kwargs):
    ...
```

Potem, w innej funkcji, możesz uzyskać dostęp do `utility()` w ten sposób:

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
    **kwargs):
    utility = functions['My Function']['utility']
    ...
```

Można również użyć funkcji do przechowywania trwałych danych, które mogą być ponownie wykorzystane przez inne funkcje. Na przykład jedna funkcja zbiera dane przy uruchomieniu *Zamień wszystko*, a następna, uruchomiona później, wykorzystuje je. Przyjrzyjmy się dwóm następującym funkcjom:

```
# Function One
persistent_data = {}

def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
↳ **kwargs):
    ...
    persistent_data['something'] = 'some data'

# Function Two
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
↳ **kwargs):
    persistent_data = functions['Function One']['persistent_data']
    ...
```

Debugowanie twoich funkcji

You can debug the functions you create by using the standard `print()` function from Python. The output of `print()` will be displayed in a popup window after the Find/replace has completed. You saw an example of using `print()` to output an entire table of contents above.

Wybierz kolejność plików, jeśli przetwarzanych jest wiele plików HTML

When you run a *Replace all* on multiple HTML files, the order in which the files are processed depends on what files you have open for editing. You can force the search to process files in the order in which they appear by setting the `file_order` attribute on your function, like this:

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
↳ **kwargs):
    ...

replace.file_order = 'spine'
```

`file_order` może przyjąć dwie wartości, `spine` i `spine-reverse`, które powodują przeszukanie plików, odpowiednio, w kolejności w jakiej są wyświetlane lub odwrotnie.

Dodatkowe wywołanie funkcji po znalezieniu ostatniego dopasowania

Czasem, jak w przykładzie ze spisem treści powyżej, przydatne jest wywołanie funkcji dodatkowo, po ostatnim dopasowaniu. Można to zrobić, ustawiając właściwość `call_after_last_match`, o, tak:

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
↳ **kwargs):
    ...

replace.call_after_last_match = True
```

Dołączanie wyniku z funkcji do zaznaczonego tekstu

When running search and replace on marked text, it is sometimes useful to append so text to the end of the marked text. You can do that by setting the `append_final_output_to_marked` attribute on your function (note that you also need to set `call_after_last_match`), like this:

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
↳ **kwargs):
    ...
    return 'some text to append'

replace.call_after_last_match = True
replace.append_final_output_to_marked = True
```

Pomija okno dialogowe wyników podczas wyszukiwania zaznaczonego tekstu

You can also suppress the result dialog (which can slow down the repeated application of a search/replace on many blocks of text) by setting the `suppress_result_dialog` attribute on your function, like this:

```
def replace(match, number, file_name, metadata, dictionaries, data, functions, *args,
↳ **kwargs):
    ...

replace.suppress_result_dialog = True
```

Więcej przykładów

More useful examples, contributed by calibre users, can be found in the [calibre E-book editor forum](https://www.mobileread.com/forums/showthread.php?t=237181)⁴⁰.

Wstawki

Edytor e-booków Calibre obsługuje *snippets*. Snippet to fragment tekstu, który jest często używany ponownie lub zawiera dużo zbędnego tekstu. Edytor pozwala na wstawienie fragmentu za pomocą kilku naciśnień klawiszy. Załóżmy na przykład, że często wstawiasz znaczniki linków podczas edytowania plików HTML, wtedy możesz po prostu wpisać `<a` w edytorze i nacisnąć `Control+J`. Edytor rozszerzy go do:

```
<a href="filename"></a>
```

Mało tego, słowo `filename` zostanie zaznaczone, a kursor umieszczony zaraz za nim, pozwalając od razu wpisać właściwą nazwę pliku, używając bardzo wygodnej funkcji *Auto-complete* (strona 111). Kiedy wpiszesz nazwę pliku, przyciśnij ponownie `Control+J`, a kursor przeskoczy do miejsca między znacznikami `<a>`, żebyś mógł wpisać tekst odsyłacza.

System wstawek jest bardzo wyrafinowany, istnieje kilka wbudowanych wstawek, możesz również tworzyć własne, dopasowując edytor do twoich potrzeb.

Poniższe omówienie wbudowanych wstawek powinno pomóc zilustrować możliwości tego systemu.

Informacja: Możesz także użyć fragmentów tekstu w polach wprowadzania tekstu w panelu *Wyszukaj i zamień*, jednak symbole zastępcze (używając `Control+J` do przeskakiwania) nie będą działać.

⁴⁰ <https://www.mobileread.com/forums/showthread.php?t=237181>

Wbudowane wstawki

Wbudowane wstawki zostały opisane poniżej. Możesz je nadpisać własnymi, jeśli ustawisz taki sam wyzwalacz - tekst, który uruchomi wstawkę.

Wstawianie tekstu wypełniającego [Lorem]

Pierwszy wbudowany fragment i najprostszy służy do wstawiania tekstu wypełniającego do dokumentu. Tekst wypełniający pochodzi z *De finibus bonorum et malorum*⁴¹ pracy filozoficznej Cyserona (przetłumaczonej na język angielski). Aby go użyć, po prostu wpisz Lorem w pliku HTML i naciśnij Control+J. Zostanie zastąpiony kilkoma akapitami wypełniacza.

Definicja tej wstawki jest bardzo prosta, wyzwalacz to Lorem, a szablon składa się wyłącznie z tekstu, który ma zostać umieszczony w dokumencie. Można go łatwo zmienić, zastępując tekst własnym.

Wstawianie pojedynczych znaczników HTML [<>]

Przyjrzyjmy się teraz prostemu przykładowi, który pozwoli zapoznać się z koncepcją symboli zastępczych. Przypuśćmy, że chcesz wstawić pojedynczy znacznik `<hr/>`. Wpisz po prostu `<>` i przyciśnij Control+J, a edytor rozwinie wstawkę do:

```
<|/>
```

W tym przypadku symbol `|` oznacza miejsce, w którym zostanie ustawiony kursor. Możesz wpisać `hr` i wcisnąć ponownie Control+J, by przeskoczyć na koniec znacznika. Ta wstawka jest zdefiniowana następująco:

```
Trigger: <>
Template: <$1/>$2
```

Symbolle zastępcze to po prostu znak dolara, po którym następuje liczba. Kiedy wstawka jest rozwijana, po wciśnięciu Control+J, kursor jest ustawiany w miejscu pierwszego symbolu (o najmniejszym numerze). Przy kolejnym wciśnięciu Control+J kursor przeskakuje do następnego symbolu (następnego o większym numerze).

Wstawianie znacznika odsyłacza HTML [<a]

Wszystkie znaczniki odsyłaczy mają tę samą strukturę. Mają atrybut `href` i jakiś tekst między otwierającym, a zamykającym znacznikiem. Wstawka umożliwiająca wpisanie znacznika odsyłacza pokaże kolejne możliwości symboli zastępczych. Aby użyć tej wstawki po prostu wpisz `<a` i wciśnij Control+J. Edytor rozwinie wstawkę do:

```
<a href="filename|" "></a>
```

Mało tego, słowo `filename` zostanie zaznaczone, tak, że możesz łatwo wpisać prawdziwą nazwę pliku, korzystając przy okazji z użytecznej funkcji *Auto-complete* (strona 111). Kiedy wpiszesz nazwę pliku przyciśnij ponownie Control+J, a kursor przeskoczy do miejsca pomiędzy znacznikami `<a>`, byś mógł wpisać tekst odsyłacza. Następnie wciśnij Control+J jeszcze raz, by przesunąć kursor zaraz za znacznik zamykający. Wstawka jest zdefiniowana następująco:

```
Trigger: <a
Template: <a href="$1:filename">${2*}</a>$3
```

⁴¹ https://en.wikipedia.org/wiki/De_finibus_bonorum_et_malorum

Jest tu kilka nowych funkcji. Po pierwsze symbol zastępczy \$1 stał się bardziej złożony. Zawiera teraz trochę *domyślnego tekstu* (słowo filename). Jeśli symbol zastępczy zawiera tekst domyślny, tekst domyślny jest zastępowany symbolem zastępczym, gdy fragment kodu jest rozwijany. Również po przejściu do symbolu zastępczego z domyślnym tekstem za pomocą Control+J, domyślny tekst zostanie zaznaczony. W ten sposób możesz użyć domyślnego tekstu jako przypomnienia o wypełnieniu ważnych części szablonu. Możesz określić domyślny tekst dla symbolu zastępczego, używając składni: \${<number>:tekst domyślny}.

Kolejna nowość to fakt, że po drugim symbolu jest gwiazdka (\${2*}). Oznacza to, że jeśli był zaznaczony jakiś tekst przed wywołaniem wstawki to zostanie on umieszczony w miejscu tego symbolu. Aby zobaczyć jak to wygląda, zaznacz jakiś fragment tekstu w edytorze, przyciśnij Control+J, wpisz <a i przyciśnij Control+J ponownie. Szablon zostanie rozwinięty do:

```
<a href="filename">whatever text you selected</a>
```

Wstawianie znacznika obrazu [<i>]

Wstawka ta jest bardzo podobna do odsyłacza, który pokazano wyżej. Umożliwia szybkie wprowadzenie znacznika i przeskakiwanie między atrybutami src i alt:

```
Trigger: <i
Template: ${3}
```

Wstawianie znacznika HTML [<<]

Pozwala to na wstawienie dowolnego pełnego znacznika HTML (lub zawinięcie wcześniej zaznaczonego tekstu w znaczniku). Aby z niego skorzystać, po prostu wpisz ``<< «» i naciśnij ctrl. Edytor rozszerzy go do:

```
<|></>
```

Wpisz nazwę znacznika, na przykład span i przyciśnij Control+J, powinieneś w rezultacie otrzymać:

```
<span>|</span>
```

Zauważ, że zamykający znacznik automatycznie uzupełniony o span. Dzieje się tak dzięki kolejnej właściwości symboli zastępczych - *powielaniu*. Powielanie oznacza po prostu, że jeśli umieścisz symbol więcej niż raz w szablonie, drugi i kolejne automatycznie otrzymają tę samą wartość kiedy przyciśniesz Control+J. Oto definicja tej wstawki:

```
Trigger: <<
Template: <$1>${2*}</$1>${3}
```

Jak widać pierwszy symbol zastępczy (\$1) został umieszczony tu dwukrotnie, drugi raz w znaczniku zamykającym i będzie kopią tego, co wpiszesz w znaczniku otwierającym.

Wstawianie znacznika z atrybutem klasy

Przykład jest bardzo podobny do powyższego, tym razem jednak zakładamy, że chcesz podać klasę znacznika:

```
Trigger: <c
Template: <$1 class="{2:classname}">${3*}</$1>$4
```

To pozwoli ci na wpisanie nazwy znacznika, po przyciśnięciu **Control+J** - nazwy klasy, po kolejnym przyciśnięciu **Control+J** zawartości, ostatnim przyciśnięciem **Control+J** zaś zakończyć wprowadzanie. Znacznik zamykający zostanie dodany automatycznie.

Tworzenie własnych wstawek

Wstawki są naprawdę wspaniałe ponieważ możesz tworzyć swoje własne i dopasować je do twojego sposobu edytowania. Aby stworzyć własne wstawki w oknie edytora wybierz *Edycja → Ustawienia → Ustawienia edytora → Zarządzaj wstawkami*. Pojawi się okno, w którym w prosty sposób możesz tworzyć własne wstawki. Po prostu kliknij przycisk *Dodaj wstawkę*, a pojawi się okno przedstawione poniżej:

Create a snippet

For help with snippets, see the [User Manual](#)

Name:

Trigger:

Template:

File types: ☒ All ☐ css ☐ html ☐ javascript ☐ text ☐ xml

Test:

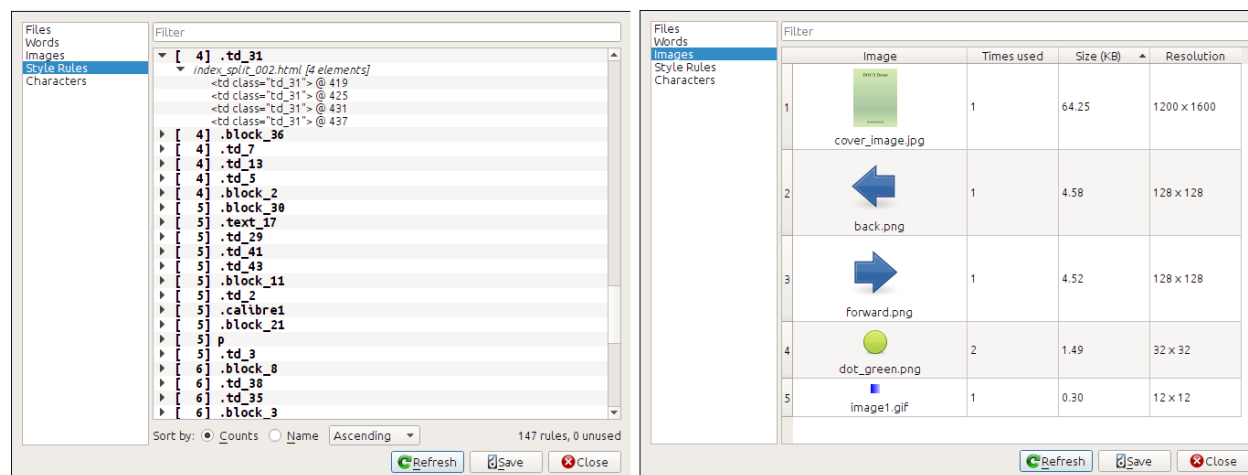
Najpierw nadaj wstawce nazwę, coś opisowego, co pozwoli zidentyfikować ją w przyszłości. Potem określ *wyzwalacz*. Wyzwalacz jest po prostu tekstem, który musisz wpisać przed przyciśnięciem **Control+J**, by rozwinąć wstawkę.

Następnie określ szablon skrawka. Powinieneś zacząć od jednego z powyższych przykładów i dostosować go do swoich potrzeb. Na koniec określ typy plików, dla których fragment ma być aktywny. W ten sposób możesz mieć wiele fragmentów kodu z tym samym tekstem wyzwalacza, które działają inaczej w różnych typach plików.

Teraz jest czas na przetestowanie nowo utworzonej wstawki. Wykorzystaj do tego pole *Test* na dole okna. Wpisz tekst wyzwalacza i użyj **Control+J**, by rozwinąć wstawkę, a potem skakać między symbolami zastępczymi.

5.8.9 The Reports tool

The editor includes a nice *Reports* tool (via *Tools* → *Reports*) that shows summaries of the files, images, links, words, characters and styles used in the book. Every line in the report is hot-linked. Double clicking a line jumps to the place in the book where that item is used or defined (as appropriate). For example, in the *Links* view, you can double click entries in the *Source* column to jump to where the link is defined and entries in the *Target* column to jump to where the link points.



5.9 Special features in the code editor

The calibre HTML editor is very powerful. It has many features that make editing of HTML (and CSS) easier.

5.9.1 Syntax highlighting

The HTML editor has very sophisticated syntax highlighting. Features include:

- The text inside bold, italic and heading tags is made bold/italic
- As you move your cursor through the HTML, the matching HTML tags are highlighted, and you can jump to the opening or closing tag with the keyboard shortcuts **Ctrl+{** and **Ctrl+}**. Similarly, you can select the contents of a tag with **Ctrl+Alt+T** or **Ctrl+Shift+T**.
- Invalid HTML is highlighted with a red underline
- Spelling errors in the text inside HTML tags and attributes such as `title` are highlighted. The spell checking is language aware, based on the value of the `lang` attribute of the current tag and the overall book language.
- CSS embedded inside `<style>` tags is highlighted
- Special characters that can be hard to distinguish such as non-breaking spaces, different types of hyphens, etc. are highlighted.
- Links to other files in `<a>` tags, `` and `<link>` tags all have the filenames highlighted. If the filename they point to does not exist, the filename is marked with a red underline.

5.9.2 Context sensitive help

You can right click on an HTML tag name or a CSS property name to get help for that tag or property.

You can also hold down the `Ctrl` key and click on any filename inside a link tag to open that file in the editor automatically. Similarly, `Ctrl` clicking a class name will take you to the first style rule that matches the tag and class.

Right clicking a class name in an HTML file will allow you to rename the class, changing all occurrences of the class throughout the book and all its stylesheets.

5.9.3 Auto-complete

When editing an e-book, one of the most tedious tasks is creating links to other files inside the book, or to CSS stylesheets, or images. You have to figure out the correct filename and relative path to the file. The editor has auto-complete to make that easier.

As you type a filename, the editor automatically pops up suggestions. Simply use the `Tab` key to select the correct file name. The editor even offers suggestions for links pointing to an anchor inside another HTML file. After you type the `#` character, the editor will show you a list of all anchors in the target file, with a small snippet of text to help you choose the right anchor.

Note that unlike most other completion systems, the editor's completion system uses subsequence matching. This means that you can type just two or three letters from anywhere in the filename to complete the filename. For example, say you want the filename `../images/arrow1.png`, you can simply type `ia1` and press `Tab` to complete the filename. When searching for matches, the completion system prioritizes letters that are at the start of a word, or immediately after a path separator. Once you get used to this system, you will find it saves you a lot of time and effort.

5.9.4 Wstawki

The calibre E-book editor supports *snippets*. A snippet is a piece of text that is either re-used often or contains a lot of redundant text. The editor allows you to insert a snippet with only a few key strokes. The snippets are very powerful, with many features, such as placeholders you can jump between, automatic mirroring of repeated text and so on. For more information, see [Wstawki](#) (strona 106).

Serwer zawartości calibre

The calibre *Content server* allows you to access your calibre libraries and read books directly in a browser on your favorite mobile phone or tablet device. As a result, you do not need to install any dedicated book reading/management apps on your phone. Just use the browser. The server downloads and stores the book you are reading in an off-line cache so that you can read it even when there is no internet connection.

Spis treści

- *Accessing the Content server from other devices* (strona 114)
 - *Accessing the server from devices on your home network* (strona 114)
 - *Accessing the server from anywhere on the internet* (strona 115)
- *The server interface* (strona 115)
 - *The book list* (strona 116)
 - *The book reader* (strona 116)
- *Browser support* (strona 116)
- *Enabling offline support* (strona 116)
- *Managing user accounts from the command-line only* (strona 117)
- *Integrating the calibre Content server into other servers* (strona 117)
 - *Using a full virtual host* (strona 117)
 - *Using a URL prefix* (strona 118)
- *Creating a service for the calibre server on a modern Linux system* (strona 119)

To start the server, click the *Connect/share* button and choose *Start Content server*. You might get a message from your computer's firewall or anti-virus program asking if it is OK to allow access to `calibre.exe`. Click the *Allow* or *OK* button. Then open a browser (preferably Chrome or Firefox) in your computer and type in the following address:

`http://127.0.0.1:8080`

This will open a page in the browser showing you your calibre libraries, click on any one and browse the books in it. Click on a book, and it will show you all the metadata about the book, along with buttons to *Read* and *Download* the book. Click the *Read* button to start reading the book.

Informacja: The address used above `http://127.0.0.1:8080` will only work on the computer that is running calibre. To access the server from other computers/phones/tablets/etc. you will need to do a little more work, as described in the next section.

6.1 Accessing the Content server from other devices

There are two types of remote device access that you will typically need. The first, simpler kind is from within your home network. If you are running calibre on a computer on your home network and you have also connected your other devices to the same home network, then you should be easily able to access the server on those devices.

6.1.1 Accessing the server from devices on your home network

After starting the server in calibre as described above, click the *Connect/share* button again. Instead of the *Start Content server* action, you should see a *Stop Content server* action instead. To the right of this action will be listed an IP address and port number. These look like a bunch of numbers separated by periods. For example:

```
Stop Content server [192.168.1.5, port 8080]
```

These numbers tell you what address to use to connect to the server in your devices. Following the example above, the address becomes:

```
http://192.168.1.5:8080
```

The first part of the address is always `http://` the next part is the IP address, which is the numbers before the comma and finally we have the port number which must be added to the IP address with a colon (:). If you are lucky, that should be all you need and you will be looking at the calibre libraries on your device. If not, read on.

Trouble-shooting the home network connection

If you are unable to access the server from your device, try the following steps:

1. Check that the server is running by opening the address `http://127.0.0.1:8080` in a browser running on the same computer as the server.
2. Check that your firewall/anti-virus is allowing connections to your computer on the port 8080 and to the calibre program. The easiest way to eliminate the firewall/anti-virus as the source of problems is to temporarily turn them both off and then try connecting. You should first disconnect from the internet, before turning off the firewall, to keep your computer safe.
3. Check that your device and computer are on the same network. This means they should both be connected to the same wireless router. In particular neither should be using a cellular or ISP provided direct-WiFi connection.
4. If you have non-standard networking setup, it might be that the IP address shown on the *Connect/share* menu is incorrect. In such a case you will have to figure out what the correct IP address to use is, yourself. Unfortunately, given the infinite diversity of network configurations possible, it is not possible to give you a roadmap for doing so.

5. If you have setup a username and password, first try it without that to see if it is causing issues. Some e-ink devices have browsers that do not handle authentication. You can sometimes workaroud this by including the username and password in the URL, for example: `http://username:password@192.168.1.2:8080`.
6. If you are stuck, you can always ask for help in the [calibre user forums](#)⁴².

6.1.2 Accessing the server from anywhere on the internet

Ostrzeżenie: Before doing this you should turn on username/password protection in the server, otherwise anyone in the world will be able to access your books. Go to *Preferences* → *Sharing* → *Sharing over the net* and enable the option to *Require username and password to access the content server*.

While the particular details on setting up internet access vary depending on the network configuration and type of computer you are using, the basic schema is as follows.

1. Find out the external IP address of the computer you are going to run the server on. You can do that by visiting the site [What is my IP address](#)⁴³ in a browser running on the computer.
2. If the computer is behind a router, enable port forwarding on the router to forward the port 8080 (or whatever port you choose to run the calibre Content server on) to the computer.
3. Make sure the calibre server is allowed through any firewalls/anti-virus programs on your computer.
4. Now you should be able to access the server on any internet-connected device using the IP address you found in the first step. For example, if the IP address you found was 123.123.123.123 and the port you are using for the calibre server is 8080, the address to use on your device becomes: `http://123.123.123.123:8080`.
5. Optionally, use a service like [no-ip](#)⁴⁴ to setup an easy to remember address to use instead of the IP address you found in the first step.

Informacja: For maximum security, you should also enable HTTPS on the Content server. You can either do so directly in the server by providing the path to the HTTPS certificate to use in the advanced configuration options for the server, or you can setup a reverse proxy as described below, to use an existing HTTPS setup.

6.2 The server interface

The server interface is a simplified version of the main calibre interface, optimised for use with touch screens. The home screen shows you books you are currently reading as well as allowing to choose a calibre library you want to browse. The server in calibre gives you access to all your libraries, not just a single one, as before.

⁴² <https://www.mobileread.com/forums/forumdisplay.php?f=166>

⁴³ <https://www.whatismyip.com/>

⁴⁴ <https://www.noip.com/free>

6.2.1 The book list

The server book list is a simple grid of covers. Tap on a cover to see the detailed metadata for a book, or to read the book. If you prefer a more detailed list, you can change the default view by clicking the three vertical dots in the top right corner.

Sorting and searching of the book list should be familiar to calibre users. They can be accessed by clicking their icons in the top right area. They both work exactly the same as in the main calibre program. The search page even allows you to construct search queries by clicking on authors/tags/etc., just as you can using the Tag browser in the main program.

A much loved feature of the main program, *Virtual libraries* is present in the server interface as well. Click the three vertical dots in the top right corner to choose a Virtual library.

6.2.2 The book reader

You can read any book in your calibre library by simply tapping on it and then tapping the *Read* button. The book reader is very simple to operate. You can both tap and swipe to turn pages. Swiping up/down skips between chapters. Tapping the top quarter of the screen gets you the detailed controls and viewer preferences.

If you leave the Content server running, you can even open the same book on multiple devices and it will remember your last read position. If it does not you can force a sync by tapping in the top quarter and choosing *Sync*.

6.3 Browser support

The new calibre server makes lots of use of advanced HTML 5 and CSS 3 features. As such it requires an up-to-date browser to use. It has been tested on Android Chrome and iOS Safari as well as Chrome and Firefox on the desktop.

The server is careful to use functionality that has either been already standardised or is on the standards track. As such if it does not currently work with your favorite browser, it probably will once that browser has caught up.

If you are using a particularly old or limited browser or you don't like to run JavaScript, you can use the *mobile* view, by simply adding `/mobile` to the server address.

Informacja: On iOS, Apple allows only a single browser engine, so Firefox, Chrome and Safari are all actually the same browser under the hood. The new server interface requires iOS 10.3.2 or newer. On Android, the server has been tested with Chrome version 58 and newer.

6.4 Enabling offline support

Browser makers have been trying to force people to use SSL by disabling advanced features in their browsers for plain HTTP connections. One such casualty is ApplicationCache, which was what was used in calibre for offline support. As a result now-a-days sadly, offline mode works only as long as you keep the browser tab open. In addition, in Firefox on Android, you will need to type `about:config` and create a preference called `browser.tabs.useCache` and set it to `true`.

6.5 Managing user accounts from the command-line only

The calibre program has a nice section in *Preferences* to allow you to manage user accounts for the server. However, if you want to run the standalone server and cannot run the main calibre program on the same computer/user account, you can also manage users using just the command-line.

You can manage user accounts using the `--manage-users` option to the standalone `calibre-server` program. Suppose you want to store the user database in the folder `/srv/calibre`, then you create it by running:

```
calibre-server --userdb /srv/calibre/users.sqlite --manage-users
```

Just follow the prompts to create user accounts, set their permission, etc. Once you are done, you can run the server as:

```
calibre-server --userdb /srv/calibre/users.sqlite --enable-auth
```

It will use the user accounts you created in the previous step.

6.6 Integrating the calibre Content server into other servers

Here, we will show you how to integrate the calibre Content server into another server. The most common reason for this is to make use of SSL or to serve the calibre library as part of a larger site. The basic technique is to run the calibre server and setup a reverse proxy to it from the main server.

A reverse proxy is when your normal server accepts incoming requests and passes them onto the calibre server. It then reads the response from the calibre server and forwards it to the client. This means that you can simply run the calibre server as normal without trying to integrate it closely with your main server.

6.6.1 Using a full virtual host

The simplest configuration is to dedicate a full virtual host to the calibre server. In this case, run the calibre server as:

```
calibre-server
```

Now setup the virtual host in your main server, for example, for nginx:

```
http {
    client_max_body_size 64M;  # needed to upload large books
}

server {
    listen [::]:80;
    server_name myserver.example.com;

    location / {
        proxy_pass http://127.0.0.1:8080;
    }
}
```

Or, for Apache:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
<VirtualHost *:80>
    ServerName myserver.example.com
    AllowEncodedSlashes On
    ProxyPreserveHost On
    ProxyPass "/" "http://localhost:8080/"
</VirtualHost>
```

6.6.2 Using a URL prefix

If you do not want to dedicate a full virtual host to calibre, you can have it use a URL prefix. Start the calibre server as:

```
calibre-server --url-prefix /calibre --port 8080
```

The key parameter here is `--url-prefix /calibre`. This causes the Content server to serve all URLs prefixed by `/calibre`. To see this in action, visit `http://localhost:8080/calibre` in your browser. You should see the normal Content server website, but now it will run under `/calibre`.

With nginx, the required configuration is:

```
http {
    client_max_body_size 64M; # needed to upload large books
}

proxy_set_header X-Forwarded-For $remote_addr;
location /calibre/ {
    proxy_buffering off;
    proxy_pass http://127.0.0.1:8080$request_uri;
}
location /calibre {
    # we need a trailing slash for the Application Cache to work
    rewrite /calibre /calibre/ permanent;
}
```

For Apache, first enable the proxy modules in Apache, by adding the following to `httpd.conf`:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
```

The exact technique for enabling the proxy modules will vary depending on your Apache installation. Once you have the proxy modules enabled, add the following rules to `httpd.conf` (or if you are using virtual hosts to the conf file for the virtual host in question):

```
AllowEncodedSlashes On
RewriteEngine on
RewriteRule ^/calibre/(.*) http://127.0.0.1:8080/calibre/$1 [proxy]
RedirectMatch permanent ^/calibre$ /calibre/
```

That's all, you will now be able to access the calibre Content server under the `/calibre` URL in your main server. The above rules pass all requests under `/calibre` to the calibre server running on port 8080 and thanks to the `--url-prefix` option above, the calibre server handles them transparently.

Informacja: When using a reverse proxy, you should tell the calibre Content server to only listen on localhost, by using `--listen-on 127.0.0.1`. That way, the server will only listen for connections coming from the same computer, i.e. from the reverse proxy.

Informacja: If you have setup SSL for your main server, you should tell the calibre server to use basic authentication instead of digest authentication, as it is faster. To do so, pass the `--auth-mode=basic` option to `calibre-server`.

6.7 Creating a service for the calibre server on a modern Linux system

You can easily create a service to run calibre at boot on a modern ([systemd](https://www.freedesktop.org/wiki/Software/systemd/)⁴⁵) based Linux system. Just create the file `/etc/systemd/system/calibre-server.service` with the contents shown below:

```
[Unit]
Description=calibre Content server
After=network.target

[Service]
Type=simple
User=mylinuxuser
Group=mylinuxgroup
ExecStart=/opt/calibre/calibre-server "/path/to/calibre library folder"

[Install]
WantedBy=multi-user.target
```

Change `mylinuxuser` and `mylinuxgroup` to whatever user and group you want the server to run as. This should be the same user and group that own the files in the calibre library folder. Note that it is generally not a good idea to run the server as root. Also change the path to the calibre library folder to suit your system. You can add multiple libraries if needed. See the help for the `calibre-server` command.

Now run:

```
sudo systemctl start calibre-server
```

to start the server. Check its status with:

```
sudo systemctl status calibre-server
```

To make it start at boot, run:

```
sudo systemctl enable calibre-server
```

Informacja: The calibre server *does not* need a running X server, but it does need the X libraries installed as some components it uses link against them.

Informacja: The calibre server also supports systemd socket activation, so you can use that, if needed, as well.

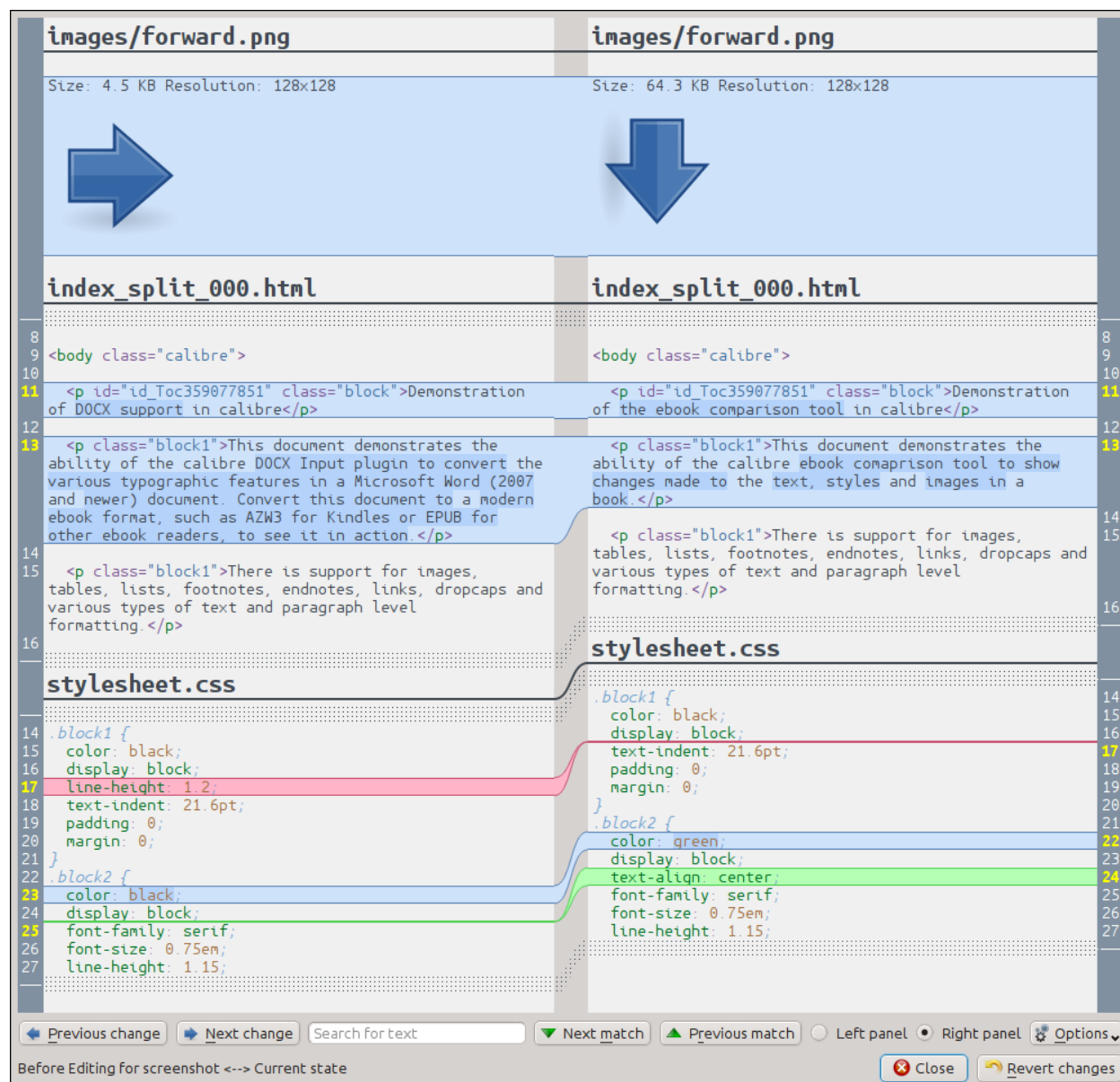
⁴⁵ <https://www.freedesktop.org/wiki/Software/systemd/>

Porównanie e-booków

calibre zawiera zintegrowane narzędzie do porównywania e-booków, za pomocą którego można zobaczyć, co się zmieniło w e-booku po jego edycji lub konwersji. Może porównywać książki w formatach EPUB i AZW3.

Aby z niego skorzystać, otwórz e-booka w narzędziu dla *Edycja e-booków* (strona 79) a następnie kliknij *Plik → Porównaj z inną książką* lub użyj panelu *Szczegóły książki* (strona 20). Jeśli wykonasz konwersję z EPUB na EPUB, oryginalny plik EPUB zostanie zapisany jako ORIGINAL_EPUB. Po prostu kliknij prawym przyciskiem myszy wpis ORIGINAL_EPUB w panelu szczegółów książki i wybierz *Porównaj z formatem EPUB*.

Okno porównywania wygląda jak pokazano poniżej. Widoczne są w nim różnice w tekście, stylach i grafikach w wybranych książkach.



7.1 Jak działa widok porównania

Jak widać na rzucie ekranu powyżej różnice pokazane są obok siebie. Są to tylko różnice oraz po kilka wierszy dookoła nich. Dzięki temu łatwo jest się zorientować co zmieniło się w dużym dokumencie, takim jak książka.

Dodany tekst jest podświetlony na zielono, usunięty na czerwono, a zmieniony na niebiesko.

Numery zmienionych wierszy są wyświetlane na marginesach, ułatwiając przejście do właściwego miejsca w edytorze. Jeśli porównywanie zostało wywołane z edytora można również kliknąć dwukrotnie linię w prawym panelu, by przejść do niej w edytorze.

Dobrze jest uporządkować wygląd plików przed porównaniem książek. Dzięki temu wynik będzie bardziej przejrzysty i łatwiej będzie śledzić różnice. Aby to zrobić, kliknij przycisk *Opcje* w prawym dolnym rogu okna i wybierz **Popraw wygląd plików przed porównaniem**. Czasem mogą pojawić się nieoczekiwane efekty ponieważ

w trakcie procesu błędne znaczniki mogą zostać poprawione na właściwe. Można również zmienić liczbę wierszy kontekstu, pojawiającego się przed i po zmienionym wierszu, klikając *Opcje*.

Możesz wyszukać dowolny tekst w różnicach za pomocą paska wyszukiwania na dole. Będziesz musiał określić, który panel przeszukać *Left* or the *Right*.

7.2 Uruchamianie narzędzia porównywania

Porównywanie przydaje się najbardziej, gdy masz dwie wersje książki i chcesz zobaczyć czym się różnią. Jest wiele sposobów na uruchomienie tego narzędzia.

7.2.1 Porównanie dwóch plików e-booków

Otwórz pierwszy plik w oknie *Edycja e-booków* (strona 79). Teraz kliknij *Plik* → *Porównaj z inną książką* i wybierz drugi plik (musi być w takim samym formacie jak pierwszy). Porównanie zostanie pokazane w nowym oknie, w którym tekst aktualnie edytowany jest po prawej, a porównywany z nim po lewej stronie.

7.2.2 Porównywanie ORIGINAL_FMT z FMT

Kiedy konwertujesz książkę w calibre z formatu FMT do tego samego formatu, oryginalny plik jest zapisywany jako ORIGINAL_FMT. Możesz sprawdzić jakie zmiany zaszły w trakcie konwersji, klikając w ORIGINAL_FMT w panelu *Szczegóły książki* (strona 20) w głównym oknie calibre i wybierając z menu *Porównaj z FMT*. Porównanie zostanie otwarte w nowym oknie z plikiem ORIGINAL_FMT po lewej i FMT po prawej stronie.

7.2.3 Porównywanie punktu kontrolnego do aktualnego stanu edytowanej książki

Edycja e-booków (strona 79) ma bardzo użyteczną funkcję, nazwaną *Punkty kontrolne* (strona 91). Pozwala ona na zapisanie bieżącego stanu książki jako nazwanego *punktu kontrolnego*, do którego możesz wrócić, jeśli nie satysfakcjonują cię zmiany dokonane po jego utworzeniu. Punkty kontrolne są również tworzone automatycznie po wykonaniu niektórych czynności w edytorze. Listę punktów kontrolnych możesz sprawdzić, zaznaczając opcję *View* → *Punkty kontrolne*, a potem używając przycisku *Porównaj*, by sprawdzić różnice między punktem kontrolnym, a stanem bieżącym książki. W oknie porównania punkt kontrolny znajdzie się po lewej, a bieżący stan po prawej stronie.

Edycja metadanych e-booków

Spis treści

- *Edytowanie metadanych jednej książki* (strona 125)
 - *Pobieranie metadanych* (strona 126)
 - *Zarządzanie formatami książek* (strona 126)
 - *Wszystko o okładkach* (strona 126)
- *Edycja metadanych wielu książek jednocześnie* (strona 126)
 - *Znajdź i zastąp* (strona 127)
 - *Masowe pobieranie metadanych* (strona 128)
- *Adding extra data files to a book* (strona 128)

E-books come in all shapes and sizes and more often than not, their metadata (things like title/author/series/publisher) is incomplete or incorrect. The simplest way to change metadata in calibre is to simply double click on an entry and type in the correct replacement. For more sophisticated, „power editing” use the edit metadata tools discussed below.

8.1 Edytowanie metadanych jednej książki

Kliknij książkę, którą chcesz edytować, a następnie kliknij *Edytuj metadane* albo przyciśnij klawisz E. Pojawi się okno, w którym można edytować wszystkie metadane. Dzięki różnym usprawnieniom edycję można przeprowadzić szybciej i sprawniej. Oto kilka najbardziej przydatnych wskazówek:

- Kliknięcie przycisku pomiędzy autorem i tytułem automatycznie zamieni miejscami wartości
- Kliknięcie przycisku tuż za polem sortowania autorów spowoduje wypełnienie tego pola dla każdego autora. Możesz użyć okna *Zarządzaj autorami*, by przejrzeć i poprawić dane autorów. Okno to można wywołać, klikając i przytrzymując przycisk obok pola sortowania autorów.

- You can click the button next to tags to use the *Tag editor* to manage the tags associated with the book.
- The „Ids” box can be used to enter an ISBN (and many other types of id), it will have a red background if you enter an invalid ISBN. It will be green for valid ISBNs.
- Pole sortowania wg autorów będzie miało czerwone tło jeśli jego wartość nie odpowiada temu, co calibre wylicza z wartości pola autora.

8.1.1 Pobieranie metadanych

The nicest feature of the edit metadata dialog is its ability to automatically fill in many metadata fields by getting metadata from various websites. Currently, calibre uses Google Books and Amazon. The metadata download can fill in Title, author, series, tags, rating, description and ISBN for you.

Aby pobrać dane, podaj autora i tytuł, a następnie kliknij przycisk *Pobierz metadane*. W oknie, które się pojawi, zobaczysz listę książek najlepiej pasujących do wpisanych przez ciebie danych. Jeśli podałeś również ISBN będzie on miał pierwszeństwo przed innymi danymi. Jeśli calibre nie znajdzie żadnych książek spróbuj podać nieco mniej danych - tylko niektóre słowa w tytule, tylko nazwisko autora.

8.1.2 Zarządzanie formatami książek

W bibliotece calibre każda z książek może mieć wiele przypisanych do niej *formatów*. Jeśli na przykład masz „Dzieła zebrane” Szekspira w formacie EPUB, to, by przeczytać je na swoim Kindle, musisz przekonwertować je na MOBI. calibre automatycznie zarządza formatami twoich książek. W oknie edycji metadanych, w sekcji *Dostępne formaty* możesz zarządzać nimi samodzielnie. Formaty mogą być dodawane, usuwane, możesz również przypisać książce metadane i okładkę jednego z formatów.

8.1.3 Wszystko o okładkach

calibre może pobrać okładkę książki z internetu jeśli znany jest jej numer ISBN. Możesz również sam przypisać okładkę, wybierając plik z grafiką. calibre jest w stanie wygenerować domyślną okładkę, zawierającą podstawowe metadane. Możesz przeciągnąć obraz na okładkę, by ją zmienić, albo kliknąć prawym klawiszem i wkleić obraz ze schowka.

Jest również przycisk, którego kliknięcie automatycznie obetnie krawędzie okładki jeśli obraz ma brzydkie brzegi.

8.2 Edycja metadanych wielu książek jednocześnie

First select the books you want to edit by holding **Ctrl** or **Shift** and clicking on them. If you select more than one book, clicking the *Edit metadata* button will cause the *Bulk* metadata edit dialog to open. Using this dialog, you can quickly set the author/publisher/rating/tags/series etc of a bunch of books to the same value. This is particularly useful if you have just imported a number of books that have some metadata in common. This dialog is very powerful, for example, it has a *Search and replace* tab that you can use to perform bulk operations on metadata and even copy metadata from one column to another.

The normal edit metadata dialog also has *Next* and *Previous* buttons that you can use to edit the metadata of several books one after the other.

8.2.1 Znajdź i zastąp

The *Edit metadata for many books* dialog allows you to perform arbitrarily powerful search and replace operations on the selected books. By default it uses a simple text search and replace, but it also support *regular expressions*. For more on regular expressions, see *Wszystko, co chcielibyście wiedzieć o wyrażeniach regularnych w calibre* (strona 213).

As noted above, there are two search and replace modes: character match and regular expression. Character match will look in the *Search field* you choose for the characters you type in the *search for* box and replace those characters with what you type in the *replace with* box. Each occurrence of the search characters in the field will be replaced. For example, assume the field being searched contains *a bad cat*. If you search for *a* to be replaced with *HELLO*, then the result will be *HELLO bHELLOd cHELLOt*.

Jeśli przeszukiwane pole jest typu *wiele wartości*, jak etykiety, jest ono przeszukiwane nieco inaczej. Na przykład jeśli pole etykiet zawiera *Horror, Straszny*, wyszukanie *r*, (z przecinkiem po *r*) nie zwróci żadnego wyniku, ponieważ podany tekst jest wyszukiwany najpierw w *Horror*, a następnie w *Straszny*.

Jeśli nie chcesz, by przy szukaniu była uwzględniana wielkość liter, odznacz pole *Uwzględniaj wielkość liter*.

Możesz również zmienić wielkość liter w wyniku wyszukiwania (już po zamianie tekstów), wybierając jedną z funkcji, dostępnych na liście *Zastosuj funkcję po zamianie*. Dostępne opcje to:

- *Małe litery* – zmień wszystkie znaki w polu na małe litery
- *Wielkie litery* – zmień wszystkie znaki w polu na wielkie litery
- *Wielkość liter jak w tytule* – Każda pierwsza litera wyrazu wielka (amerykańska zasada - przyp. tłum)

Pole *Wynik testu* służy sprawdzeniu, czy efekt jest taki, jakiego się spodziewałeś. Zazwyczaj pola testowe, zawierające dane z książek, są wystarczające, ale może się zdarzyć, że będziesz chciał sprawdzić coś, co nie pojawia się w tych polach. W takim przypadku wpisz tekst w polu *Wynik testu*.

Tryb wyrażen regularnych różni się nieco od zwykłego poprzez (oczywiście) wykorzystanie wyrażen regularnych. Pierwsza różnica jest taka, że zmiany dotyczą wyłącznie części, które pasują do tych części pola, które pasują do podanego wzorca wyszukiwania, a nie do całego pola. Druga - funkcje zostaną zastosowane do zamienianego tekstu, a nie do całego pola.

Trzecia i najważniejsza różnica jest taka, że można stosować odniesienia do części wyszukiwanego tekstu za pomocą odwołań wstecznych. Odwołanie wsteczne to $\backslash n$, gdzie *n* jest liczbą całkowitą, oznaczającą *n*-tą grupę, ujętą w nawiasy we wzorcu wyszukiwania. W użytym powyżej przykładzie, *katamaran*, wzorec wyszukiwania *a(...)(...)* razem z wzorcem do zamiany $\backslash 2\backslash 1$, da w wyniku zamiany *kara tamn*. Jeśli chcesz się dowiedzieć więcej na ten temat, zajrzyj do rozdziału *Wszystko, co chcielibyście wiedzieć o wyrażeniach regularnych w calibre* (strona 213).

One useful pattern: assume you want to change the case of an entire field. The easiest way to do this is to use character mode, but lets further assume you want to use regular expression mode. The search expression should be $(^.*\$)$, the replace expression should be $\backslash 1$, and the desired case change function should be selected.

W trybie wyrażen regularnych możesz również kopiować dane z jednego pola do innego. Po prostu wybierz różne pola wyszukiwania i docelowe. Kopiowany tekst może zastąpić tekst w polu docelowym albo zostać doklejony na początku lub na końcu. Pole *Podziel wynik* decyduje o tym, czy między kopiowanym tekstem, a dotychczasową wartością pola jest wstawiany przecinek, czy nie w trybach doklejania tekstu. Jeśli pole docelowe jest polem wielowartościowym ta opcja nie działa.

Wyszukiwanie i zamiana są dokonywane, kiedy już operacje na innych zakładkach zostały zakończone. Może to prowadzić do pewnego zamieszania, bo pola pokazują wartości pól sprzed zmian, a zamiana będzie wykonywana już po ich przetworzeniu. Jeśli nie jesteś pewien jaki będzie połączony efekt, postaraj się nie łączyć operacji na różnych zakładkach.

8.2.2 Masowe pobieranie metadanych

Jeśli chcesz pobrać metadane dla wielu książek jednocześnie, kliknij prawym klawiszem myszy przycisk *Edytuj metadane* i wybierz opcję *Pobierz metadane i okładki*. Możesz zdecydować, czy chcesz pobrać tylko metadane, tylko okładki, czy wszystko naraz.

8.3 Adding extra data files to a book

calibre can store any number of extra data files associated to a book. These can be alternate covers, supplementary material, etc. They cannot be viewed directly or used as conversion sources. Nor are they indexed by the Full text search engine in calibre. To view/add/delete them select the book and right click the *Edit metadata* button and choose *Manage data files*. This will pop-up a window where you can perform operations on these files. Alternately, you can right click the *Add books* button and choose *Add data files to selected book records* to more quickly add data files.

Często zadawane pytania

Spis treści

- *Konwersja formatu e-booka* (strona 129)
- *Integracja urządzenia* (strona 132)
- *Library management* (strona 139)
- *Różne* (strona 144)

9.1 Konwersja formatu e-booka

Spis treści

- *Jakie formaty może konwertować calibre?* (strona 130)
- *Jakie formaty najlepiej nadają się do konwersji?* (strona 130)
- *Przekonwertowałem plik PDF, ale wynik nie jest zadowalający.* (strona 130)
- *Jak przekonwertować pliki, zawierające znaki diakrytyczne lub cudzysłowy typograficzne?* (strona 130)
- *O co chodzi ze spisem treści w formacie MOBI?* (strona 131)
- *Jak mogę przekonwertować zestaw plików HTML w określonej kolejności?* (strona 131)
- *EPUB, który wyprodukowałem z calibre, jest nieprawidłowy?* (strona 132)
- *Jak korzystać z niektórych zaawansowanych funkcji narzędzi do konwersji?* (strona 132)

9.1.1 Jakie formaty może konwertować calibre?

Calibre umożliwia konwersję z i na wiele różnych formatów. Można przekonwertować każdy format wejściowy z poniższej listy na każdy format wyjściowy.

Formaty wejściowe: AZW, AZW3, AZW4, CBZ, CBR, CB7, CBC, CHM, DJVU, DOCX, EPUB, FB2, FBZ, HTML, HTMLZ, LIT, LRF, MOBI, ODT, PDF, PRC, PDB, PML, RB, RTF, SNB, TCR, TXT, TXTZ

Formaty wyjściowe: AZW3, EPUB, DOCX, FB2, HTMLZ, OEB, LIT, LRF, MOBI, PDB, PMLZ, RB, PDF, RTF, SNB, TCR, TXT, TXTZ, ZIP

Informacja: PRC to ogólny format, calibre obsługuje pliki PRC z nagłówkami TextRead i MOBIBook. PDB jest również formatem ogólnym. calibre obsługuje eReader, Plucker (tylko dane wejściowe), PML i zTxB. Obsługa DJVU służy tylko do konwersji plików DJVU zawierających osadzony tekst. Są one zazwyczaj generowane przez oprogramowanie OCR. Książki MOBI mogą mieć dwa typy Mobi6 i KF8. Calibre w pełni obsługuje oba. Pliki MOBI często mają rozszerzenia .azw lub .azw3. Obsługiwane są pliki DOCX z programu Microsoft Word 2007 i nowszego.

9.1.2 Jakie formaty najlepiej nadają się do konwersji?

W kolejności malejącej preferencji: LIT, MOBI, AZW, EPUB, AZW3, FB2, FBZ, DOCX, HTML, PRC, ODT, RTF, PDB, TXT, PDF

9.1.3 Przekonwertowałem plik PDF, ale wynik nie jest zadowalający.

PDF jest okropnym formatem do konwersji. Wiele problemów, które możesz napotkać przy konwersji PDF jest opisanych tu: [Convert PDF documents](#) (strona 73).

9.1.4 Jak przekonwertować pliki, zawierające znaki diakrytyczne lub cudzośliwy typograficzne?

Ten problem ma dwa aspekty:

1. Znając kodowanie pliku źródłowego: calibre próbuje odgadnąć, jakiego kodowania znaków używają twoje pliki źródłowe, ale często jest to niemożliwe, więc musisz powiedzieć, jakiego kodowania użyć. Można to zrobić w GUI za pomocą pola *Wprowadź kodowanie znaków* w sekcji *Wygląd i styl* → *Tekst* okna dialogowego konwersji. Narzędzia wiersza poleceń mają opcję `ebook-convert-txt-input --input-encoding`.
2. Podczas dodawania plików HTML do calibre może być konieczne podanie calibre, w jakim kodowaniu znajdują się pliki. Aby to zrobić, przejdź do *Preferencje* → *Zaawansowane* → *Wtyczki* → *Typ pliku* i dostosuj *HTML do Wtyczka ZIP*, informująca, w jakim kodowaniu znajdują się Twoje pliki HTML. Teraz, gdy dodasz pliki HTML do calibre, zostaną one poprawnie przetworzone. Pliki HTML z różnych źródeł często mają różne kodowanie, więc może być konieczne wielokrotne zmienianie tego ustawienia. Powszechnym kodowaniem dla wielu plików z sieci jest cp1252 i sugeruję, abyś najpierw spróbował tego. Pamiętaj, że podczas konwertowania plików HTML pozostaw powyższe ustawienie kodowania wejściowego puste. Dzieje się tak, ponieważ wtyczka *HTML to ZIP* automatycznie konwertuje pliki HTML do standardowego kodowania (UTF-8).

9.1.5 O co chodzi ze spisem treści w formacie MOBI?

Pierwszą rzeczą, którą należy sobie uświadomić, jest to, że większość e-booków ma dwa spisy treści. Jednym z nich jest tradycyjny spis treści, taki jak spis treści, który można znaleźć w papierowych książkach. Ten spis treści jest częścią głównego obiegu dokumentów i można go dowolnie stylizować. Ten spis treści nazywa się *spisem treści*.

Następnie jest *Spis treści metadanych*. Spis treści metadanych to spis treści, który nie jest częścią tekstu książki i jest zwykle otwierany za pomocą specjalnego przycisku na czytniku. Na przykład w przeglądarce e-booków Calibre używasz przycisku Pokaż spis treści, aby wyświetlić ten spis treści. Ten spis treści nie może być stylizowany przez twórcę książki. Sposób przedstawienia zależy od programu widza.

In the MOBI format, the situation is a little confused. This is because the MOBI format, alone amongst mainstream e-book formats, *does not* have decent support for a metadata ToC. A MOBI book simulates the presence of a metadata ToC by putting an *extra* content ToC at the end of the book. When you click Goto Table of Contents on your Kindle, it is to this extra content ToC that the Kindle takes you.

Now it might well seem to you that the MOBI book has two identical ToCs. Remember that one is semantically a content ToC and the other is a metadata ToC, even though both might have exactly the same entries and look the same. One can be accessed directly from the Kindle's menus, the other cannot.

When converting to MOBI, calibre detects the *metadata ToC* in the input document and generates an end-of-file ToC in the output MOBI file. You can turn this off by an option in the MOBI Output settings. You can also tell calibre whether to put it at the start or the end of the book via an option in the MOBI Output settings. Remember this ToC is semantically a *metadata ToC*, in any format other than MOBI it *cannot not be part of the text*. The fact that it is part of the text in MOBI is an accident caused by the limitations of MOBI. If you want a ToC at a particular location in your document text, create one by hand. So we strongly recommend that you leave the default as it is, i.e. with the metadata ToC at the end of the book. Also note that if you disable the generation of the end-of-file ToC the resulting MOBI file may not function correctly on a Kindle, since the Kindle's use the metadata ToC for many things, including the Page Flip feature.

If you have a hand edited ToC in the input document, you can use the ToC detection options in calibre to automatically generate the metadata ToC from it. See the conversion section of the User Manual for more details on how to use these options.

Finally, I encourage you to ditch the content ToC and only have a metadata ToC in your e-books. Metadata ToCs will give the people reading your e-books a much superior navigation experience (except on the Kindle, where they are essentially the same as a content ToC).

Informacja: The newer AZW3 format has proper support for a metadata ToC. However, the Kindle firmware tends to malfunction if you disable the generation of the end-of-file inline ToC. So it is recommended that you leave the generated ToC alone. If you create an AZW3 file with a metadata ToC and no end-of-file generated ToC, some features on the Kindle will not work, such as the Page Flip feature.

9.1.6 Jak mogę przekonwertować zestaw plików HTML w określonej kolejności?

By przeprowadzić konwersję zestawu plików HTML w określonej kolejności, musisz stworzyć plik spisu treści. Tak, kolejny plik HTML, zawierający odsyłacze do wszystkich innych plików, we właściwej kolejności. Taki plik wygląda następująco:

```
<html>
  <body>
    <h1>Table of Contents</h1>
    <p style="text-indent:0pt">
      <a href="file1.html">First File</a><br/>
      <a href="file2.html">Second File</a><br/>
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

.
.
.
</p>
</body>
</html>

```

Then, just add this HTML file to the GUI and use the *Convert* button to create your e-book. You can use the option in the Table of Contents section in the conversion dialog to control how the Table of Contents is generated.

Informacja: By default, when adding HTML files, calibre follows links in the files in *depth first* order. This means that if file A.html links to B.html and C.html and D.html, but B.html also links to D.html, then the files will be in the order A.html, B.html, D.html, C.html. If instead you want the order to be A.html, B.html, C.html, D.html then you must tell calibre to add your files in *breadth first* order. Do this by going to *Preferences* → *Advanced* → *Plugins* → *File type* and customizing the *HTML to ZIP* plugin.

9.1.7 EPUB, który wyprodukowałem z calibre, jest nieprawidłowy?

calibre does not guarantee that an EPUB produced by it is valid. The only guarantee it makes is that if you feed it valid XHTML 1.1 + CSS 2.1 it will output a valid EPUB. calibre tries hard to ensure that EPUBs it produces actually work as intended on a wide variety of devices, a goal that is incompatible with producing valid EPUBs, and one that is far more important to the vast majority of its users. If you need a tool that always produces valid EPUBs, calibre is not for you. This means, that if you want to send a calibre produced EPUB to an online store that uses an EPUB validity checker, you have to make sure that the EPUB is valid yourself, calibre will not do it for you – in other words you must feed calibre valid XHTML + CSS as the input documents.

9.1.8 Jak korzystać z niektórych zaawansowanych funkcji narzędzi do konwersji?

You can get help on any individual feature of the converters by mousing over it in the GUI or running `ebook-convert dummy.html .epub -h` at a terminal. A good place to start is to look at the following demo file that demonstrates some of the advanced features [html-demo.zip](https://calibre-ebook.com/downloads/html-demo.zip)⁴⁶.

9.2 Integracja urządzenia

Spis treści

- *Jakie urządzenia obsługuje calibre?* (strona 133)
- *Jak mogę pomóc w uzyskaniu obsługi urządzenia w calibre?* (strona 133)
- *Moje urządzenie nie jest wykrywane przez calibre?* (strona 134)
- *Moje urządzenie jest niestandardowe lub nietypowe. Co mogę zrobić, aby się z nim połączyć?* (strona 134)
- *Jak używać calibre na moim iPadzie/iPhonie/iPod touch?* (strona 134)
- *How do I use calibre with my Android phone/tablet or Kindle Fire?* (strona 135)

⁴⁶ <https://calibre-ebook.com/downloads/html-demo.zip>

- *Can I access my calibre books using the web browser in my Kindle or other reading device?* (strona 136)
- *Nie mogę wysłać e-maili przy użyciu calibre?* (strona 136)
- *Moje urządzenie jest montowane tylko do odczytu w systemie Linux, więc Calibre nie może się z nim połączyć?* (strona 137)
- *Dlaczego calibre nie obsługuje kolekcji na Kindle lub półkach w Nook?* (strona 137)
- *Otrzymuję błąd, gdy próbuję użyć calibre z moim Kobo Touch/Glo/td.*? (strona 138)
- *Okladki książek, które wysyłam do mojego e-ink Kindle, pojawiają się natychmiast, a następnie są zastępowane przez okładkę ogólną?* (strona 138)
- *The covers for my MOBI files have stopped showing up in Kindle for PC/Kindle for Android/iPad etc.* (strona 139)
- *I transferred some books to my Kindle using calibre and they did not show up?* (strona 139)

9.2.1 Jakie urządzenia obsługuje calibre?

calibre can directly connect to all the major (and most of the minor) e-book reading devices, smartphones, tablets, etc. In addition, using the *Connect to folder* function you can use it with any e-book reader that exports itself as a USB disk. Finally, you can connect wirelessly to any device that has a web browser using the calibre Content server.

9.2.2 Jak mogę pomóc w uzyskaniu obsługi urządzenia w calibre?

If your device appears as a USB disk to the operating system, adding support for it to calibre is very easy. We just need some information from you:

- Pełna lista formatów e-booków obsługiwanych przez Twoje urządzenie.
- Czy na urządzeniu jest specjalny folder, w którym należy umieścić wszystkie pliki e-booków? Czy urządzenie wykrywa również pliki umieszczone w podfolderach?
- Potrzebujemy również informacji o Twoim urządzeniu, które calibre będzie zbierać automatycznie. Po pierwsze, jeśli Twoje urządzenie obsługuje karty SD, włóż je. Następnie podłącz urządzenie do komputera. W calibre przejdź do *Preferencje* → *Różne* i kliknij przycisk „Debuguj wykrywanie urządzeń”. Spowoduje to utworzenie danych wyjściowych debugowania. Skopiuj go do pliku i powtórz proces, tym razem z odłączonym urządzeniem od komputera.
- Wyślij do nas oba powyższe dane wyjściowe wraz z innymi informacjami, a my napiszemy sterownik urządzenia dla Twojego urządzenia.

Once you send us the output for a particular operating system, support for the device in that operating system will appear in the next release of calibre. To send us the output, open a bug report and attach the output to it. See [how to report bugs](#)⁴⁷.

⁴⁷ <https://calibre-ebook.com/pl/bugs>

9.2.3 Moje urządzenie nie jest wykrywane przez calibre?

Wykonaj następujące kroki, aby znaleźć problem:

- Upewnij się, że jednocześnie łączysz tylko jedno urządzenie z komputerem. Nie używaj jednocześnie innego obsługiwanego urządzenia, takiego jak iPhone/iPad itp.
- Jeśli podłączasz urządzenie Apple iDevice (iPad, iPod Touch, iPhone), Apple nie zezwala już oprogramowaniu innych firm na łączenie się z ich urządzeniami za pomocą kabla USB. Zamiast tego użyj połączenia bezprzewodowego za pośrednictwem serwera zawartości calibre.
- Jeśli podłączasz Kindle Fire lub inne urządzenie z Androidem, przeczytaj notatkę pod *Korzystanie z kabla USB* (strona 135).
- W systemie macOS, jeśli podczas podłączania urządzenia do calibre pojawiają się błędy uprawnień, możesz to naprawić, zaglądając do: *guilabel: Preferencje systemowe > Bezpieczeństwo i prywatność > Prywatność > Pliki i foldery*.
- Make sure you are running the latest version of calibre (currently 7.9.0). The latest version can always be downloaded from [the calibre website](https://calibre-ebook.com/pl/download)⁴⁸. You can tell what version of calibre you are currently running by looking at the bottom line of the main calibre window.
- Upewnij się, że Twój system operacyjny widzi urządzenie. Oznacza to, że urządzenie powinno pojawić się w Eksploratorze Windows (w systemie Windows) lub Finderze (w systemie macOS).
- W calibre przejdź do *Preferencje → Ignorowane urządzenia* i sprawdź, czy Twoje urządzenie nie jest ignorowane
- Jeśli wszystkie powyższe kroki nie powiodą się, przejdź do *Preferencje → Różne* i kliknij *Debugowanie wykrywania urządzenia* z podłączonym urządzeniem i opublikuj dane wyjściowe jako zgłoszenie w [calibre bug tracker](https://bugs.launchpad.net/calibre)⁴⁹.

9.2.4 Moje urządzenie jest niestandardowe lub nietypowe. Co mogę zrobić, aby się z nim połączyć?

In addition to the *Connect to folder* function found under the *Connect/share* button, calibre provides a User defined device plugin that can be used to connect to any USB device that shows up as a disk drive in your operating system. Note: on Windows, the device must have a drive letter for calibre to use it. See the device plugin Preferences -> Plugins -> Device plugins -> User defined and Preferences -> Miscellaneous -> Get information to setup the user defined device for more information. Note that if you are using the user defined plugin for a device normally detected by a builtin calibre plugin, you must disable the builtin plugin first, so that your user defined plugin is used instead.

9.2.5 Jak używać calibre na moim iPadzie/iPhonie/iPod touch?

An easy way to browse your calibre collection from your Apple device is by using *Serwer zawartości calibre* (strona 113), which makes your collection available over the net. First perform the following steps in calibre

- Set the Preferred Output Format in calibre to EPUB (The output format can be set under *Preferences → Interface → Behavior*)
- Set the output profile to iPad (this will work for iPhone/iPods as well), under *Preferences → Conversion → Common options → Page setup*
- Convert the books you want to read on your iDevice to EPUB format by selecting them and clicking the *Convert* button.

⁴⁸ <https://calibre-ebook.com/pl/download>

⁴⁹ <https://bugs.launchpad.net/calibre>

- Turn on the Content server by clicking the *Connect/share* button and leave calibre running. You can also tell calibre to automatically start the Content server via *Preferences* → *Sharing* → *Sharing over the net*.

The Content server allows you to read books directly in Safari itself. In addition, there are many apps for your iDevice that can connect to the calibre Content server. Examples include: Marvin, Mapleread and iBooks itself.

Korzystanie z serwera treści

Uruchom przeglądarkę Safari i wpisz adres IP oraz port komputera, na którym działa serwer calibre, w następujący sposób:

```
http://192.168.1.2:8080/
```

Zastąp 192.168.1.2 lokalnym adresem IP komputera z uruchomionym calibre. Zobacz *Serwer zawartości calibre* (strona 113), aby uzyskać szczegółowe informacje o uruchomieniu serwera i znalezieniu właściwego adresu IP.

Zobaczysz listę książek w Safari, dotknij dowolnej książki, a będziesz mieć możliwość jej pobrania lub przeczytania w samej przeglądarce. Jeśli zdecydujesz się go pobrać, Safari zapyta, czy chcesz go otworzyć w iBooks.

Many reading apps support browsing the calibre library directly via its **OPDS** support. In such apps you can go to the online catalog screen and add the IP address of the calibre server to browse and download books from your calibre library within the app.

9.2.6 How do I use calibre with my Android phone/tablet or Kindle Fire?

There are two ways that you can connect your Android device to calibre. Using a USB cable – or wirelessly, over the air. The first step to using an Android device is installing an e-book reading application on it. There are many free and paid e-book reading applications for Android: Some examples (in no particular order): [FBReader](#)⁵⁰, [Moon+](#)⁵¹, [Mantano](#)⁵², [Aldiko](#)⁵³, [Kindle](#)⁵⁴.

Korzystanie z kabla USB

Simply plug your device into the computer with a USB cable. calibre should automatically detect the device and then you can transfer books to it by clicking the *Send to device* button. Note that on macOS and Linux only a single program can connect to an Android device at a time, so make sure the device is not opened in the OS File manager, or the Android File Transfer utility, etc.

Informacja: With newer Android devices, you might have to jump through a few hoops to get the connection working, as Google really does not want you to be independent of its cloud. First, unlock the screen before plugging in the USB cable. When you plugin in the USB cable you will get a popup notification. Make sure it says some thing like „Transferring Media files” or „MTP (Media Transfer mode)”. If it does not, tap the notification, and change the mode to Media Transfer (MTP). You may need to restart calibre at this point in order for your device to be recognized. Finally, you might get a popup on the device every time calibre or the operating system actually tries to connect to it, asking for permission, tap OK.

Informacja: With the Kindle Fire 8 or newer there is an icon that shows up when the USB cable is plugged in, showing that the device is charging. Tap that and switch the device to data transfer mode, and then start calibre, it should then be

⁵⁰ <https://play.google.com/store/apps/details?id=org.geometerplus.zlibrary.ui.android&hl=en>

⁵¹ <https://play.google.com/store/apps/details?id=com.flyersoft.moonreader&hl=en>

⁵² <https://play.google.com/store/apps/details?id=com.mantano.reader.android.lite&hl=en>

⁵³ <https://play.google.com/store/apps/details?id=com.aldiko.android&hl=en>

⁵⁴ https://play.google.com/store/apps/details?id=com.amazon.kindle&feature=related_apps

detected.

Bezprzewodowo

calibre has a builtin web server, the *Serwer zawartości calibre* (strona 113). It makes your calibre collection available over the net. You can browse it on your device using a simple browser or a dedicated application. First perform the following steps in calibre:

- Set the *Preferred Output Format* in calibre to EPUB for normal Android devices or MOBI for Kindles (The output format can be set under *Preferences* → *Interface* → *Behavior*)
- Convert the books you want to read on your device to EPUB/MOBI format by selecting them and clicking the *Convert* button.
- Turn on the *Content server* in calibre's preferences and leave calibre running.

Teraz na urządzeniu z Androidem otwórz przeglądarkę i przejdź do

<http://192.168.1.2:8080/>

Zastąp 192.168.1.2 lokalnym adresem IP komputera z uruchomionym calibre. Zobacz *Serwer zawartości calibre* (strona 113), aby uzyskać szczegółowe informacje o uruchomieniu serwera i znalezieniu właściwego adresu IP.

You can now browse your book collection and download books from calibre to your device to open with whatever e-book reading software you have on your Android device.

Many reading apps support browsing the calibre library directly via its *OPDS* support. In such apps you can go to the online catalog screen and add the IP address of the calibre server to browse and download books from your calibre library within the app.

9.2.7 Can I access my calibre books using the web browser in my Kindle or other reading device?

calibre has a *Content server* that exports the books in calibre as a web page. See *Serwer zawartości calibre* (strona 113) for details.

Some devices, like the Kindle (1/2/DX), do not allow you to access port 8080 (the default port on which the content server runs). In that case, change the port in the calibre Preferences to 80. (On some operating systems, you may not be able to run the server on a port number less than 1024 because of security settings. In this case the simplest solution is to adjust your router to forward requests on port 80 to port 8080).

Also some devices do not have browsers advanced enough to run the app-like interface used by the Content server. For such devices, you can simply add */mobile* to the server URL to get a simplified, non-JavaScript interface.

9.2.8 Nie mogę wysłać e-maili przy użyciu calibre?

Because of the large amount of spam in email, sending email can be tricky, as different mail servers use different strategies to block email. The most common problem is if you are sending email directly (without a mail relay) in calibre. Many servers (for example, Amazon) block email that does not come from a well known relay. The most robust way to setup email sending in calibre is to do the following:

- Create a free GMX account at [GMX](https://www.gmx.com)⁵⁵.
- Goto *Preferences* → *Sharing* → *Sharing books by email* in calibre and click the *Use GMX* button and fill in the information asked for.

⁵⁵ <https://www.gmx.com>

- Zaloguj się do swojego konta GMX na stronie internetowej i włącz wysyłanie SMTP (*Ustawienia->POP3 i IMAP->Wysyłaj i odbieraj wiadomości e-mail przez zewnętrzny program*)
- calibre będzie wtedy mógł używać GMX do wysyłania poczty.
- Jeśli wysyłasz na swój Kindle, pamiętaj, aby zaktualizować preferencje e-mail na stronie Amazon Kindle, aby umożliwić wysyłanie wiadomości e-mail z Twojego adresu e-mail GMX. Należy również pamiętać, że Amazon nie zezwala na dostarczanie pocztą e-mail plików AZW3 i MOBI w nowym stylu (KF8). Wreszcie, Amazon zaczął ostatnio wysyłać e-maile z potwierdzeniem, które musisz kliknąć z powrotem na swoje konto GMX, zanim książka zostanie faktycznie dostarczona.

Nawet po wykonaniu tej czynności możesz mieć problemy. Jednym z typowych źródeł problemów jest to, że niektóre źle zaprojektowane programy antywirusowe blokują Calibre przed otwarciem połączenia w celu wysłania wiadomości e-mail. Spróbuj dodać wykluczenie dla calibre w swoim programie antywirusowym.

Informacja: Microsoft/GMX can disable your account if you use it to send large amounts of email. So, when using these services to send mail calibre automatically restricts itself to sending one book every five minutes. If you don't mind risking your account being blocked you can reduce this wait interval by going to *Preferences* → *Advanced* → *Tweaks* in calibre.

Informacja: Google recently deliberately broke their email sending protocol (SMTP) support in an attempt to force everyone to use their web interface so they can show you more ads. They are trying to claim that SMTP is insecure, that is incorrect and simply an excuse. Use some other email provider instead.

Informacja: If you are concerned about giving calibre access to your email account, simply create a new free email account with GMX or Outlook and use it only for calibre.

9.2.9 Moje urządzenie jest montowane tylko do odczytu w systemie Linux, więc Calibre nie może się z nim połączyć?

Linux kernels mount devices read-only when their filesystems have errors. You can repair the filesystem with:

```
sudo fsck.vfat -y /dev/sdc
```

Replace /dev/sdc with the path to the device node of your device. You can find the device node of your device, which will always be under /dev by examining the output of:

```
mount
```

9.2.10 Dlaczego calibre nie obsługuje kolekcji na Kindle lub półkach w Nook?

Neither the Kindle nor the Nook provide any way to manipulate collections over a USB connection. If you really care about using collections, I would urge you to sell your Kindle/Nook and get a Kobo. Only Kobo seems to understand that life is too short to be entering collections one by one on an e-ink screen ☹

Note that in the case of the Kindle, there is a way to manipulate collections via USB, but it requires that the Kindle be rebooted *every time* it is disconnected from the computer, for the changes to the collections to be recognized. As such,

it is unlikely that any calibre developers will ever feel motivated enough to support it. There is however, a calibre plugin that allows you to create collections on your Kindle from the calibre metadata. It is available [from here](#)⁵⁶.

Informacja: Amazon have removed the ability to manipulate collections completely in their newer models, like the Kindle Touch and Kindle Fire, making even the above plugin useless, unless you root your Kindle and install custom firmware.

9.2.11 Otrzymuję błąd, gdy próbuję użyć calibre z moim Kobo Touch/Glo/itd.?

Kobo ma bardzo wadliwe oprogramowanie. Wiadomo, że połączenie się z nim kończy się niepowodzeniem. Niektóre kombinacje płyty głównej, portów/kabli/koncentratorów USB mogą nasilić tę tendencję do awarii. Jeśli pojawia się błąd podczas łączenia się z touch z calibre, spróbuj wykonać następujące czynności, z których każda rozwiązała problem dla *niektórych* użytkowników calibre.

- Podłącz Kobo bezpośrednio do komputera, nie przez koncentrator USB
- Wypróbuj inny kabel USB i inny port USB w komputerze
- Wyloguj się z Kobo i zaloguj się ponownie, co spowoduje odbudowanie bazy danych, naprawiając uszkodzone błędy bazy danych.
- Spróbuj zaktualizować oprogramowanie w swoim Kobo Touch do najnowszej
- Spróbuj zresetować Kobo (czasami to leczy problem na chwilę, ale potem pojawia się ponownie, w takim przypadku musisz resetować raz za razem)
- Staraj się umieszczać tylko jedną lub dwie książki na Kobo na raz i nie przechowuj dużych zbiorów na Kobo

9.2.12 Okładki książek, które wysyłam do mojego e-ink Kindle, pojawiają się natychmiast, a następnie są zastępowane przez okładkę ogólną?

This happens because of an Amazon bug. They try to download a cover for the book from their servers and when that fails, they replace the existing cover that calibre created with a generic cover. For details see [this forum thread](#)⁵⁷. As of version 4.17, calibre has a workaround, where if you connect the Kindle to calibre after the covers have been destroyed by Amazon, calibre will restore them automatically. So in order to see the covers on your Kindle, you have to:

- 1) Wyślij książkę do Kindle przez calibre
- 2) Odłącz Kindle i poczekaj, aż Amazon zniszczy obudowę
- 3) Podłącz ponownie Kindle do calibre

Note that this workaround only works for books sent with calibre 4.17 or later. Alternately, simply keep your Kindle in airplane mode, you don't really want Amazon knowing every book you read anyway. I encourage you to contact Amazon customer support and complain loudly about this bug. Maybe Amazon will listen.

Informacja: If the workaround is not working for you make sure you Kindle firmware is at least version 5.12.5, released in April 2020.

⁵⁶ <https://www.mobileread.com/forums/showthread.php?t=244202>

⁵⁷ <https://www.mobileread.com/forums/showthread.php?t=329945>

9.2.13 The covers for my MOBI files have stopped showing up in Kindle for PC/Kindle for Android/iPad etc.

This is caused by a bug in the Amazon software. You can work around it by going to *Preferences* → *Conversion* → *Output Options* → *MOBI output* and setting the *Enable sharing of book content* option. If you are reconverting a previously converted book, you will also have to enable the option in the conversion dialog for that individual book (as per book conversion settings are saved and take precedence).

Note that doing this will mean that the generated MOBI will show up under personal documents instead of Books on the Kindle Fire and Amazon whispersync will not work, but the covers will. It's your choice which functionality is more important to you. I encourage you to contact Amazon and ask them to fix this bug.

The bug in Amazon's software is that when you put a MOBI file on a Kindle, unless the file is marked as a Personal document, Amazon assumes you bought the book from it and tries to download the cover thumbnail for it from its servers. When the download fails, it refuses to fallback to the cover defined in the MOBI file. This is likely deliberate on Amazon's part to try to force authors to sell only through them. In other words, the Kindle only displays covers for books marked as Personal Documents or books bought directly from Amazon.

If you send a MOBI file to an e-ink Kindle with calibre using a USB connection, calibre works around this Amazon bug by uploading a cover thumbnail itself. However, that workaround is only possible when using a USB connection and sending with calibre. Note that if you send using email, Amazon will automatically mark the MOBI file as a Personal Document and the cover will work, but the book will show up in Personal Documents.

9.2.14 I transferred some books to my Kindle using calibre and they did not show up?

Books sent to the Kindle only show up on the Kindle after they have been *indexed* by the Kindle. This can take some time. If the book still does not show up after some time, then it is likely that the Kindle indexer crashed. Sometimes a particular book can cause the indexer to crash. Unfortunately, Amazon has not provided any way to deduce which book is causing a crash on the Kindle. Your only recourse is to either reset the Kindle, or delete all files from its memory using Windows Explorer (or whatever file manager you use) and then send the books to it again, one by one, until you discover the problem book. Once you have found the problem book, delete it off the Kindle and do a MOBI to MOBI or MOBI to AZW3 conversion in calibre and then send it back. This will most likely take care of the problem.

9.3 Library management

Spis treści

- *Where are the book files stored?* (strona 140)
- *How does calibre manage author names and sorting?* (strona 140)
- *Why doesn't calibre let me store books in my own folder structure?* (strona 141)
- *Why doesn't calibre have a column for foo?* (strona 142)
- *Can I have a column showing the formats or the ISBN?* (strona 142)
- *How do I move my calibre data from one computer to another?* (strona 142)
- *The list of books in calibre is blank!* (strona 143)
- *I am getting errors with my calibre library on a networked drive/NAS?* (strona 143)

9.3.1 Where are the book files stored?

When you first run calibre, it will ask you for a folder in which to store your books. Whenever you add a book to calibre, it will copy the book into that folder. Books in the folder are nicely arranged into sub-folders by Author and Title. Note that the contents of this folder are automatically managed by calibre, **do not** add any files/folders manually to this folder, as they may be automatically deleted. If you want to add a file associated to a particular book, use the top right area of *Edit metadata* dialog to do so. Then, calibre will automatically put that file into the correct folder and move it around when the title/author changes.

Metadata about the books is stored in the file `metadata.db` at the top level of the library folder. This file is a sqlite database. When backing up your library make sure you copy the entire folder and all its sub-folders.

The library folder and all its contents make up what is called a calibre library. You can have multiple such libraries. To manage the libraries, click the calibre icon on the toolbar. You can create new libraries, remove/rename existing ones and switch between libraries easily.

You can copy or move books between different libraries (once you have more than one library setup) by right clicking on a book and selecting the *Copy to library* action.

9.3.2 How does calibre manage author names and sorting?

Author names are complex, especially across cultures, see [this note](#)⁵⁸ for some of the complexities. calibre has a very flexible strategy for managing author names. The first thing to understand is that books and authors are separate entities in calibre. A book can have more than one author, and an author can have more than one book. You can manage the authors of a book by the edit metadata dialog. You can manage individual authors by right clicking on the author in the Tag browser on the left of the main calibre window and selecting *Manage authors*. Using this dialog you can change the name of an author and also how that name is sorted. This will automatically change the name of the author in all the books of that author. When a book has multiple authors, separate their names using the & character.

Now coming to author name sorting:

- When a new author is added to calibre (this happens whenever a book by a new author is added), calibre automatically computes a sort string for both the book and the author.
- Authors in the Tag browser are sorted by the sort value for the **authors**. Remember that this is different from the Author sort field for a book.
- By default, this sort algorithm assumes that the author name is in `First name Last name` format and generates a `Last name, First name` sort value.
- You can change this algorithm by going to *Preferences* → *Advanced* → *Tweaks* and setting the `author_sort_copy_method` tweak.
- You can force calibre to recalculate the author sort values for every author by right clicking on any author and selecting *Manage authors*, then pushing the *Recalculate all author sort values* button. Do this after you have set the `author_sort_copy_method` tweak to what you want.
- You can force calibre to recalculate the author sort values for all books by using the bulk metadata edit dialog (select all books and click edit metadata, check the *Automatically set author sort* checkbox, then press OK).
- When recalculating the author sort values for books, calibre uses the author sort values for each individual author. Therefore, ensure that the individual author sort values are correct before recalculating the books' author sort values.
- You can control whether the Tag browser display authors using their names or their sort values by setting the `categories_use_field_for_author_name` tweak in *Preferences* → *Advanced* → *Tweaks*

⁵⁸ <https://www.w3.org/International/questions/qa-personal-names.en.php?changelang=en>

Note that you can set an individual author's sort value to whatever you want using *Manage authors*. This is useful when dealing with names that calibre will not get right, such as complex multi-part names like Miguel de Cervantes Saavedra or when dealing with Asian names like Sun Tzu.

With all this flexibility, it is possible to have calibre manage your author names however you like. For example, one common request is to have calibre display author names LN, FN. To do this, and if the note below does not apply to you, then:

- Set the `author_sort_copy_method` tweak to `copy` as described above.
- Restart calibre. Do not change any book metadata before doing the remaining steps.
- Change all author names to LN, FN using the Manage authors dialog.
- After you have changed all the authors, press the *Recalculate all author sort values* button.
- Press OK, at which point calibre will change the authors in all your books. This can take a while.

Informacja:

When changing from FN LN to LN, FN, it is often the case that the values in `author_sort` are already in LN, FN format. If this is your case, then do the following:

- Set the `author_sort_copy_method` tweak to `copy` as described above.
 - Restart calibre. Do not change any book metadata before doing the remaining steps.
 - Open the Manage authors dialog. Press the `copy all author sort values to author` button.
 - Check through the authors to be sure you are happy. You can still press Cancel to abandon the changes. Once you press OK, there is no undo.
 - Press OK, at which point calibre will change the authors in all your books. This can take a while.
-

9.3.3 Why doesn't calibre let me store books in my own folder structure?

The whole point of calibre's library management features is that they provide a search and sort based interface for locating books that is *much* more efficient than any possible folder scheme you could come up with for your collection. Indeed, once you become comfortable using calibre's interface to find, sort and browse your collection, you won't ever feel the need to hunt through the files on your disk to find a book again. By managing books in its own folder structure of Author -> Title -> Book files, calibre is able to achieve a high level of reliability and standardization. To illustrate why a search/tagging based interface is superior to folders, consider the following. Suppose your book collection is nicely sorted into folders with the following scheme:

```
Genre -> Author -> Series -> ReadStatus
```

Now this makes it very easy to find for example all science fiction books by Isaac Asimov in the Foundation series. But suppose you want to find all unread science fiction books. There's no easy way to do this with this folder scheme, you would instead need a folder scheme that looks like:

```
ReadStatus -> Genre -> Author -> Series
```

In calibre, you would instead use tags to mark genre and read status and then just use a simple search query like `tag:scifi` and not `tag:read`. calibre even has a nice graphical interface, so you don't need to learn its search language instead you can just click on tags to include or exclude them from the search.

To those of you that claim that you need access to the filesystem, so that you can have access to your books over the network, calibre has an excellent Content server that gives you access to your calibre library over the net.

If you are worried that someday calibre will cease to be developed, leaving all your books marooned in its folder structure, explore the powerful *Save to disk* feature in calibre that lets you export all your files into a folder structure of arbitrary complexity based on their metadata.

Finally, the reason there are numbers at the end of every title folder, is for *robustness*. That number is the id number of the book record in the calibre database. The presence of the number allows you to have multiple records with the same title and author names. It is also part of what allows calibre to magically regenerate the database with all metadata if the database file gets corrupted. Given that calibre's mission is to get you to stop storing metadata in filenames and stop using the filesystem to find things, the increased robustness afforded by the id numbers is well worth the uglier folder names.

If you are still not convinced, then I'm afraid calibre is not for you. Look elsewhere for your book cataloguing needs. Just so we're clear, **this is not going to change**. Kindly do not contact us in an attempt to get us to change this.

9.3.4 Why doesn't calibre have a column for foo?

calibre is designed to have columns for the most frequently and widely used fields. In addition, you can add any columns you like. Columns can be added via *Preferences* → *Interface* → *Add your own columns*. Watch the tutorial [UI Power tips](#)⁵⁹ to learn how to create your own columns, or read [this blog post](#)⁶⁰.

You can also create „virtual columns” that contain combinations of the metadata from other columns. In the add column dialog use the *Quick create* links to easily create columns to show the book ISBN or formats. You can use the powerful calibre template language to do much more with columns. For more details, see *Język szablonów calibre* (strona 161).

9.3.5 Can I have a column showing the formats or the ISBN?

Yes, you can. Follow the instructions in the answer above for adding custom columns.

9.3.6 How do I move my calibre data from one computer to another?

You can export all calibre data (books, settings and plugins) and then import it on another computer. First let's see how to export the data:

- Right click the calibre icon in the main calibre toolbar and select *Export/import all calibre data*. Note that if there is currently a device connected, this menu option will not be available – so, disconnect any connected devices. Then click the button labelled *Export all your calibre data*. You will see a list of all your calibre libraries. Click OK and choose an empty folder somewhere on your computer. The exported data will be saved in this folder. Simply copy this folder to your new computer and follow the instructions below to import the data.
- Install calibre on your new computer and run through the *Welcome wizard*, it does not matter what you do there, as you will be importing your old settings in the next step. You will now have an empty calibre, with just the *Getting Started* guide in your library. Once again, right click the calibre button and choose *Export/import all calibre data*. Then click the button labelled *Import previously exported data*. Select the folder with the exported data that you copied over earlier. You will now have a list of libraries you can import. Go through the list one by one, and select the new location for each library (a location is just an empty folder somewhere on your computer). Click OK. After the import completes, calibre will restart, with all your old libraries, settings and calibre plugins.

Informacja: This import/export functionality is only available from calibre version 2.47 onwards. If you have an older version of calibre, or if you encounter problems with the import/export, you can just copy over your calibre library folder manually, as described in the next paragraph.

⁵⁹ <https://calibre-ebook.com/pl/demo#tutorials>

⁶⁰ <https://blog.calibre-ebook.com/calibre-custom-columns/>

Simply copy the calibre library folder from the old to the new computer. You can find out what the library folder is by clicking the calibre icon in the toolbar. Choose the *Switch/create calibre library* action and you will see the path to the current calibre library.

Now on the new computer, start calibre for the first time. It will run the *Welcome wizard* asking you for the location of the calibre library. Point it to the previously copied folder. If the computer you are transferring to already has a calibre installation, then the *Welcome wizard* won't run. In that case, right-click the calibre icon in the toolbar and point it to the newly copied folder. You will now have two calibre libraries on your computer and you can switch between them by clicking the calibre icon on the toolbar. Transferring your library in this manner preserves all your metadata, tags, custom columns, etc.

9.3.7 The list of books in calibre is blank!

In order to understand why that happened, you have to understand what a calibre library is. At the most basic level, a calibre library is just a folder. Whenever you add a book to calibre, that book's files are copied into this folder (arranged into sub folders by author and title). Inside the calibre library folder, at the top level, you will see a file called `metadata.db`. This file is where calibre stores the metadata like title/author/rating/tags etc. for *every* book in your calibre library. The list of books that calibre displays is created by reading the contents of this `metadata.db` file.

There can be two reasons why calibre is showing a empty list of books:

- Your calibre library folder changed its location. This can happen if it was on an external disk and the drive letter for that disk changed. Or if you accidentally moved the folder. In this case, calibre cannot find its library and so starts up with an empty library instead. To remedy this, do a right-click on the calibre icon in the calibre toolbar and select *Switch/create library*. Click the little blue icon to select the new location of your calibre library and click OK. If you don't know the new location search your computer for the file `metadata.db`.
- Your `metadata.db` file was deleted/corrupted. In this case, you can ask calibre to rebuild the `metadata.db` from its backups. Right click the calibre icon in the calibre toolbar and select *Library maintenance->Restore database*. calibre will automatically rebuild `metadata.db`.

9.3.8 I am getting errors with my calibre library on a networked drive/NAS?

Do not put your calibre library on a networked drive.

A filesystem is a complex beast. Most network filesystems lack various filesystem features that calibre uses. Some don't support file locking, some don't support hardlinking, some are just flaky. Additionally, calibre is a single user application, if you accidentally run two copies of calibre on the same networked library, bad things will happen. Finally, different OSes impose different limitations on filesystems, so if you share your networked drive across OSes, once again, bad things *will* happen.

Consider using the calibre Content server to make your books available on other computers. Run calibre on a single computer and access it via the Content server or a Remote Desktop solution.

If you must share the actual library, use a file syncing tool like DropBox or rsync instead of a networked drive. If you are using a file-syncing tool it is **essential** that you make sure that both calibre and the file syncing tool do not try to access the calibre library at the same time. In other words, **do not** run the file syncing tool and calibre at the same time.

Even with these tools there is danger of data corruption/loss, so only do this if you are willing to live with that risk. In particular, be aware that **Google Drive** is incompatible with calibre, if you put your calibre library in Google Drive, **you will suffer data loss**. See [this thread](https://www.mobileread.com/forums/showthread.php?t=205581)⁶¹ for details.

⁶¹ <https://www.mobileread.com/forums/showthread.php?t=205581>

9.4 Różne

Spis treści

- *Amazon is stopping email delivery of MOBI files?* (strona 144)
- *I want calibre to download news from my favorite news website.* (strona 145)
- *Why the name calibre?* (strona 145)
- *Why does calibre show only some of my fonts on macOS?* (strona 145)
- *calibre is not starting on Windows?* (strona 145)
- *calibre freezes/crashes occasionally?* (strona 146)
- *The calibre E-book viewer and Edit book tools do not work on Windows?* (strona 147)
- *Using the viewer or doing any conversions results in a permission denied error on Windows* (strona 147)
- *calibre is not starting/crashing on macOS?* (strona 147)
- *I get only a black or white screen when running the calibre E-book viewer?* (strona 148)
- *I downloaded the installer, but it is not working?* (strona 148)
- *My antivirus program claims calibre is a virus/trojan?* (strona 148)
- *How do I backup calibre?* (strona 149)
- *How do I use purchased EPUB books with calibre (or what do I do with .acsm files)?* (strona 149)
- *I am getting a „Permission Denied” error?* (strona 149)
- *Can I have the comment metadata show up on my reader?* (strona 150)
- *How do I get calibre to use my HTTP proxy?* (strona 150)
- *I want some feature added to calibre. What can I do?* (strona 150)
- *Why doesn't calibre have an automatic update?* (strona 150)
- *How is calibre licensed?* (strona 151)
- *How do I run calibre from my USB stick?* (strona 151)
- *How do I run parts of calibre like news download and the Content server on my own Linux server?* (strona 151)

9.4.1 Amazon is stopping email delivery of MOBI files?

Amazon have announced⁶² that they will stop accepting MOBI files emailed to @kindle.com email addresses. You can instruct calibre to send EPUB instead of MOBI by going to *Preferences* → *Sharing books by email* and then removing MOBI from the list of formats to send to your @kindle.com email address and adding EPUB instead.

Note however, that Amazon's EPUB intake is very flawed, they will reject a number of EPUB files that work everywhere else. In such cases you can try the following trick:

1. Convert the EPUB file to MOBI
2. Then convert the MOBI file back to EPUB and send the resulting EPUB file

⁶² <https://blog.the-ebook-reader.com/2022/05/03/amazon-dropping-mobi-support-on-send-to-kindle-apps/>

This will remove all advanced formatting, embedded fonts, etc., but greatly increase the chances of Amazon accepting the EPUB.

Informacja: If you were previously using email delivery of periodicals downloaded by calibre, you will be better off sending those by USB cable or downloading them from the calibre Content server via the Kindle's built-in browser. However, if you want to continue using email delivery you can try changing the output format in Preferences->Behavior to EPUB, then calibre will download the news in EPUB format. Whether Amazon will accept the EPUB or not is a whole other question.

9.4.2 I want calibre to download news from my favorite news website.

If you are reasonably proficient with computers, you can teach calibre to download news from any website of your choosing. To learn how to do this see [Dodawanie ulubionej strony z wiadomościami](#) (strona 31).

Otherwise, you can request a particular news site by posting in the [calibre Recipes forum](#)⁶³.

9.4.3 Why the name calibre?

Wybierz

- Converter And LIBRARY for E-books
- A high *calibre* product
- A tribute to the SONY Librie which was the first e-ink based e-book reader
- My wife chose it ;-)

calibre is pronounced as cal-i-ber *not* ca-li-bre. If you're wondering, calibre is the British/commonwealth spelling for caliber. Being Indian, that's the natural spelling for me.

9.4.4 Why does calibre show only some of my fonts on macOS?

calibre embeds fonts in e-book files it creates. E-book files support embedding only TrueType and OpenType (.ttf and .otf) fonts. Most fonts on macOS systems are in .dfont format, thus they cannot be embedded. calibre shows only TrueType and OpenType fonts found on your system. You can obtain many such fonts on the web. Simply download the .ttf/.otf files and add them to the Library/Fonts folder in your home folder.

9.4.5 calibre is not starting on Windows?

There can be several causes for this:

- If you get no errors but the calibre window does not appear, it has probably just appeared off screen. You can gather all windows onto the current screen using one of the techniques described [here](#)⁶⁴.
- If you get an error about calibre not being able to open a file because it is in use by another program, do the following:
 - Uninstall calibre
 - Reboot your computer

⁶³ <https://www.mobileread.com/forums/forumdisplay.php?f=228>

⁶⁴ <https://www.wikihow.com/Bring-an-Off-Screen-Window-Back-on-Windows>

- Re-install calibre. But do not start calibre from the installation wizard.
 - Temporarily disable your antivirus program (disconnect from the Internet before doing so, to be safe)
 - Look inside the folder you chose for your calibre library. If you see a file named `metadata.db`, delete it.
 - Uruchom calibre
 - From now on you should be able to start calibre normally.
- If you get an error about a Python function terminating unexpectedly after upgrading calibre, first uninstall calibre, then delete the folders (if they exist) `C:\Program Files\Calibre` and `C:\Program Files\Calibre2`. Now re-install and you should be fine.
 - If you get an error in the *Welcome wizard* on an initial run of calibre, try choosing a folder like `C:\library` as the calibre library (calibre sometimes has trouble with library locations if the path contains non-English characters, or only numbers, etc.)
 - Try running it as administrator (Right click on the icon and select *Run as administrator*)

If it still won't launch, start a command prompt (press the Windows key and R; then type `cmd.exe` in the Run dialog that appears). At the command prompt type the following command and press Enter:

```
calibre-debug -g
```

Post any output you see in a help message on the [Forum](#)⁶⁵.

9.4.6 calibre freezes/crashes occasionally?

There are several possible things I know of, that can cause this:

- You recently connected an external monitor or TV to your computer. In this case, whenever calibre opens a new window like the edit metadata window or the conversion dialog, it appears on the second monitor where you don't notice it and so you think calibre has frozen. Disconnect your second monitor and restart calibre.
- The following programs have been reported to cause crashes in calibre: If you are running any of these, close them before starting calibre, or uninstall them: *RoboForm*, *Logitech SetPoint Settings*, *Constant Guard Protection by Xfinity*, *Spybot*, *Killer Network Manager*, *Nahimic UI Interface*, *Acronis True Image*.
- You are using a Wacom branded USB mouse/tablet. There is an incompatibility between Wacom drivers and the graphics toolkit calibre uses. Try using a non-Wacom mouse.
- On some 64 bit versions of Windows there are security software/settings that prevent 64-bit calibre from working properly. If you are using the 64-bit version of calibre try switching to the 32-bit version.
- If the crash happens when you are trying to copy text from the calibre E-book viewer, it is most likely caused by some clipboard monitoring/managing application you have running. Turn it off and you should be fine.
- If the crashes happen specifically when you are using a file dialog, like clicking on the *Add books* button or the *Save to Disk* button, then you have some software that has installed broken Shell extensions on your computer. Known culprits include: *SpiderOak*, *odrive sync* and *Dell Backup and Recovery* and *NetDrive*. If you have one of these, uninstall them and you will be fine. You can also use the [NirSoft Shell Extension Viewer](#)⁶⁶ to see what shell extensions are installed on your system and disable them individually, if you don't want to uninstall the full program. Remember to use „Restart Explorer” or reboot your computer after disabling the shell extensions.

If none of the above apply to you, then there is some other program on your computer that is interfering with calibre. First reboot your computer in safe mode, to have as few running programs as possible, and see if the crashes still happen. If they do not, then you know it is some program causing the problem. The most likely such culprit is a program that

⁶⁵ <https://www.mobileread.com/forums/forumdisplay.php?f=166>

⁶⁶ <https://www.nirsoft.net/utils/shexview.html>

modifies other programs» behavior, such as an antivirus, a device driver, something like RoboForm (an automatic form filling app) or an assistive technology like Voice Control or a Screen Reader.

The only way to find the culprit is to eliminate the programs one by one and see which one is causing the issue. Basically, stop a program, run calibre, check for crashes. If they still happen, stop another program and repeat.

9.4.7 The calibre E-book viewer and Edit book tools do not work on Windows?

These two programs use hardware acceleration as they embed a version of the Chrome browser to render HTML. If they do not work it will be because of incompatibility with your system's GPU (graphics) drivers. Try updating these first, and reboot. If that does not fix it, you can set the `QTWEBENGINE_CHROMIUM_FLAGS` environment variable to the value `--disable-gpu` to turn off hardware acceleration. See [this page](#)⁶⁷ for details.

9.4.8 Using the viewer or doing any conversions results in a permission denied error on Windows

Something on your computer is preventing calibre from accessing its own temporary files. Most likely the permissions on your Temp folder are incorrect. Go to the folder file: `C:\Users\USERNAME\AppData\Local` in Windows Explorer and then right click on the file: Temp folder, select *Properties* and go to the *Security* tab. Make sure that your user account has full control for this folder.

Some users have reported that running the following command in an Administrator Command Prompt fixed their permissions. To get an Administrator Command Prompt search for `cmd.exe` in the start menu, then right click on the command prompt entry and select *Run as administrator*. At the command prompt type the following command and press Enter:

```
icacls "%appdata%\..\Local\Temp" /reset /T
```

Alternately, you can run calibre as Administrator, but doing so will cause some functionality, such as drag and drop to not work.

Finally, some users have reported that disabling UAC fixes the problem.

9.4.9 calibre is not starting/crashing on macOS?

One common cause of failures on macOS is the use of accessibility technologies that are incompatible with the graphics toolkit calibre uses. Try turning off VoiceOver if you have it on. Also go to System Preferences->System->Universal Access and turn off the setting for enabling access for assistive devices in all the tabs. Another cause can be some third party apps that modify system behavior, such as Smart Scroll.

You can obtain debug output about why calibre is not starting by running *Console.app*. Debug output will be printed to it. If the debug output contains a line that looks like:

```
Qt: internal: -108: Error ATSUMeasureTextImage text/qfontengine_mac.mm
```

then the problem is probably a corrupted font cache. You can clear the cache by following these [instructions](#)⁶⁸. If that doesn't solve it, look for a corrupted font file on your system, in `~/Library/Fonts` or the like. An easy way to check for corrupted fonts in macOS is to start the „Font Book” application, select all fonts and then in the File menu, choose „Validate fonts”.

⁶⁷ <https://doc.qt.io/qt-6/qtwebengine-debugging.html>

⁶⁸ <https://www.macworld.com/article/1139383/fontcacheclear.html>

9.4.10 I get only a black or white screen when running the calibre E-book viewer?

This will be because of an incompatibility between Qt WebEngine, which the viewer uses to render and the GPU drivers on your system. First try upgrading the GPU drivers. If that does not help, you can try turning off hardware acceleration in Qt WebEngine by setting the environment variable `QTWEBENGINE_CHROMIUM_FLAGS` to the value `--disable-gpu`. See *Zmienne środowiskowe* (strona 289) for how to change environment variables.

9.4.11 I downloaded the installer, but it is not working?

Downloading from the Internet can sometimes result in a corrupted download. If the calibre installer you downloaded is not opening, try downloading it again. If re-downloading it does not work, download it from [an alternate location](#)⁶⁹. If the installer still doesn't work, then something on your computer is preventing it from running.

- Try temporarily disabling your antivirus program (Microsoft Security Essentials, or Kaspersky or Norton or McAfee or whatever). This is most likely the culprit if the upgrade process is hanging in the middle.
- Similarly, if the installer is failing/rolling back and you have Microsoft PowerToys running, quit it.
- Try rebooting your computer and running a registry cleaner like [Wise registry cleaner](#)⁷⁰.
- Try a clean install. That is, uninstall calibre, delete `C:\Program Files\Calibre2` (or wherever you previously chose to install calibre). Then re-install calibre. Note that uninstalling does not touch your books or settings.
- Try downloading the installer with an alternate browser. For example if you are using Microsoft Edge, try using Firefox or Chrome instead.
- If you get an error about a missing DLL on Windows, then most likely, the permissions on your temporary folder are incorrect. Go to the folder `C:\Users\USERNAME\AppData\Local` in Windows Explorer and then right click on the Temp folder and select *Properties* and go to the *Security* tab. Make sure that your user account has full control for this folder.

If you still cannot get the installer to work and you are on Windows, you can use the [calibre portable install](#)⁷¹, which does not need an installer (it is just a ZIP file).

9.4.12 My antivirus program claims calibre is a virus/trojan?

The first thing to check is that you are downloading calibre from the [official website](#)⁷². Make sure you are clicking the download links on the left, not the advertisements on the right. calibre is a very popular program and unscrupulous people try to setup websites offering it for download to fool the unwary.

If you have the official download and your antivirus program is still claiming calibre is a virus, then, your antivirus program is wrong. Antivirus programs use heuristics, patterns of code that „look suspicious” to detect viruses. It's rather like racial profiling. calibre is a completely open source product. You can actually browse the source code yourself (or hire someone to do it for you) to verify that it is not a virus. Please report the false identification to whatever company you buy your antivirus software from. If the antivirus program is preventing you from downloading/installing calibre, disable it temporarily, install calibre and then re-enable it.

⁶⁹ <https://github.com/kovidgoyal/calibre/releases/latest>

⁷⁰ <https://www.wisecleaner.com>

⁷¹ https://calibre-ebook.com/pl/download_portable

⁷² <https://calibre-ebook.com/pl/download>

9.4.13 How do I backup calibre?

The most important thing to backup is the calibre library folder, that contains all your books and metadata. This is the folder you chose for your calibre library when you ran calibre for the first time. You can get the path to the library folder by clicking the calibre icon on the main toolbar. You must backup this complete folder with all its files and sub-folders.

You can switch calibre to using a backed up library folder by simply clicking the calibre icon on the toolbar and choosing your backup library folder. A backed up library folder backs up your custom columns and saved searches as well as all your books and metadata.

If you want to backup the calibre configuration/plugins, you have to backup the config folder. You can find this config folder via *Preferences* → *Miscellaneous*. Note that restoring configuration folders is not officially supported, but should work in most cases. Just copy the contents of the backup folder into the current configuration folder to restore.

9.4.14 How do I use purchased EPUB books with calibre (or what do I do with .acsm files)?

Most purchased EPUB books have *DRM* (strona 375). This prevents calibre from opening them. You can still use calibre to store and transfer them to your e-book reader. First, you must authorize your reader on a Windows machine with Adobe Digital Editions. Once this is done, EPUB books transferred with calibre will work fine on your reader. When you purchase an epub book from a website, you will get an „.acsm” file. This file should be opened with Adobe Digital Editions, which will then download the actual „.epub” e-book. The e-book file will be stored in the folder „My Digital Editions”, from where you can add it to calibre.

9.4.15 I am getting a „Permission Denied” error?

A permission denied error can occur because of many possible reasons, none of them having anything to do with calibre.

- You can get permission denied errors if you are using an SD card with write protect enabled.
- W systemie macOS, jeśli podczas podłączania urządzenia do calibre pojawiają się błędy uprawnień, możesz to naprawić, zaglądając do: *guilabel: Preferencje systemowe > Bezpieczeństwo i prywatność > Prywatność > Pliki i foldery*.
- If you, or some program you used changed the file permissions of the files in question to read only.
- If there is a filesystem error on the device which caused your operating system to mount the filesystem in read only mode or mark a particular file as read only pending recovery.
- If the files have their owner set to a user other than you.
- If your file is open in another program.
- If the file resides on a device, you may have reached the limit of a maximum of 256 files in the root of the device. In this case you need to reformat the device/sd card referred to in the error message with a FAT32 filesystem, or delete some files from the SD card/device memory.

You will need to fix the underlying cause of the permissions error before resuming to use calibre. Read the error message carefully, see what file it points to and fix the permissions on that file or its containing folders.

9.4.16 Can I have the comment metadata show up on my reader?

Most readers do not support this. You should complain to the manufacturer about it and hopefully if enough people complain, things will change. In the meantime, you can insert the metadata, including comments into a „Jacket page” at the start of the e-book, by using the option to „Insert metadata as page at start of book” during conversion. The option is found in the *Structure detection* section of the conversion settings. Note that for this to have effect you have to *convert* the book. If your book is already in a format that does not need conversion, you can convert from that format to the same format.

Another alternative is to create a catalog in e-book form containing a listing of all the books in your calibre library, with their metadata. Click-and-hold the *Convert* button to access the catalog creation tool. And before you ask, no you cannot have the catalog „link directly to” books on your reader.

9.4.17 How do I get calibre to use my HTTP proxy?

By default, calibre uses whatever proxy settings are set in your OS. Sometimes these are incorrect, for example, on Windows if you don't use Microsoft Edge then the proxy settings may not be up to date. You can tell calibre to use a particular proxy server by setting the `http_proxy` and `https_proxy` environment variables. The format of the variable is: `http://username:password@servername` you should ask your network administrator to give you the correct value for this variable. Note that calibre only supports HTTP proxies not SOCKS proxies. You can see the current proxies used by calibre in Preferences->Miscellaneous.

9.4.18 I want some feature added to calibre. What can I do?

You have two choices:

1. Create a patch by hacking on calibre and send it to me for review and inclusion. See [Development](#)⁷³.
2. [Open a bug requesting the feature](#)⁷⁴. Remember that while you may think your feature request is extremely important/essential, calibre developers might not agree. Fortunately, calibre is open source, which means you always have the option of implementing your feature yourself, or hiring someone to do it for you. Furthermore, calibre has a comprehensive plugin architecture, so you might be able to develop your feature as a plugin, see *Pisanie własnych wtyczek, by rozszerzyć funkcjonalność calibre* (strona 224).

9.4.19 Why doesn't calibre have an automatic update?

For many reasons:

- *There is no need to update every week.* If you are happy with how calibre works turn off the update notification and be on your merry way. Check back to see if you want to update once a year or so. There is a check box to turn off the update notification, on the update notification itself.
- calibre downloads currently use [about 150TB of bandwidth a month](#)⁷⁵. Implementing automatic updates would greatly increase that and end up costing thousands of dollars a month, which someone has to pay.
- If I implement a dialog that downloads the update and launches it, instead of going to the website as it does now, that would save the most ardent calibre updater, *at most five clicks a week*. There are far higher priority things to do in calibre development.
- If you really, really hate downloading calibre every week but still want to be up to the latest, I encourage you to run from source, which makes updating trivial. Instructions are [available here](#) (strona 345).

⁷³ <https://calibre-ebook.com/pl/get-involved>

⁷⁴ <https://calibre-ebook.com/pl/bugs>

⁷⁵ <https://calibre-ebook.com/dynamic/downloads>

- There are third party automatic updaters for calibre made by calibre users in the [calibre forum](#)⁷⁶.

9.4.20 How is calibre licensed?

calibre is licensed under the GNU General Public License v3 (an open source license). This means that you are free to redistribute calibre as long as you make the source code available. So if you want to put calibre on a CD with your product, you must also put the calibre source code on the CD. The source code is available [for download](#)⁷⁷. You are free to use the results of conversions from calibre however you want. You cannot use either code or libraries from calibre in your software without making your software open source. For details, see [The GNU GPL v3](#)⁷⁸.

9.4.21 How do I run calibre from my USB stick?

A portable version of calibre is available [here](#)⁷⁹.

9.4.22 How do I run parts of calibre like news download and the Content server on my own Linux server?

First, you must install calibre onto your Linux server. If your server is using a modern Linux distribution, you should have no problems installing calibre onto it.

Informacja: calibre needs GLIBC ≥ 2.31 and libstdc++ $\geq 6.0.28$. If you have an older server, you will either need to compile these from source, or use calibre 3.48 which requires GLIBC ≥ 2.17 or 2.85.1 which requires GLIBC ≥ 2.13 or calibre 1.48 which requires only GLIBC ≥ 2.10 . In addition, although the calibre command line utilities do not need a running X server, some of them do require the X server libraries to be installed on your system. This is because of Qt, which is used for various image processing tasks, and links against these libraries. If you get an ImportError about some Qt modules, you are likely missing some X libraries. Typical candidates are: `libxcb-cursor0`, `libxcb-xinerama0`, `libegl1`, `libopengl0`.

You can run the calibre server via the command:

```
/opt/calibre/calibre-server /path/to/the/library/you/want/to/share
```

You can download news and convert it into an e-book with the command:

```
/opt/calibre/ebook-convert "Title of news source.recipe" outfile.epub
```

If you want to generate MOBI, use `outfile.mobi` instead and use `--output-profile kindle`.

You can email downloaded news with the command:

```
/opt/calibre/calibre-smtp
```

I leave figuring out the exact command line as an exercise for the reader.

Finally, you can add downloaded news to the calibre library with:

```
/opt/calibre/calibredb add --with-library /path/to/library outfile.epub
```

⁷⁶ <https://www.mobileread.com/forums/forumdisplay.php?f=238>

⁷⁷ <https://download.calibre-ebook.com>

⁷⁸ <https://www.gnu.org/licenses/gpl.html>

⁷⁹ https://calibre-ebook.com/pl/download_portable

Remember to read the *Linia poleceń* (strona 303) section of the calibre User Manual to learn more about these, and other commands.

Tu znajdziesz samouczki, z których nauczysz się używania zaawansowanych funkcji calibre, takich jak XPath i szablony.

10.1 Zarządzanie grupami książek, np. „gatunek”

Bywa, że ktoś chce poukładać książki w bibliotece w podgrupy, tak jak to jest w przypadku folderów i podfolderów. Najczęściej w ten sposób tworzy się hierarchię gatunków literackich, ale niekoniecznie. Jeden z użytkowników chciał uporządkować podręczniki wg przedmiotu i kursu. Ktoś inny chciał ułożyć książki, dane w prezencie, wg tematu i osoby, która dostała prezent. W tym rozdziale posłużymy się przykładem uporządkowania wg gatunków literackich.

Zanim przejdziemy dalej, należy wskazać, że nie mówimy tu o folderach na dysku. Podgrupy to nie foldery. Książki nie są nigdzie kopiowane, struktura plików biblioteki calibre nie zostanie naruszona. Pokazujemy natomiast sposób na organizowanie i wyświetlanie podgrup książek w bibliotece calibre.

- *Ustawienia* (strona 155)
- *Szukanie* (strona 157)
- *Ograniczenia* (strona 158)
- *Przydatne funkcje szablonów* (strona 158)

Oto ogólne wymagania dotyczące podgrup, takich jak gatunki literackie:

- Podgrupa (tj. gatunek) musi zawierać (wskazywać na) książki, a nie kategorie. To odróżnia podgrupy od kategorii definiowanych przez użytkownika.
- Książka może być w wielu podgrupach (gatunkach). To odróżnia podgrupy od folderów na dyskach.
- Podgrupy (gatunki) muszą tworzyć hierarchie; podgrupy mogą zawierać inne podgrupy.

Tags give you the first two. If you tag a book with the genre then you can use the Tag browser (or search) to find the books with that genre, giving you the first. Many books can have the same tag(s), giving you the second. The problem is that tags don't satisfy the third requirement. They don't provide a hierarchy.



The calibre hierarchy feature gives you the third – the ability to see the genres in a «tree» and the ability to easily search for books in genre or sub-genre. For example, assume that your genre structure is similar to the following:

```
Genre
. History
.. Japanese
.. Military
.. Roman
. Mysteries
.. English
.. Vampire
. Science Fiction
.. Alternate History
.. Military
.. Space Opera
. Thrillers
.. Crime
.. Horror
etc.
```

By using the hierarchy feature, you can see these genres in the Tag browser in tree form, as shown in the screen image. In this example the outermost level (Genre) is a custom column that contains the genres. Genres containing sub-genres appear with a small triangle next to them. Clicking on that triangle will open the item and show the sub-genres, as you can see with History and Science Fiction.

Kliknięcie gatunku spowoduje wyszukanie wszystkich książek przypisanych do niego oraz wszystkich podporządkowanych mu podgatunków. Na przykład kliknięcie Science Fiction da w efekcie wszystkie trzy podgatunki - Alternate History, Military i Space Opera. Kliknięcie Alternate History wyszuka książki z tego gatunku, ale pominie Military i Space Opera. Rzecz jasna książka może być przypisana do kilku gatunków. Jeżeli należy np. do Space Opera i Military, zostanie odnaleziona po kliknięciu któregośkolwiek z tych gatunków. Wyszukiwanie jest dokładniej omówione poniżej.

Zwróć uwagę, że na rysunku gatunek Military występuje dwukrotnie, raz pod Historią, a raz pod Science Fiction. Ponieważ mamy do czynienia z hierarchią, są to dwa oddzielne gatunki. Książka może być przypisana do jednego, do drugiego, albo

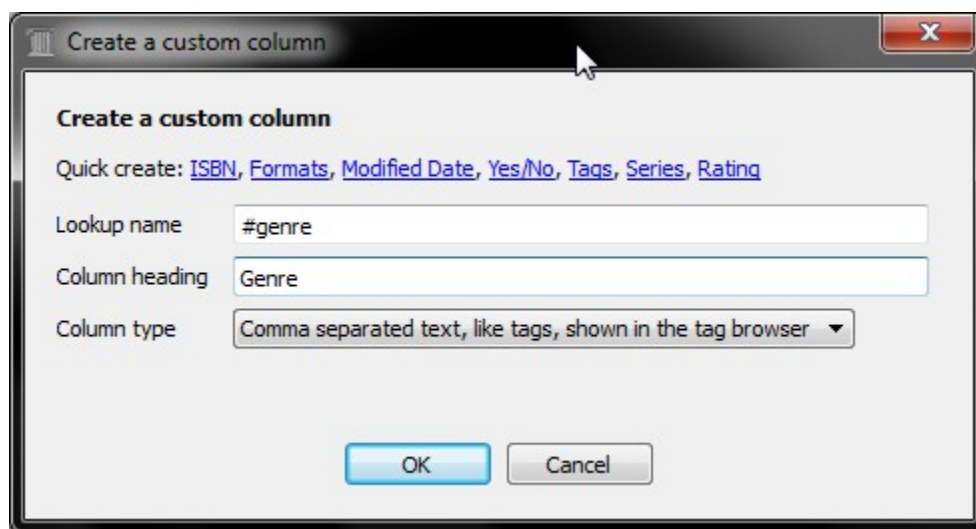
(co w tym przypadku jest raczej wątpliwe) do obu. Na przykład książka Winstona Churchilla „Druga Wojna Światowa” mogłaby być w „History.Military”. Książki z serii „Honor Harrington” Davida Webera mogłyby się znaleźć w „Science Fiction.Military” oraz w „Science Fiction.Space Opera.”

Kiedy już gatunek istnieje, czyli jest choć jedna przypisana do niego książka, możesz łatwo dodać do niego inne książki z biblioteki, przeciągając je z listy i upuszczając na niego. Przyporządkować książki do gatunków możesz również w edytorze metadanych. Więcej na ten temat znajdziesz poniżej.

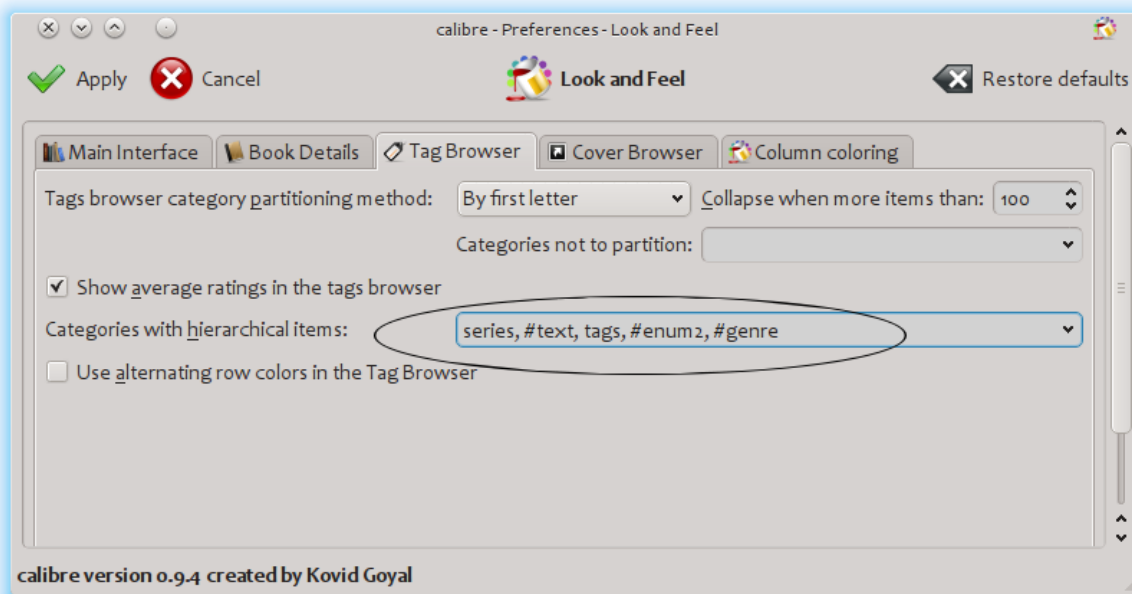
10.1.1 Ustawienia

By now, your question might be „How was all of this setup?” There are three steps: 1) create the custom column, 2) tell calibre that the new column is to be treated as a hierarchy, and 3) add genres.

You create the custom column in the usual way, using Preferences -> Add your own columns. This example uses „#genre” as the lookup name and „Genre” as the column heading. The column type is „Comma-separated text, like tags, shown in the tag browser.”

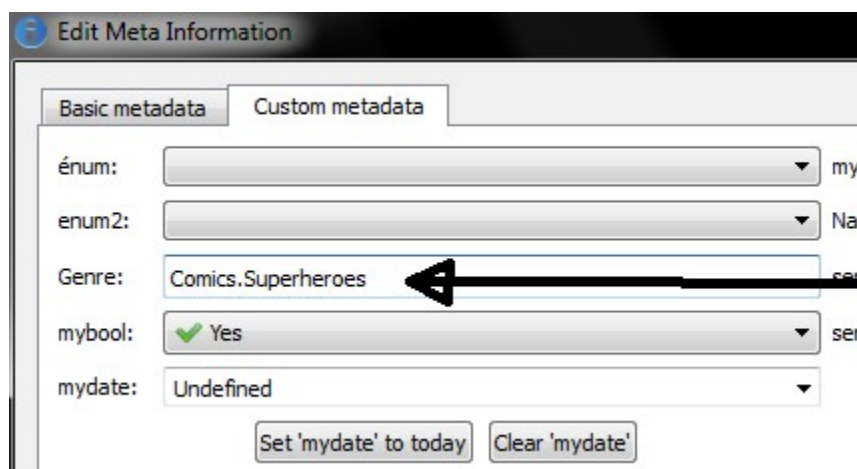


Then after restarting calibre, you must tell calibre that the column is to be treated as a hierarchy. Go to *Preferences* → *Look & feel* → *Tag browser* and enter the lookup name „#genre” into the „Categories with hierarchical items” box. Press *Apply*, and you are done with setting up.



W tej chwili nie ma jeszcze wprowadzonych żadnych gatunków. Został nam jeszcze jeden krok: jak przypisać gatunki do książek. Gatunek nie istnieje w bazie danych, dopóki nie jest do niego przypisana choć jedna książka. Aby nauczyć się je dodawać, musisz się dowiedzieć trochę o tym, czym właściwie jest gatunek w metadanych książki.

Hierarchia jest budowana poprzez tworzenie elementów, składających się z nazw oddzielanych kropkami. Pozostając przy przykładzie podanym wyżej, elementy te to „History.Military”, „Mysteries.Vampire”, „Science Fiction.Space Opera” itp. Tak więc, by stworzyć nowy gatunek, wybierasz książkę, która do niego należy, edytujesz metadane i wprowadzasz nowy gatunek w kolumnie, którą uprzednio dodałeś. I dalej: jeśli chcesz przypisać książkę do gatunku „Comics” i podgatunku „Superheroes”, musisz otworzyć okno edycji metadanych tej książki (komiksu) i na zakładce Własne metadane, w polu oznaczonym „Genre” wprowadzić „Comics.Superheroes”, tak, jak to pokazano poniżej (pomijając inne kolumny).



Po wykonaniu powyższego w przeglądarce tagów zobaczysz:



Od tej chwili, by przypisać książkę do gatunku, możesz albo przeciągnąć ją z listy i upuścić w przeglądarce etykiet na wybrany gatunek, albo edytować metadane dokładnie tak samo, jak przy pierwszej książce.

Informacja: Hierarchical display only works if the Tag browser is set to sort items by name. This is the default and can be checked by clicking the *Configure* button at the bottom of the Tag browser.

10.1.2 Szukanie

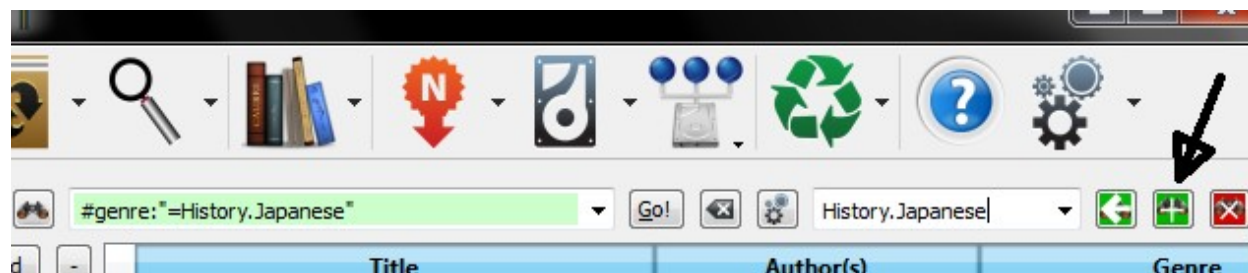


The easiest way to search for genres is using the Tag browser, clicking on the genre you wish to see. Clicking on a genre with children will show you books with that genre and all child genres. However, this might bring up a question. Just because a genre has children doesn't mean that it isn't a genre in its own right. For example, a book can have the genre „History” but not „History.Military”. How do you search for books with only „History”?

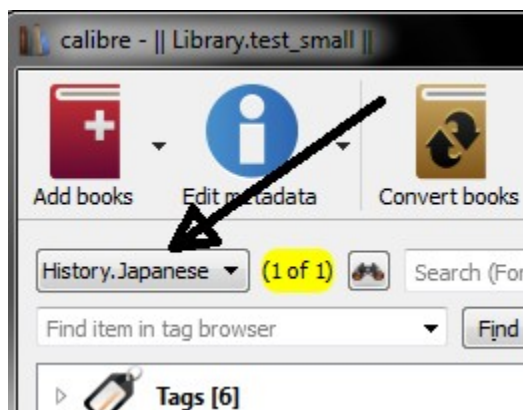
The Tag browser search mechanism knows if an item has children. If it does, clicking on the item cycles through 5 searches instead of the normal three. The first is the normal green plus, which shows you books with that genre only (e.g., History). The second is a doubled plus (shown above), which shows you books with that genre and all sub-genres (e.g., History and History.Military). The third is the normal red minus, which shows you books without that exact genre. The fourth is a doubled minus, which shows you books without that genre or sub-genres. The fifth is back to the beginning, no mark, meaning no search.

10.1.3 Ograniczenia

If you search for a genre then create a saved search for it, you can use the «restrict to» box to create a Virtual library of books with that genre. This is useful if you want to do other searches within the genre or to manage/update metadata for books in the genre. Continuing our example, you can create a Saved search named «History.Japanese» by first clicking on the genre Japanese in the Tag browser to get a search into the search field, entering History.Japanese into the saved search field, then pushing the „Save search” button (the green box with the white plus, on the right-hand side).



Po zapisaniu wyszukiwania możesz go użyć jako ograniczenia.



10.1.4 Przydatne funkcje szablonów

Możesz też użyć informacji o gatunku w szablonie, np. w przypadku zapisywania na dysku lub wysyłania do urządzenia. Pojawia się jednak pytanie: „Jak mogę uzyskać nazwę gatunku literackiego?”. Można tego łatwo dokonać za pomocą funkcji szablonów - subitems.

Przypuśćmy na przykład, że chcesz dodać najwyższy poziom z hierarchii gatunków literackich do szablonu zapisu na dysk, by utworzyć foldery, np. „History/The Gathering Storm - Churchill, Winston”. Aby to zrobić musisz wydzielić pierwszy poziom hierarchii i dodać go na początku wraz z ukośnikiem, by zaznaczyć, że musi być utworzony folder. Szablon powinien wyglądać następująco:

```
{#genre:subitems(0,1)||/}{title} - {authors}
```

Więcej informacji o szablonach i funkcji `subitems()` znajdziesz w rozdziale *Język szablonów* (strona 161)

10.2 Samouczek XPath

W tym samouczku otrzymasz delikatne wprowadzenie do XPath⁸⁰, języka zapytań, którego można użyć do wybrania dowolnych części HTML⁸¹ dokumenty w calibre. XPath jest powszechnie używanym standardem, a wyszukiwanie go w Google przyniesie mnóstwo informacji. Ten samouczek koncentruje się jednak na używaniu XPath do zadań związanych z e-bookami, takich jak znajdowanie nagłówków rozdziałów w nieustrukturyzowanym dokumencie HTML.

Spis treści

- *Wybieranie według nazwy tagu* (strona 159)
- *Wybieranie wg atrybutów* (strona 160)
- *Wybieranie po zawartości znacznika* (strona 160)
- *Przykładowy e-book* (strona 160)
- *Wbudowane funkcje XPath* (strona 161)

10.2.1 Wybieranie według nazwy tagu

Najprostszą formą wyboru jest wyszukanie znacznika wg nazwy. Na przykład przypuśćmy, że chcesz odnaleźć wszystkie znaczniki <h2> w dokumencie. Zapytanie jest proste:

```
//h:h2      (Selects all <h2> tags)
```

Przedrostek // oznacza *wyszukaj na każdym poziomie dokumentu*. A teraz przypuśćmy, że chcesz wybrać tylko te znaczniki , które zawierają znacznik <a>. Można to zrobić tak:

```
//h:a/h:span  (Selects <span> tags inside <a> tags)
```

Jeśli chcesz wyszukać znaczniki na określonym poziomie dokumentu, zmień przedrostek:

```
/h:body/h:div/h:p (Selects <p> tags that are children of <div> tags that are children of the <body> tag)
```

This will match only <p>A very short e-book to demonstrate the use of XPath.</p> in the *Przykładowy e-book* (strona 160) but not any of the other <p> tags. The h: prefix in the above examples is needed to match XHTML tags. This is because internally, calibre represents all content as XHTML. In XHTML tags have a *namespace*, and h: is the namespace prefix for HTML tags.

Teraz przypuśćmy, że chcesz wybrać znaczniki <h1> i <h2>. Do tego potrzebna jest konstrukcja XPath nazywana *predykatem*. *predicate* jest po prostu testem, używanym do wybierania znaczników. Test te mają ogromne możliwości i w dalszej części samouczka zobaczysz co można dzięki nim uzyskać. Predykat jest tworzony poprzez umieszczenie wyrażenia testowego w nawiasach kwadratowych:

```
//*[name()='h1' or name()='h2']
```

Jest kilka nowych elementów w tym wyrażeniu XPath. Pierwszym jest użycie symbolu *. Oznacza on *dopasuj dowolny znacznik*. Teraz popatrzmy na wyrażenie name()='h1' or name()='h2'. *name()* jest przykładem *wbudowanej funkcji*. Zwraca ona po prostu nazwę znacznika. Używając jej możemy więc wybrać znaczniki, których nazwa to *h1*

⁸⁰ <https://en.wikipedia.org/wiki/XPath>

⁸¹ <https://en.wikipedia.org/wiki/HTML>

lub `h2`. Funkcja `name()` nie uwzględnia przestrzeni nazw, nie ma więc konieczności użycia przedrostka `h:`. XPath ma wbudowanych wiele przydatnych funkcji. Kilka z nich poznasz w tym samouczku.

10.2.2 Wybieranie wg atrybutów

Aby wybrać znaczniki wg ich atrybutów wymagane jest użycie predykatów:

```
//*[@style]           (Select all tags that have a style attribute)
//*[class="chapter"]  (Select all tags that have class="chapter")
//h:h1[@class="bookTitle"] (Select all h1 tags that have class="bookTitle")
```

W tym przypadku operator `@` odnosi się do atrybutów znacznika. Można ponadto użyć niektórych *funkcji wbudowanych XPath* (strona 161), aby dokładniej dopasować wartości atrybutów.

10.2.3 Wybieranie po zawartości znacznika

Przy użyciu XPath możesz nawet wybrać znaczniki, bazując na tekście, który zawierają. Najlepszym na to sposobem jest użycie *wyrażeń regularnych* poprzez wbudowaną funkcję „`term:re:test()`”:

```
//h:h2[re:test(., 'chapter|section', 'i')] (Selects <h2> tags that contain the words
↪chapter or
section)
```

W tym przypadku operator `.` odnosi się do zawartości znacznika, tak jak operator `@` odnosił się do atrybutów.

10.2.4 Przykładowy e-book

```
<html>
  <head>
    <title>A very short e-book</title>
    <meta name="charset" value="utf-8" />
  </head>
  <body>
    <h1 class="bookTitle">A very short e-book</h1>
    <p style="text-align:right">Written by Kovid Goyal</p>
    <div class="introduction">
      <p>A very short e-book to demonstrate the use of XPath.</p>
    </div>

    <h2 class="chapter">Chapter One</h2>
    <p>This is a truly fascinating chapter.</p>

    <h2 class="chapter">Chapter Two</h2>
    <p>A worthy continuation of a fine tradition.</p>
  </body>
</html>
```

10.2.5 Wbudowane funkcje XPath

name()

Nazwa aktualnie wybranego znacznika

contains()

`contains(s1, s2)` zwraca *true* jeśli *s1* zawiera *s2*.

re:test()

`re:test(src, pattern, flags)` returns *true* if the string *src* matches the regular expression *pattern*. A particularly useful flag is *i*, it makes matching case insensitive. A good primer on the syntax for regular expressions can be found at [regex syntax](https://docs.python.org/library/re.html)⁸²

10.3 Język szablonów calibre

The calibre template language is a calibre-specific language used throughout calibre for tasks such as specifying file paths, formatting values, and computing the value for user-specified columns. Examples:

- Specify the folder structure and file names when saving files from the calibre library to the disk or e-book reader.
- Define rules for adding icons and colors to the calibre book list.
- Define *virtual columns* that contain data from other columns.
- Advanced library searching.
- Advanced metadata search and replace.

The language is built around the notion of a *template*, which specifies which book metadata to use, computations on that metadata, and how it is to be formatted.

10.3.1 Basic templates

A basic template consists one or more `template` expressions. A `template` expression consists of text and names in curly brackets (`{ }`) that is replaced by the corresponding metadata from the book being processed. For example, the default template in calibre used for saving books to device has 4 `template` expressions:

```
{author_sort}/{title}/{title} - {authors}
```

For the book „The Foundation” by „Isaac Asimov” the will become:

```
Asimov, Isaac/The Foundation/The Foundation - Isaac Asimov
```

The slashes are not `template` expressions because they are in between in `{ }`. Such text is left where it appears. For example, if the template is:

```
{author_sort} Some Important Text {title}/{title} - {authors}
```

then for „The Foundation” the template produces:

```
Asimov, Isaac Some Important Text The Foundation/The Foundation - Isaac Asimov
```

A `template` expression can access all the metadata available in calibre, including custom columns (columns you create yourself), by using a column's `lookup` name. To find the `lookup` name for a *column* (sometimes called *fields*), hover your mouse over the column header in calibre's book list. `Lookup` names for custom columns always begin with

⁸² <https://docs.python.org/library/re.html>

#. For series type columns there is an additional field named `#lookup name_index` that is the series index for that book in the series. For example, if you have a custom series column named `#myseries`, there will also be a column named `#myseries_index`. The standard series column's index is named `series_index`.

In addition to the standard column based fields, you also can use:

- `{formats}` - A list of formats available in the calibre library for a book
- `{identifiers:select(isbn)}` - The ISBN of the book

If the metadata for the field for a given book is not defined then the field in the template is replaced by the empty string (''). For example, consider the following template:

```
{author_sort}/{series}/{title} {series_index}
```

If Asimov's book „Second Foundation” is in the series „Foundation” then the template produces:

```
Asimov, Isaac/Foundation/Second Foundation 3
```

If a series has not been entered for the book then the template produces:

```
Asimov, Isaac/Second Foundation
```

The template processor automatically removes multiple slashes and leading or trailing spaces.

10.3.2 Zaawansowane formatowanie

In addition to metadata substitution, templates can conditionally include additional text and control how substituted data is formatted.

Conditionally including text

Sometimes you want text to appear in the output only if a field is not empty. A common case is `series` and `series_index` where you want either nothing or the two values separated by a hyphen. calibre handles this case using a special template expression syntax.

For example and using the above Foundation example, assume you want the template to produce *Foundation - 3 - Second Foundation*. This template produces that output:

```
{series} - {series_index} - {title}
```

However, if a book has no series the template will produce *- - the title*, which is probably not what you want. Generally, people want the result be the title without the extraneous hyphens. You can accomplish this using the following template syntax:

```
{field:|prefix_text|suffix_text}
```

This template expression says that if `field` has the value `XXXX` then the result will be *prefix_textXXXXsuffix_text*. If `field` is empty (has no value) then the result will be the empty string (nothing) because the prefix and suffix are ignored. The prefix and suffix can contain blanks.

Do not use subtemplates (`{ ... }`) or functions (see below) in the prefix or the suffix.

Using this syntax, we can solve the above no-series problem with the template:

```
{series}{series_index:| - | - }{title}
```

The hyphens will be included only if the book has a series index, which it has only if it has a series. Continuing the Foundation example again, the template will produce *Foundation - 1 - Second Foundation*.

Notes:

- You must include the colon after the `lookup` name if you are using a prefix or a suffix.
- You must either use either no or both `|` characters. Using one, as in `{field:| - }`, is not allowed.
- It is OK to provide no text for either the prefix or the suffix, such as in `{series:|| - }`. The template `{title:||}` is the same as `{title}`.

Formatting

Suppose you want the `series_index` to be formatted as three digits with leading zeros. This does the trick:

```
{series_index:0>3s} - Three digits with leading zeros
```

For trailing zeros, use:

```
{series_index:0<3s} - Three digits with trailing zeros
```

If you use series indices with fractional values, e.g., 1.1, you might want the decimal points to line up. For example, you might want the indices 1 and 2.5 to appear as 01.00 and 02.50 so that they will sort correctly on a device that does lexical sorting. To do this, use:

```
{series_index:0>5.2f} - Five characters consisting of two digits with leading zeros, a decimal point, then 2 digits after the decimal point.
```

If you want only the first two letters of the data, use:

```
{author_sort:.2} - Only the first two letters of the author sort name
```

Much of the calibre template language formatting comes from Python. For more details on the syntax of these advanced formatting operations see the [Python documentation](https://docs.python.org/3/library/string.html#formatstrings)⁸³.

10.3.3 Using templates to define custom columns

Templates can be used to display information that isn't in calibre metadata, or to display metadata differently from calibre's normal format. For example, you might want to show the `ISBN`, a field that calibre does not display. You can accomplish this creating a custom column with the type *Column built from other columns* (hereafter called *composite columns*) and providing a template to generate the displayed text. The column will display the result of evaluating the template. For example, to display the ISBN, create the column and enter `{identifiers:select(isbn)}` in the template box. To display a column containing the values of two series custom columns, separated by a comma, use `{#series1:||,}{#series2}`.

Composite columns can use any template option, including formatting.

Note: You cannot edit the data displayed in a composite column. Instead you edit the source columns. If you edit a composite column, for example by double-clicking it, calibre will open the template for editing, not the underlying data.

10.3.4 Templates and plugboards

Plugboards are used for changing the metadata written into books during send-to-device and save-to-disk operations. A plugboard permits you to specify a template to provide the data to write into the book's metadata. You can use plugboards to modify the following fields: `authors`, `author_sort`, `language`, `publisher`, `tags`, `title`, `title_sort`. This feature helps people who want to use different metadata in books on devices to solve sorting or display issues.

When you create a plugboard, you specify the format and device for which the plugboard is to be used. A special device is provided, `save_to_disk`, that is used when saving formats (as opposed to sending them to a device). Once you have chosen the format and device, you choose the metadata fields to change, providing templates to supply the new values.

⁸³ <https://docs.python.org/3/library/string.html#formatstrings>

These templates are *connected* to their destination fields, hence the name *plugboards*. You can of course use composite columns in these templates.

Plugboards are quite flexible and can be written in Single Function Mode, Template Program Mode, General Program Mode, or Python Template mode.

When a plugboard might apply (Content server, save to disk, or send to device), calibre searches the defined plugboards to choose the correct one for the given format and device. For example, to find the appropriate plugboard for an EPUB book being sent to an ANDROID device, calibre searches the plugboards using the following search order:

- a plugboard with an exact match on format and device, e.g., EPUB and ANDROID
- a plugboard with an exact match on format and the special any device choice, e.g., EPUB and any device
- a plugboard with the special any format choice and an exact match on device, e.g., any format and ANDROID
- a plugboard with any format and any device

The tags and authors fields have special treatment, because both of these fields can hold more than one item. A book can have many tags and many authors. When you specify that one of these two fields is to be changed, the template's result is examined to see if more than one item is there. For tags, the result is cut apart wherever calibre finds a comma. For example, if the template produces the value `Thriller, Horror`, then the result will be two tags, `Thriller` and `Horror`. There is no way to put a comma in the middle of a tag.

The same thing happens for authors, but using a different character for the cut, a & (ampersand) instead of a comma. For example, if the template produces the value `Blogs, Joe&Posts, Susan`, then the book will end up with two authors, `Blogs, Joe` and `Posts, Susan`. If the template produces the value `Blogs, Joe;Posts, Susan`, then the book will have one author with a rather strange name.

Plugboards affect the metadata written into the book when it is saved to disk or written to the device. Plugboards do not affect the metadata used by `save to disk` and `send to device` to create the file names. Instead, file names are constructed using the templates entered on the appropriate preferences window.

10.3.5 Using functions in templates - Single Function Mode

Suppose you want to display the value of a field in upper case when that field is normally in title case. You can do this using *template functions*. For example, to display the title in upper case use the `uppercase` function, as in `{title:uppercase() }`. To display it in title case, use `{title:titlecase() }`.

Functions go into the format part of the template, after the `:` and before the first `|` or the closing `}` if no prefix/suffix is used. If you have both a format and a function reference, the function comes after a second `:`. Functions return the value of the column specified in the template, suitably modified.

The syntax for using functions is one of:

```
{lookup_name:function(arguments) }
{lookup_name:format:function(arguments) }
{lookup_name:function(arguments) |prefix|suffix}
{lookup_name:format:function(arguments) |prefix|suffix}
```

Function names must always be followed by opening and closing parentheses. Some functions require extra values (arguments), and these go inside the parentheses. Arguments are separated by commas. Literal commas (commas as text, not argument separators) must be preceded by a backslash (`\`). The last (or only) argument cannot contain a textual closing parenthesis.

Functions are evaluated before format specifications and the prefix/suffix. See further down for an example of using both a format and a function.

Important: If you have programming experience, please note that the syntax in *Single Function Mode* is not what you expect. Strings are not quoted and spaces are significant. All arguments are considered to be constants; there are no expressions.

Do not use subtemplates (`{ ... }`) as function arguments. Instead, use *Template Program Mode* (strona 181) and *General Program Mode* (strona 167).

Some functions require regular expressions. In the template language regular expression matching is case-insensitive.

In the function documentation below, the notation `[something]*` means that `something` can be repeated zero or more times. The notation `[something]+` means that the `something` is repeated one or more times (must exist at least one time).

The functions intended for use in Single Function Mode are:

- `capitalize()` – returns the value with the first letter upper case and the rest lower case.
- `contains(pattern, text if match, text if not match)` – checks if the value is matched by the regular expression `pattern`. Returns `text if match` if the pattern matches the value, otherwise returns `text if no match`.
- `count(separator)` – interprets the value as a list of items separated by `separator` and returns the number of items in the list. Most lists use a comma as the separator, but `authors` uses an ampersand (&). Examples: `{tags:count(,)}`, `{authors:count(&)}`. Aliases: `count()`, `list_count()`
- `format_number(template)` – interprets the value as a number and formats that number using a Python formatting template such as `{0:5.2f}` or `{0:,d}` or `${0:5,.2f}`. The formatting template must begin with `{0:` and end with `}` as in the above examples. Exception: you can leave off the leading „0:” and trailing „,” if the format template contains only a format. See the template language and the [Python documentation](#)⁸⁴ for more examples. Returns the empty string if formatting fails.
- `human_readable()` – expects the value to be a number and returns a string representing that number in KB, MB, GB, etc.
- `ifempty(text if empty)` – if the value is not empty then return the value of the field, otherwise return `text if empty`.
- `in_list(separator, [pattern, found_val,]* not_found_val)` – interpret the value as a list of items separated by `separator`, checking the `pattern` against each item in the list. If the `pattern` matches an item then return `found_val`, otherwise return `not_found_val`. The pair `pattern` and `found_value` can be repeated as many times as desired, permitting returning different values depending on the item’s value. The patterns are checked in order, and the first match is returned.
- `language_strings(localize)` – return the [language names](#)⁸⁵ for the [language codes](#)⁸⁶ passed in as the value. Example: `{languages:language_strings()}`. If `localize` is zero, return the strings in English. If `localize` is not zero, return the strings in the language of the current locale. `Lang_codes` is a comma-separated list.
- `list_item(index, separator)` – interpret the value as a list of items separated by `separator`, returning the «index’t’h item. The first item is number zero. The last item has the index -1 as in `list_item(-1, separator)`. If the item is not in the list, then the empty string is returned.
- `lookup([pattern, key,]* else_key)` – The patterns will be checked against the value in order. If a pattern matches then the value of the field named by `key` is returned. If no pattern matches then the value of the field named by `else_key` is returned. See “switch” (below).
- `lowercase()` – returns the value of the field in lower case.

⁸⁴ <https://docs.python.org/3/library/string.html#formatstrings>

⁸⁵ https://www.loc.gov/standards/iso639-2/php/code_list.php

⁸⁶ https://www.loc.gov/standards/iso639-2/php/code_list.php

- `rating_to_stars(use_half_stars)` – Returns the rating as string of star (★) characters. The value must be a number between 0 and 5. Set `use_half_stars` to 1 if you want half star characters for fractional numbers available with custom ratings columns.
- `re(pattern, replacement)` – return the value after applying the regular expression. All instances of `pattern` in the value are replaced with `replacement`. The template language uses case insensitive [Python regular expressions](https://docs.python.org/3/library/re.html)⁸⁷.
- `select(key)` – interpret the value as a comma-separated list of items with each item having the form `id:value` (the calibre identifier format). The function finds the first pair with the `id` equal to `key` and returns the corresponding value. If no `id` matches then the function returns the empty string.
- `shorten(left_chars, middle text, right_chars)` – Return a shortened version of the value, consisting of `left_chars` characters from the beginning of the value, followed by `middle text`, followed by `right_chars` characters from the end of the value. `left_chars` and `right_chars` must be non-negative integers. Example: assume you want to display the title with a length of at most 15 characters in length. One template that does this is `{title:shorten(9,-,5)}`. For a book with the title *Ancient English Laws in the Times of Ivanhoe* the result will be *Ancient E-anhoe*: the first 9 characters of the title, a -, then the last 5 characters. If the value's length is less than `left_chars + right_chars + the length of middle text` then the value will be returned unchanged. For example, the title *The Dome* would not be changed.
- `str_in_list(separator, [string, found_val,]+ not_found_val)` – interpret the value as a list of items separated by `separator` then compare `string` against each value in the list. The `string` is not a regular expression. If `string` is equal to any item (ignoring case) then return the corresponding `found_val`. If `string` contains separators then it is also treated as a list and each subvalue is checked. The `string` and `found_value` pairs can be repeated as many times as desired, permitting returning different values depending on `string`'s value. If none of the strings match then `not_found_value` is returned. The strings are checked in order. The first match is returned.
- `subitems(start_index, end_index)` – This function breaks apart lists of tag-like hierarchical items such as genres. It interprets the value as a comma-separated list of tag-like items, where each item is a period-separated list. It returns a new list made by extracting from each item the components from `start_index` to `end_index`, then merging the results back together. Duplicates are removed. The first subitem in a period-separated list has an index of zero. If an index is negative then it counts from the end of the list. As a special case, an `end_index` of zero is assumed to be the length of the list.

Przykłady:

- Assuming a `#genre` column containing *A.B.C*:
 - * `{#genre:subitems(0,1)}` returns „A”
 - * `{#genre:subitems(0,2)}` returns „A.B”
 - * `{#genre:subitems(1,0)}` returns „B.C”
- Assuming a `#genre` column containing „A.B.C, D.E”:
 - * `{#genre:subitems(0,1)}` returns „A, D”
 - * `{#genre:subitems(0,2)}` returns „A.B, D.E”
- `sublist(start_index, end_index, separator)` – interpret the value as a list of items separated by `separator`, returning a new list made from the items from `start_index` to `end_index`. The first item is number zero. If an index is negative, then it counts from the end of the list. As a special case, an `end_index` of zero is assumed to be the length of the list.

Examples assuming that the tags column (which is comma-separated) contains „A, B ,C”:

- `{tags:sublist(0,1,\,)}` returns „A”

⁸⁷ <https://docs.python.org/3/library/re.html>

- `{tags:sublist(-1,0,\,)} returns „C”`
- `{tags:sublist(0,-1,\,)} returns „A, B”`
- `swap_around_articles(separator)` – returns the value with articles moved to the end. The value can be a list, in which case each item in the list is processed. If the value is a list then you must provide the `separator`. If no `separator` is provided then the value is treated as being a single value, not a list. The *articles* are those used by calibre to generate the `title_sort`.
- `swap_around_comma()` – given a value of the form `B, A`, return `A B`. This is most useful for converting names in LN, FN format to FN LN. If there is no comma in the value then the function returns the value unchanged.
- `switch([pattern, value,]+ else_value)` – for each `pattern, value` pair, checks if the value matches the regular expression `pattern` and if so returns the associated value. If no `pattern` matches, then `else_value` is returned. You can have as many `pattern, value` pairs as you wish. The first match is returned.
- `test(text if not empty, text if empty)` – return `text if not empty` if the value is not empty, otherwise return `text if empty`.
- `titlecase()` – returns the value of the field in title case.
- `transliterate()` – Return a string in a latin alphabet formed by approximating the sound of the words in the source field. For example, if the source field is `Фёдор Михайлович Достоевский` this function returns `Fiodor Mikhailovich Dostoievskii`.
- `uppercase()` – returns the value of the field in upper case.

Using functions and formatting in the same template

Suppose you have an integer custom column `#myint` that you want displayed with leading zeros, as in `003`. One way to do this is to use a format of `0>3s`. However, by default if a number (integer or float) equals zero then the value is displayed as the empty string so zero values will produce the empty string, not `000`. If you want to see `000` values then you use both the format string and the `ifempty` function to change the empty value back to a zero. The template would be:

```
{#myint:0>3s;ifempty(0)}
```

Note that you can use the prefix and suffix as well. If you want the number to appear as `[003]` or `[000]`, then use the template:

```
{#myint:0>3s;ifempty(0)|[]}
```

10.3.6 General Program Mode

General Program Mode (GPM) replaces *template expressions* with a program written in the *template language*. The syntax of the language is defined by the following grammar:

```
program      ::= 'program:' expression_list
expression_list ::= top_expression [ ';' top_expression ]*
top_expression ::= or_expression
or_expression ::= and_expression [ '|' and_expression ]*
and_expression ::= not_expression [ '&&' not_expression ]*
not_expression ::= [ '!' not_expression ]* | concatenate_expr
concatenate_expr ::= compare_expr [ '&' compare_expr ]*
compare_expr    ::= add_sub_expr [ compare_op add_sub_expr ]
compare_op      ::= '==' | '!=' | '>=' | '>' | '<=' | '<' | 'in' | 'inlist' |
                  '==#' | '!=#' | '>=#' | '>#' | '<=#' | '<#'
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

add_sub_expr      ::= times_div_expr [ add_sub_op times_div_expr ]*
add_sub_op        ::= '+' | '-'
times_div_expr    ::= unary_op_expr [ times_div_op unary_op_expr ]*
times_div_op      ::= '*' | '/'
unary_op_expr     ::= [ add_sub_op unary_op_expr ]* | expression
expression        ::= identifier | constant | function | assignment | field_reference |
                    if_expr | for_expr | break_expr | continue_expr |
                    '(' expression_list ')' | function_def
field_reference   ::= '$' [ '$' ] [ '#' ] identifier
identifier        ::= id_start [ id_rest ]*
id_start          ::= letter | underscore
id_rest           ::= id_start | digit
constant          ::= " string " | ' string ' | number
function          ::= identifier '(' expression_list [ ',' expression_list ]* ')'
function_def      ::= 'def' identifier '(' top_expression [ ',' top_expression ]* ')' ':'
                    ↪
                    expression_list 'fed'
assignment        ::= identifier '=' top_expression
if_expr           ::= 'if' condition 'then' expression_list
                    [ elif_expr ] [ 'else' expression_list ] 'fi'
condition         ::= top_expression
elif_expr         ::= 'elif' condition 'then' expression_list elif_expr | ''
for_expr          ::= for_list | for_range
for_list          ::= 'for' identifier 'in' list_expr
                    [ 'separator' separator_expr ] ':' expression_list 'rof'
for_range         ::= 'for' identifier 'in' range_expr ':' expression_list 'rof'
range_expr        ::= 'range' '(' [ start_expr ',' ] stop_expr
                    [ ',' step_expr [ ',' limit_expr ] ] ')'
list_expr         ::= top_expression
break_expr        ::= 'break'
continue_expr     ::= 'continue'
separator_expr    ::= top_expression
start_expr        ::= top_expression
stop_expr         ::= top_expression
step_expr         ::= top_expression
limit_expr        ::= top_expression

```

Notes:

- a top_expression always has a value. The value of an expression_list is the value of the last top_expression in the list. For example, the value of the expression list 1;2; 'foobar';3 is 3.
- In a logical context, any non-empty value is True
- In a logical context, the empty value is False
- Strings and numbers can be used interchangeably. For example, 10 and '10' are the same thing.
- Comments are lines starting with a «#» character. Comments beginning later in a line are not supported.

Operator precedence

The operator precedence (order of evaluation) from highest (evaluated first) to lowest (evaluated last) is:

- Function calls, constants, parenthesized expressions, statement expressions, assignment expressions, field references.
- Unary plus (+) and minus (-). These operators evaluate right to left.

These and all the other arithmetic operators return integers if the expression results in a fractional part equal to zero. For example, if an expression returns 3.0 it is changed to 3.

- Multiply (*) and divide (/). These operators are associative and evaluate left to right. Use parentheses if you want to change the order of evaluation.
- Add (+) and subtract (-). These operators are associative and evaluate left to right.
- Numeric and string comparisons. These operators return '1' if the comparison succeeds, otherwise the empty string (''). Comparisons are not associative: `a < b < c` is a syntax error.
- String concatenation (&). The & operator returns a string formed by concatenating the left-hand and right-hand expressions. Example: `'aaa' & 'bbb'` returns `'aaabbb'`. The operator is associative and evaluates left to right.
- Unary logical not (!). This operator returns '1' if the expression is False (evaluates to the empty string), otherwise ''.
- Logical and (&&). This operator returns «1» if both the left-hand and right-hand expressions are True, or the empty string '' if either is False. It is associative, evaluates left to right, and does [short-circuiting](#)⁸⁸.
- Logical or (||). This operator returns '1' if either the left-hand or right-hand expression is True, or '' if both are False. It is associative, evaluates left to right, and does [short-circuiting](#)⁸⁹. It is an *inclusive or*, returning '1' if both the left- and right-hand expressions are True.

Field references

A `field_reference` evaluates to the value of the metadata field named by lookup name that follows the `$` or `$$`. Using `$` is equivalent to using the `field()` function. Using `$$` is equivalent to using the `raw_field` function. Examples:

```
* $authors ==> field('authors')
* $#genre ==> field('#genre')
* $$pubdate ==> raw_field('pubdate')
* $$#my_int ==> raw_field('#my_int')
```

If expressions

If expressions first evaluate the condition. If the condition is True (a non-empty value) then the `expression_list` in the `then` clause is evaluated. If it is False then if present the `expression_list` in the `elif` or `else` clause is evaluated. The `elif` and `else` parts are optional. The words `if`, `then`, `elif`, `else`, and `fi` are reserved; you cannot use them as identifier names. You can put newlines and white space wherever they make sense. The condition is a `top_expression` not an `expression_list`; semicolons are not allowed. The `expression_lists` are semicolon-separated sequences of `top_expressions`. An if expression returns the result of the last `top_expression` in the evaluated `expression_list`, or the empty string if no expression list was evaluated.

Examples:

```
* program: if field('series') then 'yes' else 'no' fi
* program:
    if field('series') then
        a = 'yes';
        b = 'no'
    else
        a = 'no';
        b = 'yes'
    fi;
    strcat(a, '-', b)
```

Nested if example:

⁸⁸ https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html

⁸⁹ https://chortle.ccsu.edu/java5/Notes/chap40/ch40_2.html

```

program:
  if field('series') then
    if check_yes_no(field('#mybool'), '', '1') then
      'yes'
    else
      'no'
    fi
  else
    'no series'
  fi

```

As said above, an `if` produces a value. This means that all the following are equivalent:

```

* program: if field('series') then 'foo' else 'bar' fi
* program: if field('series') then a = 'foo' else a = 'bar' fi; a
* program: a = if field('series') then 'foo' else 'bar' fi; a

```

As a last example, this program returns the value of the `series` column if the book has a series, otherwise the value of the `title` column:

```

program: field(if field('series') then 'series' else 'title' fi)

```

For expressions

The `for` expression iterates over a list of values, processing them one at a time. The `list_expression` must evaluate either to a metadata field lookup name e.g., `tags` or `#genre`, or to a list of values. The `range()` function (strona 179) (see below) generates a list of numbers. If the result is a valid lookup name then the field's value is fetched and the separator specified for that field type is used. If the result isn't a valid lookup name then it is assumed to be a list of values. The list is assumed to be separated by commas unless the optional keyword `separator` is supplied, in which case the list values must be separated by the result of evaluating the `separator_expr`. A separator cannot be used if the list is generated by `range()`. Each value in the list is assigned to the specified variable then the `expression_list` is evaluated. You can use `break` to jump out of the loop, and `continue` to jump to the beginning of the loop for the next iteration.

Example: This template removes the first hierarchical name for each value in `Genre (#genre)`, constructing a list with the new names:

```

program:
  new_tags = '';
  for i in '#genre':
    j = re(i, '^.??.*$', '\1');
    new_tags = list_union(new_tags, j, ',')
  rof;
  new_tags

```

If the original `Genre` is *History.Military, Science Fiction.Alternate History, ReadMe* then the template returns *Military, Alternate History, ReadMe*. You could use this template in calibre's *Edit metadata in bulk* → *Search & replace* with *Search for* set to `template` to strip off the first level of the hierarchy and assign the resulting value to `Genre`.

Note: the last line in the template, `new_tags`, isn't strictly necessary in this case because `for` returns the value of the last `top_expression` in the expression list. The value of an assignment is the value of its expression, so the value of the `for` statement is what was assigned to `new_tags`.

Function definition

If you have code in a template that repeats then you can put that code into a local function. The `def` keyword starts the definition. It is followed by the function name, the argument list, then the code in the function. The function definition ends with the `fed` keyword.

Arguments are positional. When a function is called the supplied arguments are matched left to right against the defined parameters, with the value of the argument assigned to the parameter. It is an error to provide more arguments than defined parameters. Parameters can have default values, such as `a = 25`. If an argument is not supplied for that parameter then the default value is used, otherwise the parameter is set to the empty string.

The `return` statement can be used in a local function.

A function must be defined before it can be used.

Example: This template computes an approximate duration in years, months, and days from a number of days. The function `to_plural()` formats the computed values. Note that the example also uses the `&` operator:

```
program:
    days = 2112;
    years = floor(days/360);
    months = floor(mod(days, 360)/30);
    days = days - ((years*360) + (months * 30));

    def to_plural(v, str):
        if v == 0 then return ' ' fi;
        return v & ' ' & (if v == 1 then str else str & 's' fi) & ' '
    fed;

    to_plural(years, 'year') & to_plural(months, 'month') & to_plural(days, 'day')
```

Relational operators

Relational operators return `'1'` if the comparison is true, otherwise the empty string (`«»`).

There are two forms of relational operators: string comparisons and numeric comparisons.

String comparisons do case-insensitive string comparison using lexical order. The supported string comparison operators are `==`, `!=`, `<`, `<=`, `>`, `>=`, `in`, and `inlist`. For the `in` operator, the result of the left hand expression is interpreted as a regular expression pattern. The `in` operator is `True` if the value of left-hand regular expression matches the value of the right hand expression. The `inlist` operator is true if the left hand regular expression matches any one of the items in the right hand list where the items in the list are separated by commas. The matches are case-insensitive.

The numeric comparison operators are `==#`, `!=#`, `<#`, `<=#`, `>#`, `>=#`. The left and right expressions must evaluate to numeric values with two exceptions: both the string value „None” (undefined field) and the empty string evaluate to the value zero.

Przykłady:

- program: `field('series') == 'foo'` returns `'1'` if the book's series is `«foo»`, otherwise `' '`.
- program: `'f.o' in field('series')` returns `'1'` if the book's series matches the regular expression `f.o` (e.g., *foo*, *Off Onyx*, etc.), otherwise `' '`.
- program: `'science' inlist field('#genre')` returns `'1'` if any of the book's genres match the regular expression `science`, e.g., *Science*, *History of Science*, *Science Fiction* etc., otherwise `' '`.
- program: `'^science$' inlist field('#genre')` returns `'1'` if any of the book's genres exactly match the regular expression `^science$`, e.g., *Science*. The genres *History of Science* and *Science Fiction* don't match. If there isn't a match then returns `' '`.
- program: `if field('series') != 'foo' then 'bar' else 'mumble' fi` returns `'bar'` if the book's series is not `foo`. Otherwise it returns `'mumble'`.
- program: `if field('series') == 'foo' || field('series') == '1632' then 'yes' else 'no' fi` returns `'yes'` if series is either `'foo'` or `'1632'`, otherwise `'no'`.
- program: `if '^(foo|1632)$' in field('series') then 'yes' else 'no' fi` returns `'yes'` if series is either `'foo'` or `'1632'`, otherwise `'no'`.

- program: `if 11 > 2 then 'yes' else 'no' fi` returns 'no' because the `>` operator does a lexical comparison.
- program: `if 11 ># 2 then 'yes' else 'no' fi` returns 'yes' because the `>#` operator does a numeric comparison.

Additional available functions

The following functions are available in addition to those described in *Single Function Mode* (strona 164).

In *GPM* the functions described in *Single Function Mode* all require an additional first parameter specifying the value to operate upon. All parameters are expression_lists (see the grammar above).

- `add(x [, y]*)` – returns the sum of its arguments. Throws an exception if an argument is not a number. In most cases you can use the `+` operator instead of this function.
- `and(value [, value]*)` – returns the string „1” if all values are not empty, otherwise returns the empty string. You can have as many values as you want. In most cases you can use the `&&` operator instead of this function. One reason not to replace `and` with `&&` is if short-circuiting can change the results because of side effects. For example, `and(a=' ', b=5)` will always do both assignments, where the `&&` operator won't do the second.
- `assign(id, val)` – assigns `val` to `id`, then returns `val`. `id` must be an identifier, not an expression. In most cases you can use the `=` operator instead of this function.
- `approximate_formats()` – return a comma-separated list of formats associated with the book. There is no guarantee that the list is correct, although it probably is. This and other zero-parameter functions can be called in Template Program Mode (see below) using the template `{ : 'approximate_formats()' }`. Note that resulting format names are always uppercase, as in EPUB. The `approximate_formats()` function is significantly faster than the `formats_...` functions discussed below.
- `author_links(val_separator, pair_separator)` – returns a string containing a list of authors and those authors» link values in the form:

```
author1 val_separator author1_link pair_separator author2 val_separator author2_
↪link etc.
```

An author is separated from its link value by the `val_separator` string with no added spaces. `author:linkvalue` pairs are separated by the `pair_separator` string argument with no added spaces. It is up to you to choose separator strings that do not occur in author names or links. An author is included even if the author link is empty.

- `author_sorts(val_separator)` – returns a string containing a list of author's sort values for the authors of the book. The sort is the one in the author metadata information (different from the `author_sort` in books). The returned list has the form `author sort 1 val_separator author sort 2` etc. with no added spaces. The author sort values in this list are in the same order as the authors of the book. If you want spaces around `val_separator` then include them in the `val_separator` string.
- `book_count(query, use_vl)` – returns the count of books found by searching for `query`. If `use_vl` is 0 (zero) then virtual libraries are ignored. This function and its companion `book_values()` are particularly useful in template searches, supporting searches that combine information from many books such as looking for series with only one book. It cannot be used in composite columns unless the `tweak allow_template_database_functions_in_composites` is set to True. It can be used only in the GUI.

For example this template search uses this function and its companion to find all series with only one book:

- 1) Define a stored template (using *Preferences* → *Advanced* → *Template functions*) named `series_only_one_book` (the name is arbitrary). The template is:


```

program:
    vals = globals(vals='');
    if !vals then
        all_series = book_values('series', 'series:true', ',', 0);
        for series in all_series:
            if book_count('series:="' & series & '"', 0) == 1 then
                vals = list_join(',', vals, ',', series, ',')
            fi
        rof;
        set_globals(vals)
    fi;
    str_in_list(vals, ',', $series, 1, '')

```

The first time the template runs (the first book checked) it stores the results of the database lookups in a global template variable named `vals`. These results are used to check subsequent books without redoing the lookups.

2) Use the stored template in a template search:

```
template:"program: series_only_one_book()#@#:n:1"
```

Using a stored template instead of putting the template into the search eliminates problems caused by the requirement to escape quotes in search expressions.

- `book_values(column, query, sep, use_vl)` – returns a list of the unique values contained in the column `column` (a lookup name), separated by `sep`, in the books found by searching for `query`. If `use_vl` is 0 (zero) then virtual libraries are ignored. This function and its companion `book_count()` are particularly useful in template searches, supporting searches that combine information from many books such as looking for series with only one book. It cannot be used in composite columns unless the tweak `allow_template_database_functions_in_composites` is set to `True`. It can be used only in the GUI.
- `booksize()` – returns the value of the calibre «size» field. Returns «» if there are no formats.
- `check_yes_no(field_name, is_undefined, is_false, is_true)` – checks if the value of the yes/no field named by the lookup name `field_name` is one of the values specified by the parameters, returning 'yes' if a match is found otherwise returning the empty string. Set the parameter `is_undefined`, `is_false`, or `is_true` to 1 (the number) to check that condition, otherwise set it to 0. Example:

`check_yes_no("#bool", 1, 0, 1)` returns 'Yes' if the yes/no field `#bool` is either `True` or `undefined` (neither `True` nor `False`).

More than one of `is_undefined`, `is_false`, or `is_true` can be set to 1.

- `ceiling(x)` – returns the smallest integer greater than or equal to `x`. Throws an exception if `x` is not a number.
- `character(character_name)` – returns the character named by `character_name`. For example, `character('newline')` returns a newline character (`'\n'`). The supported character names are `newline`, `return`, `tab`, and `backslash`.
- `cmp(x, y, lt, eq, gt)` – compares `x` and `y` after converting both to numbers. Returns `lt` if `x <# y`, `eq` if `x ==# y`, otherwise `gt`. This function can usually be replaced with one of the numeric compare operators (`==#`, `<#`, `>#`, etc).
- `connected_device_name(storage_location_key)` – if a device is connected then return the device name, otherwise return the empty string. Each storage location on a device has its own device name. The `storage_location_key` names are `'main'`, `'carda'` and `'cardb'`. This function works only in the GUI.

- `connected_device_uuid(storage_location_key)` – if a device is connected then return the device uuid (unique id), otherwise return the empty string. Each storage location on a device has a different uuid. The `storage_location_key` location names are 'main', 'carda' and 'cardb'. This function works only in the GUI.
 - `current_library_name()` – return the last name on the path to the current calibre library.
 - `current_library_path()` – return the full path to the current calibre library.
 - `current_virtual_library_name()` – return the name of the current virtual library if there is one, otherwise the empty string. Library name case is preserved. Example: `program:current_virtual_library_name()`. This function works only in the GUI.
 - `date_arithmetic(date, calc_spec, fmt)` – Calculate a new date from `date` using `calc_spec`. Return the new date formatted according to optional `fmt`: if not supplied then the result will be in ISO format. The `calc_spec` is a string formed by concatenating pairs of `vW` (valueWhat) where `v` is a possibly-negative number and `W` is one of the following letters:
 - s: add `v` seconds to date
 - m: add `v` minutes to date
 - h: add `v` hours to date
 - d: add `v` days to date
 - w: add `v` weeks to date
 - y: add `v` years to date, where a year is 365 days.
- Example: '1s3d-1m' will add 1 second, add 3 days, and subtract 1 minute from `date`.
- `days_between(date1, date2)` – return the number of days between `date1` and `date2`. The number is positive if `date1` is greater than `date2`, otherwise negative. If either `date1` or `date2` are not dates, the function returns the empty string.
 - `divide(x, y)` – returns `x / y`. Throws an exception if either `x` or `y` are not numbers. This function can usually be replaced by the `/` operator.
 - `eval(string)` – evaluates the string as a program, passing the local variables. This permits using the template processor to construct complex results from local variables. In *Template Program Mode* (strona 181), because the `{` and `}` characters are interpreted before the template is evaluated you must use `[[` for the `{` character and `]]` for the `}` character. They are converted automatically. Note also that prefixes and suffixes (the `|prefix|suffix` syntax) cannot be used in the argument to this function when using *Template Program Mode* (strona 181).
 - `extra_file_size(file_name)` – returns the size in bytes of the extra file `file_name` in the book's data/ folder if it exists, otherwise -1. See also the functions `has_extra_files()`, `extra_file_names()` and `extra_file_modtime()`. This function can be used only in the GUI.
 - `extra_file_modtime(file_name, format_string)` – returns the modification time of the extra file `file_name` in the book's data/ folder if it exists, otherwise -1. The modtime is formatted according to `format_string` (see `format_date()` for details). If `format_string` is the empty string, returns the modtime as the floating point number of seconds since the epoch. See also the functions `has_extra_files()`, `extra_file_names()` and `extra_file_size()`. The epoch is OS dependent. This function can be used only in the GUI.
 - `extra_file_names(sep [, pattern])` returns a sep-separated list of extra files in the book's data/ folder. If the optional parameter `pattern`, a regular expression, is supplied then the list is filtered to files that match `pattern`. The pattern match is case insensitive. See also the functions `has_extra_files()`, `extra_file_modtime()` and `extra_file_size()`. This function can be used only in the GUI.
 - `field(lookup_name)` – returns the value of the metadata field with lookup name `lookup_name`.

- `field_exists(field_name)` – checks if a field (column) with the lookup name `field_name` exists, returning '1' if so and the empty string if not.
- `finish_formatting(val, fmt, prefix, suffix)` – apply the format, prefix, and suffix to a value in the same way as done in a template like `{series_index:05.2f| - | - }`. This function is provided to ease conversion of complex single-function- or template-program-mode templates to *GPM* Templates. For example, the following program produces the same output as the above template:

```
program: finish_formatting(field("series_index"), "05.2f", " - ", " - ")
```

Jeszcze jeden przykład: dla szablonu ``{series:re(([\s])([\s]+(s|S),1)){series_index:0>2s| - | - }{title}}`` użyj:

```
program:
  strcat(
    re(field('series'), '([\s])([\s]+(s|S),1)', '\1'),
    finish_formatting(field('series_index'), '0>2s', ' - ', ' - '),
    field('title')
  )
```

- `first_matching_cmp(val, [cmp, result,]* else_result)` – compares `val < cmp` in sequence, returning the associated result for the first comparison that succeeds. Returns `else_result` if no comparison succeeds. Example:

```
i = 10;
first_matching_cmp(i, 5, "small", 10, "middle", 15, "large", "giant")
```

returns "large". The same example with a first value of 16 returns "giant".

- `first_non_empty(value [, value]*)` – returns the first value that is not empty. If all values are empty, then the empty string is returned. You can have as many values as you want.
- `floor(x)` – returns the largest integer less than or equal to `x`. Throws an exception if `x` is not a number.
- `format_date(val, format_string)` – format the value, which must be a date string, using the `format_string`, returning a string. The formatting codes are:
 - `d` : the day as number without a leading zero (1 to 31)
 - `dd` : the day as number with a leading zero (01 to 31)
 - `ddd` : the abbreviated localized day name (e.g. „Mon” to „Sun”).
 - `dddd` : the long localized day name (e.g. „Monday” to „Sunday”).
 - `M` : the month as number without a leading zero (1 to 12).
 - `MM` : the month as number with a leading zero (01 to 12)
 - `MMM` : the abbreviated localized month name (e.g. „Jan” to „Dec”).
 - `MMMM` : the long localized month name (e.g. „January” to „December”).
 - `yy` : the year as two digit number (00 to 99).
 - `yyyy` : the year as four digit number.
 - `h` : the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm)
 - `hh` : the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm)
 - `m` : the minutes without a leading 0 (0 to 59)
 - `mm` : the minutes with a leading 0 (00 to 59)
 - `s` : the seconds without a leading 0 (0 to 59)

- `ss` : the seconds with a leading 0 (00 to 59)
- `ap` : use a 12-hour clock instead of a 24-hour clock, with «ap» replaced by the localized string for am or pm.
- `AP` : use a 12-hour clock instead of a 24-hour clock, with «AP» replaced by the localized string for AM or PM.
- `iso` : the date with time and timezone. Must be the only format present.
- `to_number` : convert the date & time into a floating point number (a *timestamp*)
- `from_number` : convert a floating point number (a *timestamp*) into an `iso` formatted date. If you want a different date format then add the desired formatting string after `from_number` and a colon (:). Example:
`from_number:MMM dd yyyy`

You might get unexpected results if the date you are formatting contains localized month names, which can happen if you changed the date format tweaks to contain MMMM. In this case, instead of using the `field()` function as in:

```
format_date(field('pubdate'), 'yyyy')
```

use the `raw_field()` function as in:

```
format_date(raw_field('pubdate'), 'yyyy')
```

- `format_date_field(field_name, format_string)` – format the value in the field `field_name`, which must be the lookup name of date field, either standard or custom. See `format_date()` for the formatting codes. This function is much faster than `format_date` and should be used when you are formatting the value in a field (column). It can't be used for computed dates or dates in string variables. Examples:

```
format_date_field('pubdate', 'yyyy.MM.dd')
format_date_field('#date_read', 'MMM dd, yyyy')
```

- `formats_modtimes(date_format_string)` – return a comma-separated list of colon-separated items `FMT:DATE` representing modification times for the formats of a book. The `date_format_string` parameter specifies how the date is to be formatted. See the `format_date()` function for details. You can use the `select` function to get the modification time for a specific format. Note that format names are always uppercase, as in EPUB.
- `formats_paths()` – return a comma-separated list of colon-separated items `FMT:PATH` giving the full path to the formats of a book. You can use the `select` function to get the path for a specific format. Note that format names are always uppercase, as in EPUB.
- `formats_sizes()` – return a comma-separated list of colon-separated `FMT:SIZE` items giving the sizes in bytes of the formats of a book. You can use the `select` function to get the size for a specific format. Note that format names are always uppercase, as in EPUB.
- `fractional_part(x)` – returns the value after the decimal point. For example, `fractional_part(3.14)` returns 0.14. Throws an exception if `x` is not a number.
- `get_link(field_name, field_value)` – fetch the link for field `field_name` with value `field_value`. If there is no attached link, return the empty string. Examples:
- The following returns the link attached to the tag `Fiction`:

```
get_link('tags', 'Fiction')
```

- This template makes a list of the links for all the tags associated with a book in the form `value:link, ...:`

```
program:
ans = '';
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

for t in $tags:
    l = get_link('tags', t);
    if l then
        ans = list_join(' ', ans, ',', t & ':' & get_link('tags', t), ',')
    fi
roff;
ans

```

- `has_cover()` – return 'Yes' if the book has a cover, otherwise the empty string.
- `has_extra_files([pattern])` – returns the count of extra files, otherwise «» (the empty string). If the optional parameter `pattern` (a regular expression) is supplied then the list is filtered to files that match `pattern` before the files are counted. The pattern match is case insensitive. See also the functions `extra_file_names()`, `extra_file_size()` and `extra_file_modtime()`. This function can be used only in the GUI.
- `identifier_in_list(val, id_name [, found_val, not_found_val])` – treat `val` as a list of identifiers separated by commas. An identifier has the format `id_name:value`. The `id_name` parameter is the `id_name` text to search for, either `id_name` or `id_name:regexp`. The first case matches if there is any identifier matching that `id_name`. The second case matches if `id_name` matches an identifier and the `regexp` matches the identifier's value. If `found_val` and `not_found_val` are provided then if there is a match then return `found_val`, otherwise return `not_found_val`. If `found_val` and `not_found_val` are not provided then if there is a match then return the `identifier:value` pair, otherwise the empty string ('').
- `is_marked()` – check whether the book is *marked* in calibre. If it is then return the value of the mark, either 'true' (lower case) or a comma-separated list of named marks. Returns '' (the empty string) if the book is not marked. This function works only in the GUI.
- `language_codes(lang_strings)` – return the [language codes](https://www.loc.gov/standards/iso639-2/php/code_list.php)⁹⁰ for the language names passed in `lang_strings`. The strings must be in the language of the current locale. `Lang_strings` is a comma-separated list.
- `list_contains(value, separator, [pattern, found_val,]* not_found_val)` – (Alias of `in_list`) Interpreting the value as a list of items separated by `separator`, evaluate the `pattern` against each value in the list. If the `pattern` matches any value then return `found_val`, otherwise return `not_found_val`. The `pattern` and `found_value` can be repeated as many times as desired, permitting returning different values depending on the search. The patterns are checked in order. The first match is returned. Aliases: `in_list()`, `list_contains()`
- `list_count(value, separator)` – interprets value as a list of items separated by `separator`, returning the count of items in the list. Aliases: `count()`, `list_count()`
- `list_count_matching(list, pattern, separator)` – interprets `list` as a list of items separated by `separator`, returning the number of items in the list that match the regular expression `pattern`. Aliases: `list_count_matching()`, `count_matching()`
- `list_difference(list1, list2, separator)` – return a list made by removing from `list1` any item found in `list2` using a case-insensitive comparison. The items in `list1` and `list2` are separated by `separator`, as are the items in the returned list.
- `list_equals(list1, sep1, list2, sep2, yes_val, no_val)` – return `yes_val` if `list1` and `list2` contain the same items, otherwise return `no_val`. The items are determined by splitting each list using the appropriate separator character (`sep1` or `sep2`). The order of items in the lists is not relevant. The comparison is case-insensitive.
- `list_intersection(list1, list2, separator)` – return a list made by removing from `list1` any item not found in `list2`, using a case-insensitive comparison. The items in `list1` and `list2` are separated by `separator`, as are the items in the returned list.

⁹⁰ https://www.loc.gov/standards/iso639-2/php/code_list.php

- `list_join(with_separator, list1, separator1 [, list2, separator2]*)` – return a list made by joining the items in the source lists (`list1` etc) using `with_separator` between the items in the result list. Items in each source list [`123...`] are separated by the associated `separator` [`123...`]. A list can contain zero values. It can be a field like `publisher` that is single-valued, effectively a one-item list. Duplicates are removed using a case-insensitive comparison. Items are returned in the order they appear in the source lists. If items on lists differ only in letter case then the last is used. All separators can be more than one character.

Przykład:

```
program:
    list_join('#@#', $authors, '&', $tags, ',')
```

You can use `list_join` on the results of previous calls to `list_join` as follows:

```
program:
    a = list_join('#@#', $authors, '&', $tags, ',');
    b = list_join('#@#', a, '@#', $genre, ',', $people, '&', 'some value', ',')
```

You can use expressions to generate a list. For example, assume you want items for authors and `#genre`, but with the genre changed to the word „Genre: „, followed by the first letter of the genre, i.e. the genre „Fiction” becomes „Genre: F”. The following will do that:

```
program:
    list_join('#@#', $authors, '&', list_re($genre, ',', '^(.)*$', 'Genre: \1'),
    ↪ ',')
```

- `list_re(src_list, separator, include_re, opt_replace)` – Construct a list by first separating `src_list` into items using the `separator` character. For each item in the list, check if it matches `include_re`. If it does then add it to the list to be returned. If `opt_replace` is not the empty string then apply the replacement before adding the item to the returned list.
- `list_re_group(src_list, separator, include_re, search_re [, template_for_group]*)` – Like `list_re` except replacements are not optional. It uses `re_group(item, search_re, template ...)` when doing the replacements.
- `list_remove_duplicates(list, separator)` – return a list made by removing duplicate items in `list`. If items differ only in case then the last is returned. The items in `list` are separated by `separator`, as are the items in the returned list.
- `list_sort(list, direction, separator)` – return `list` sorted using a case-insensitive lexical sort. If `direction` is zero, `list` is sorted ascending, otherwise descending. The list items are separated by `separator`, as are the items in the returned list.
- `list_split(list_val, sep, id_prefix)` – splits `list_val` into separate values using `sep`, then assigns the values to local variables named `id_prefix_N` where `N` is the position of the value in the list. The first item has position 0 (zero). The function returns the last element in the list.

Przykład:

```
list_split('one:two:foo', ':', 'var')
```

is equivalent to:

```
var_0 = 'one';
var_1 = 'two';
var_2 = 'foo'
```

- `list_union(list1, list2, separator)` – return a list made by merging the items in `list1` and `list2`, removing duplicate items using a case-insensitive comparison. If items differ in case, the one in `list1` is used. The items in `list1` and `list2` are separated by `separator`, as are the items in the returned list. Aliases: `merge_lists()`, `list_union()`
- `mod(x, y)` – returns the floor of the remainder of `x / y`. Throws an exception if either `x` or `y` is not a number.
- `multiply(x [, y]*)` – returns the product of its arguments. Throws an exception if any argument is not a number. This function can usually be replaced by the `*` operator.
- `not(value)` – returns the string „1” if the value is empty, otherwise returns the empty string. This function can usually be replaced with the unary not (!) operator.
- `ondevice()` – return the string 'Yes' if `ondevice` is set, otherwise return the empty string.
- `or(value [, value]*)` – returns the string '1' if any value is not empty, otherwise returns the empty string. You can have as many values as you want. This function can usually be replaced by the `||` operator. A reason it cannot be replaced is if short-circuiting will change the results because of side effects.
- `print(a [, b]*)` – prints the arguments to standard output. Unless you start calibre from the command line (`calibre-debug -g`), the output will go into a black hole. The `print` function always returns its first argument.
- `range(start, stop, step, limit)` – returns a list of numbers generated by looping over the range specified by the parameters `start`, `stop`, and `step`, with a maximum length of `limit`. The first value produced is «start». Subsequent values `next_v = current_v + step`. The loop continues while `next_v < stop` assuming `step` is positive, otherwise while `next_v > stop`. An empty list is produced if `start` fails the test: `start >= stop` if `step` is positive. The `limit` sets the maximum length of the list and has a default of 1000. The parameters `start`, `step`, and `limit` are optional. Calling `range()` with one argument specifies `stop`. Two arguments specify `start` and `stop`. Three arguments specify `start`, `stop`, and `step`. Four arguments specify `start`, `stop`, `step` and `limit`. Examples:

```
range(5) -> '0, 1, 2, 3, 4'
range(0, 5) -> '0, 1, 2, 3, 4'
range(-1, 5) -> '-1, 0, 1, 2, 3, 4'
range(1, 5) -> '1, 2, 3, 4'
range(1, 5, 2) -> '1, 3'
range(1, 5, 2, 5) -> '1, 3'
range(1, 5, 2, 1) -> error(limit exceeded)
```

- `raw_field(lookup_name [, optional_default])` – returns the metadata field named by `lookup_name` without applying any formatting. It evaluates and returns the optional second argument `optional_default` if the field's value is undefined (None).
- `raw_list(lookup_name, separator)` – returns the metadata list named by `lookup_name` without applying any formatting or sorting, with the items separated by `separator`.
- `re_group(value, pattern [, template_for_group]*)` – return a string made by applying the regular expression pattern to `value` and replacing each matched instance with the value returned by the corresponding template. In *Template Program Mode* (strona 181), like for the `template` and the `eval` functions, you use `[[for { and]]` for `}`.

The following example looks for a series with more than one word and uppercases the first word:

```
program: re_group(field('series'), "(\\S* )(.*)", "{$:uppercase()}", "{$}")'
```

- `round(x)` – returns the nearest integer to `x`. Throws an exception if `x` is not a number.
- `series_sort()` – returns the series sort value.

- `strcat(a [, b]*)` – can take any number of arguments. Returns a string formed by concatenating all the arguments.
- `strcat_max(max, string1 [, prefix2, string2]*)` – Returns a string formed by concatenating the arguments. The returned value is initialized to `string1`. Strings made from `prefix`, `string` pairs are added to the end of the value as long as the resulting string length is less than `max`. Prefixes can be empty. Returns `string1` even if `string1` is longer than `max`. You can pass as many `prefix`, `string` pairs as you wish.
- `strcmp(x, y, lt, eq, gt)` – does a case-insensitive lexical comparison of `x` and `y`. Returns `lt` if `x < y`, `eq` if `x == y`, otherwise `gt`. This function can often be replaced by one of the lexical comparison operators (`==`, `>`, `<`, etc.)
- `strcmpcase(x, y, lt, eq, gt)` – does a case-sensitive lexical comparison of `x` and `y`. Returns `lt` if `x < y`, `eq` if `x == y`, otherwise `gt`.

Note: This is NOT the default behavior used by calibre, for example, in the lexical comparison operators (`==`, `>`, `<`, etc.). This function could cause unexpected results, preferably use `strcmp()` whenever possible.

- `strlen(value)` – Returns the length of the string `value`.
- `substr(str, start, end)` – returns the start'th through the end'th characters of `str`. The first character in `str` is the zero'th character. If `end` is negative, then it indicates that many characters counting from the right. If `end` is zero, then it indicates the last character. For example, `substr('12345', 1, 0)` returns `'2345'`, and `substr('12345', 1, -1)` returns `'234'`.
- `subtract(x, y)` – returns `x - y`. Throws an exception if either `x` or `y` are not numbers. This function can usually be replaced by the `-` operator.
- `switch_if([test_expression, value_expression,]+ else_expression)` – for each `test_expression`, `value_expression` pair, checks if `test_expression` is `True` (non-empty) and if so returns the result of `value_expression`. If no `test_expression` is `True` then the result of `else_expression` is returned. You can have as many `test_expression`, `value_expression` pairs as you want.
- `today()` – return a date+time string for today (now). This value is designed for use in *format_date* or *days_between*, but can be manipulated like any other string. The date is in ISO⁹¹ date/time format.
- `template(x)` – evaluates `x` as a template. The evaluation is done in its own context, meaning that variables are not shared between the caller and the template evaluation.
- `to_hex(val)` – returns the string `val` encoded in hex. This is useful when constructing calibre URLs.
- `urls_from_identifiers(identifiers, sort_results)` – given a comma-separated list of `identifiers`, where an *identifier* is a colon-separated pair of values (`id_name:id_value`), returns a comma-separated list of HTML URLs generated from the identifiers. The list not sorted if `sort_results` is 0 (character or number), otherwise it is sorted alphabetically by the identifier name. The URLs are generated in the same way as the built-in identifiers column when shown in *Book details*.

⁹¹ https://en.wikipedia.org/wiki/ISO_8601

10.3.7 More complex programs in template expressions - Template Program Mode

Template Program Mode (TPM) is a blend of *General Program Mode* (strona 167) and *Single Function Mode* (strona 164). *TPM* differs from *Single Function Mode* in that it permits writing template expressions that refer to other metadata fields, use nested functions, modify variables, and do arithmetic. It differs from *General Program Mode* in that the template is contained between { and } characters and doesn't begin with the word `program:`. The program portion of the template is a *General Program Mode* expression list.

Example: assume you want a template to show the series for a book if it has one, otherwise show the value of a custom field `#genre`. You cannot do this in the *Single Function Mode* (strona 164) because you cannot make reference to another metadata field within a template expression. In *TPM* you can, as the following expression demonstrates:

```
{#series:'ifempty($, field('#genre'))'}
```

Przykład pokazuje kilka rzeczy:

- *TPM* is used if the expression begins with `:` and ends with `}`. Anything else is assumed to be in *Single Function Mode* (strona 164).
- the variable `$` stands for the field named in the template: the expression is operating upon, `#series` in this case.
- functions must be given all their arguments. There is no default value. For example, the standard built-in functions must be given an additional initial parameter indicating the source field.
- white space is ignored and can be used anywhere within the expression.
- constant strings are enclosed in matching quotes, either `'` or `"`.

All the functions listed under *Single Function Mode* and *General Program Mode* can be used in *TPM*.

In *TPM*, using { and } characters in string literals can lead to errors or unexpected results because they confuse the template processor. It tries to treat them as template expression boundaries, not characters. In some but not all cases you can replace a { with `[[` and a } with `]]`. Generally, if your program contains { and } characters then you should use *General Program Mode*.

As with *General Program Mode*, for functions documented under *Single Function Mode* (strona 164) you must supply the value the function is to act upon as the first parameter in addition to the documented parameters. In *TPM* you can use `$` to access the value specified by the `lookup` name for the template expression.

10.3.8 Python Template Mode

Python Template Mode (PTM) lets you write templates using native python and the [calibre API](https://manual.calibre-ebook.com/develop.html#api-documentation-for-various-parts-of-calibre)⁹². The database API will be of most use; further discussion is beyond the scope of this manual. PTM templates are faster and can do more complicated operations but you must know how to write code in python using the calibre API.

A PTM template begins with:

```
python:
def evaluate(book, context):
    # book is a calibre metadata object
    # context is an instance of calibre.utils.formatter.PythonTemplateContext,
    # which currently contains the following attributes:
    # db: a calibre legacy database object.
    # globals: the template global variable dictionary.
    # arguments: is a list of arguments if the template is called by a GPM template,
    # otherwise None.
    # funcs: used to call Built-in/User functions and Stored GPM/Python templates.
```

(ciąg dalszy na następnej stronie)

⁹² <https://manual.calibre-ebook.com/develop.html#api-documentation-for-various-parts-of-calibre>

(kontynuacja poprzedniej strony)

```
# Example: context.funcs.list_re_group()

# your Python code goes here
return 'a string'
```

You can add the above text to your template using the context menu, usually accessed with a right click. The comments are not significant and can be removed. You must use python indenting.

The `context` object supports `str(context)` that returns a string of the context's contents, and `context.attributes` that returns a list of the attribute names in the context.

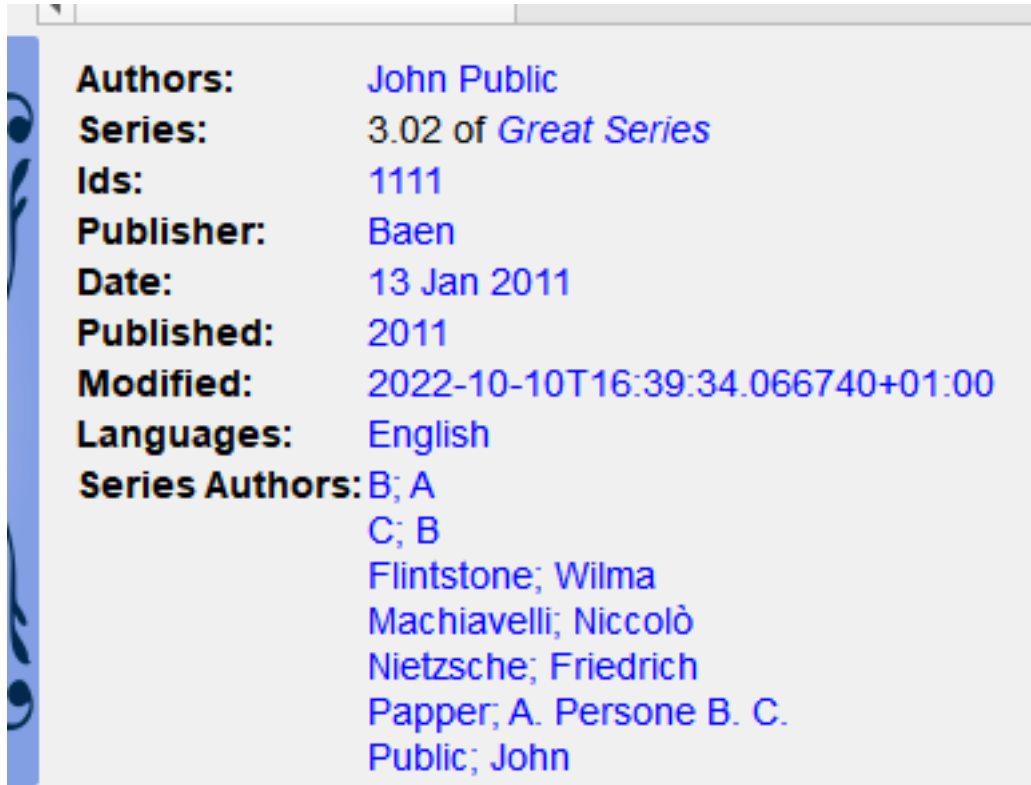
The `context.funcs` attribute allows calling Built-in and User template functions, and Stored GPM/Python templates, so that you can execute them directly in your code. The functions are retrieved using their names. If the name conflicts with a Python keyword, add an underscore to the end of the name. Examples:

```
context.funcs.list_re_group()
context.funcs.assert_()
```

Here is an example of a PTM template that produces a list of all the authors for a series. The list is stored in a *Column built from other columns, behaves like tags*. It shows in *Book details* and has the *on separate lines* checked (in *Preferences* → *Look & feel* → *Book details*). That option requires the list to be comma-separated. To satisfy that requirement the template converts commas in author names to semicolons then builds a comma-separated list of authors. The authors are then sorted, which is why the template uses `author_sort`.

```
python:
def evaluate(book, context):
    if book.series is None:
        return ''
    db = context.db.new_api
    ans = set()
    # Get the list of books in the series
    ids = db.search(f'series="{book.series}"', '')
    if ids:
        # Get all the author_sort values for the books in the series
        author_sorts = (v for v in db.all_field_for('author_sort', ids).values())
        # Add the names to the result set, removing duplicates
        for aus in author_sorts:
            ans.update(v.strip() for v in aus.split('&'))
        # Make a sorted comma-separated string from the result set
    return ', '.join(v.replace(',', ';') for v in sorted(ans))
```

The output in *Book details* looks like this:



10.3.9 Zapisane szablony

Both *General Program Mode* (strona 167) and *Python Template Mode* (strona 181) support saving templates and calling those templates from another template, much like calling stored functions. You save templates using *Preferences* → *Advanced* → *Template functions*. More information is provided in that dialog. You call a template the same way you call a function, passing positional arguments if desired. An argument can be any expression. Examples of calling a template, assuming the stored template is named `foo`:

- `foo()` – call the template passing no arguments.
- `foo(a, b)` call the template passing the values of the two variables `a` and `b`.
- `foo(if field('series') then field('series_index') else 0 fi)` – if the book has a series then pass the `series_index`, otherwise pass the value `0`.

In GPM you retrieve the arguments passed in the call to the stored template using the `arguments` function. It both declares and initializes local variables, effectively parameters. The variables are positional; they get the value of the parameter given in the call in the same position. If the corresponding parameter is not provided in the call then `arguments` assigns that variable the provided default value. If there is no default value then the variable is set to the empty string. For example, the following `arguments` function declares 2 variables, `key`, `alternate`:

```
arguments(key, alternate='series')
```

Examples, again assuming the stored template is named `foo`:

- `foo('#myseries')` – argument `key` is assigned the value `'myseries'` and the argument `alternate` is assigned the default value `'series'`.
- `foo('series', '#genre')` the variable `key` is assigned the value `'series'` and the variable `alternate` is assigned the value `'#genre'`.

- `foo()` – the variable `key` is assigned the empty string and the variable `alternate` is assigned the value `'series'`.

In PTM the arguments are passed in the `arguments` parameter, which is a list of strings. There isn't any way to specify default values. You must check the length of the `arguments` list to be sure that the number of arguments is what you expect.

An easy way to test stored templates is using the `Template tester` dialog. For ease of access give it a keyboard shortcut in *Preferences* → *Advanced* → *Keyboard shortcuts* → *Template tester*. Giving the `Stored templates` dialog a shortcut will help switching more rapidly between the tester and editing the stored template's source code.

10.3.10 Providing additional information to templates

A developer can choose to pass additional information to the template processor, such as application-specific book metadata or information about what the processor is being asked to do. A template can access this information and use it during the evaluation.

Developer: how to pass additional information

The additional information is a Python dictionary containing pairs `variable_name: variable_value` where the values must be strings. The template can access the dictionary, creating template local variables named `variable_name` containing the value `variable_value`. The user cannot change the name so it is best to use names that won't collide with other template local variables, for example by prefixing the name with an underscore.

This dictionary is passed to the template processor (the `formatter`) using the named parameter `global_vars=your_dict`. The full method signature is:

```
def safe_format(self, fmt, kwargs, error_value, book,
                column_name=None, template_cache=None,
                strip_results=True, template_functions=None,
                global_vars={})
```

Template writer: how to access the additional information

You access the additional information (the `globals` dictionary) in a template using the template function:

```
globals(id[=expression] [, id[=expression]]*)
```

where `id` is any legal variable name. This function checks whether the additional information provided by the developer contains the name. If it does then the function assigns the provided value to a template local variable with that name. If the name is not in the additional information and if an `expression` is provided, the `expression` is evaluated and the result is assigned to the local variable. If neither a value nor an expression is provided, the function assigns the empty string (`' '`) to the local variable.

A template can set a value in the `globals` dictionary using the template function:

```
set_globals(id[=expression] [, id[=expression]]*)
```

This function sets the `globals` dictionary key:value pair `id:value` where `value` is the value of the template local variable `id`. If that local variable doesn't exist then `value` is set to the result of evaluating `expression`.

10.3.11 Notes on the difference between modes

The three program modes, *Single Function Mode* (strona 164) (SFM), *Template Program Mode* (strona 181) (TPM), and *General Program Mode* (strona 167) (GPM), work differently. SFM is intended to be «simple» so it hides a lot of programming language bits.

Differences:

- In SFM the value of the column is always passed as an «invisible» first argument to a function included in the template.
- SFM doesn't support the difference between variables and strings; all values are strings.
- The following SFM template returns either the series name or the string „no series”:

```
{series:ifempty(no series)}
```

The equivalent template in *TPM* is

```
{series:'ifempty($, 'no series')'}
```

The equivalent template in *GPM* is:

```
program: ifempty(field('series'), 'no series')
```

The first argument to `ifempty` is the value of the field `series`. The second argument is the string `no series`. In SFM the first argument, the value of the field, is automatically passed (the invisible argument).

- Several template functions, for example `booksize()` and `current_library_name()`, take no arguments. Because of the «invisible argument» you cannot use these functions in SFM.
- Nested functions, where a function calls another function to compute an argument, cannot be used in SFM. For example this template, intended to return the first 5 characters of the series value uppercased, won't work in SFM:

```
{series:uppercase(substr(0,5))}
```

- *TPM* and *GPM* support nested functions. The above template in *TPM* would be:

```
{series:'uppercase(substr($, 0,5))'}
```

In *GPM* it would be:

```
program: uppercase(substr(field('series'), 0,5))
```

- As noted in the above *Template Program Mode* (strona 181) section, using `{` and `}` characters in *TPM* string literals can lead to errors or unexpected results because they confuse the template processor. It tries to treat them as template boundaries, not characters. In some but not all cases you can replace a `{` with `[` and a `}` with `]`. Generally, if your program contains `{` and `}` characters then you should use *General Program Mode*.

10.3.12 User-defined Python template functions

You can add your own Python functions to the template processor. Such functions can be used in any of the three template programming modes. The functions are added by going to *Preferences* → *Advanced* → *Template functions*. Instructions are shown in that dialog.

10.3.13 Special notes for save/send templates

Special processing is applied when a template is used in a *save to disk* or *send to device* template. The values of the fields are cleaned, replacing characters that are special to file systems with underscores, including slashes. This means that field text cannot be used to create folders. However, slashes are not changed in prefix or suffix strings, so slashes in these strings will cause folders to be created. Because of this, you can create variable-depth folder structure.

For example, assume we want the folder structure *series/series_index - title*, with the caveat that if series does not exist, then the title should be in the top folder. The template to do this is:

```
{series:||/}{series_index:|| - }{title}
```

The slash and the hyphen appear only if series is not empty.

The lookup function lets us do even fancier processing. For example, assume that if a book has a series, then we want the folder structure *series/series_index - title.fmt*. If the book does not have a series then we want the folder structure *genre/author_sort/title.fmt*. If the book has no genre then we want to use «Unknown». We want two completely different paths, depending on the value of series.

Aby to osiągnąć potrzebujemy:

1. Create a composite field (give it lookup name #aa) containing `{series}/{series_index} - {title}`. If the series is not empty, then this template will produce *series/series_index - title*.
2. Create a composite field (give it lookup name #bb) containing `{#genre:ifempty(Unknown)}/{author_sort}/{title}`. This template produces *genre/author_sort/title*, where an empty genre is replaced with *Unknown*.
3. Set the save template to `{series:lookup(., #aa, #bb)}`. This template chooses composite field #aa if series is not empty and composite field #bb if series is empty. We therefore have two completely different save paths, depending on whether or not *series* is empty.

10.3.14 Tips

- Use the Template Tester to test templates. Add the tester to the context menu for books in the library and/or give it a keyboard shortcut.
- Templates can use other templates by referencing composite columns built with the desired template. Alternatively, you can use Stored Templates.
- In a plugboard, you can set a field to empty (or whatever is equivalent to empty) by using the special template `{ }`. This template will always evaluate to an empty string.
- The technique described above to show numbers even if they have a zero value works with the standard field *series_index*.

10.3.15 Function reference

Reference for all built-in template language functions

Here, we document all the built-in functions available in the calibre template language. Every function is implemented as a class in python and you can click the source links to see the source code, in case the documentation is insufficient. The functions are arranged in logical groups by type.

- *Arithmetic* (strona 190)
 - *add*(*x* [, *y*]*) (strona 190)
 - *ceiling*(*x*) (strona 191)
 - *divide*(*x*, *y*) (strona 191)
 - *floor*(*x*) (strona 191)
 - *fractional_part*(*x*) (strona 191)
 - *mod*(*x*) (strona 191)
 - *multiply*(*x* [, *y*]*) (strona 191)
 - *odejmuj*(*x*, *y*) (strona 191)
 - *round*(*x*) (strona 192)
- *Boolean* (strona 192)
 - *and*(*value* [, *value*]*) (strona 192)
 - *not*(*value*) (strona 192)
 - *or*(*wartość* [, *wartość*]*) (strona 192)
- *Date functions* (strona 192)
 - *date_arithmetic*(*date*, *calc_spec*, *fmt*) (strona 192)
 - *days_between*(*data1*, *data2*) (strona 193)
 - *today*() (strona 193)
- *Formatting values* (strona 193)
 - *finish_formatting*(*wartość*, *fmt*, *przedrostek*, *przyrostek*) (strona 193)
 - *format_date*(*val*, *format_string*) (strona 193)
 - *format_date_field*(*field_name*, *format_string*) (strona 194)
 - *format_number*(*v*, *template*) (strona 194)
 - *human_readable*(*v*) (strona 194)
- *Get values from metadata* (strona 194)
 - *annotation_count*() (strona 194)
 - *approximate_formats*() (strona 194)
 - *author_links*(*val_separator*, *pair_separator*) (strona 195)
 - *author_sorts*(*val_separator*) (strona 195)
 - *booksize*() (strona 195)

- *connected_device_name(storage_location)* (strona 195)
- *connected_device_uuid(storage_location)* (strona 195)
- *current_library_name()* (strona 195)
- *current_library_path()* (strona 196)
- *current_virtual_library_name()* (strona 196)
- *field(lookup_name)* (strona 196)
- *formats_modtimes(date_format)* (strona 196)
- *formats_paths()* (strona 196)
- *formats_sizes()* (strona 196)
- *has_cover()* (strona 197)
- *is_marked()* (strona 197)
- *language_codes(lang_strings)* (strona 197)
- *ondevice()* (strona 197)
- *series_sort()* (strona 197)
- *user_categories()* (strona 197)
- *If-then-else* (strona 198)
 - *check_yes_no(field_name, is_undefined, is_false, is_true)* (strona 198)
 - *contains(val, pattern, text if match, text if not match)* (strona 198)
 - *field_exists(field_name)* (strona 198)
 - *ifempty(val, tekst jeśli pusty)* (strona 198)
 - *test(val, tekst, jeśli nie jest pusty, tekst, jeśli jest pusty)* (strona 198)
- *Iterating over values* (strona 198)
 - *first_non_empty(value [, value]*)* (strona 198)
 - *lookup(val, [pattern, field,]+ else_field)* (strona 199)
 - *switch(val, [wzorzec, wartość,]+ else_value)* (strona 199)
 - *switch_if([test_expression, value_expression,]+ else_expression)* (strona 199)
- *List lookup* (strona 199)
 - *identifier_in_list(val, id_name [, found_val, not_found_val])* (strona 199)
 - *in_list(val, separator, [pattern, found_val,]+ not_found_val)* (strona 199)
 - *list_item(val, index, separator)* (strona 200)
 - *str_in_list(val, separator, [string, found_val,]+ not_found_val)* (strona 200)
- *List manipulation* (strona 200)
 - *count(val, separator)* (strona 200)
 - *list_count_matching(list, pattern, separator)* (strona 200)
 - *list_difference(list1, list2, separator)* (strona 200)

- *list_equals(list1, sep1, list2, sep2, yes_val, no_val)* (strona 201)
- *list_intersection(list1, list2, separator)* (strona 201)
- *list_join(with_separator, list1, separator1 [, list2, separator2]*)* (strona 201)
- *list_re(src_list, separator, include_re, opt_replace)* (strona 201)
- *list_re_group(src_list, separator, include_re, search_re [, group_template]+)* (strona 202)
- *list_remove_duplicates(list, separator)* (strona 202)
- *list_sort(lista, kolejność, separator)* (strona 202)
- *list_split(list_val, sep, id_prefix)* (strona 202)
- *list_union(list1, list2, separator)* (strona 202)
- *range(start, stop, step, limit)* (strona 202)
- *subitems(val, start_index, end_index)* (strona 203)
- *sublist(val, start_index, end_index, separator)* (strona 203)
- *Other* (strona 203)
 - *arguments(id[=expression] [, id[=expression]]*)* (strona 203)
 - *assign(id, val)* (strona 203)
 - *globals(id[=expression] [, id[=expression]]*)* (strona 204)
 - *print(a[, b]*)* (strona 204)
- *Recursion* (strona 204)
 - *eval(szablon)* (strona 204)
 - *template(x)* (strona 204)
- *Relational* (strona 204)
 - *cmp(x, y, lt, eq, gt)* (strona 204)
 - *first_matching_cmp(val, [cmp1, result1,]+, else_result)* (strona 205)
 - *strcmp(x, y, lt, eq, gt)* (strona 205)
 - *strcmpcase(x, y, lt, eq, gt)* (strona 205)
- *String case changes* (strona 205)
 - *capitalize(val)* (strona 205)
 - *lowercase(val)* (strona 205)
 - *titlecase(val)* (strona 205)
 - *uppercase(val)* (strona 206)
- *String manipulation* (strona 206)
 - *character(character_name)* (strona 206)
 - *re(val, pattern, replacement)* (strona 206)
 - *re_group(val, pattern [, template_for_group]*)* (strona 206)
 - *shorten(val, left chars, middle text, right chars)* (strona 206)

- *strcat_max(max, string1 [, prefix2, string2]*)* (strona 207)
- *strlen(a)* (strona 207)
- *substr(str, startowy, końcowy)* (strona 207)
- *swap_around_articles(val, separator)* (strona 207)
- *swap_around_comma(val)* (strona 207)
- *to_hex(val)* (strona 207)
- *transliterate(a)* (strona 208)
- *Template database functions* (strona 208)
 - *book_count(query, use_vl)* (strona 208)
 - *book_values(column, query, sep, use_vl)* (strona 208)
 - *extra_file_modtime(file_name, format_string)* (strona 208)
 - *extra_file_names(sep [, pattern])* (strona 208)
 - *extra_file_size(file_name)* (strona 208)
 - *get_link(field_name, field_value)* (strona 209)
 - *get_note(field_name, field_value, plain_text)* (strona 209)
 - *has_extra_files([pattern])* (strona 209)
 - *has_note(field_name, field_value)* (strona 209)
- *other* (strona 209)
 - *set_globals(id[=expression] [, id[=expression]])** (strona 209)
- *API of the Metadata objects* (strona 209)

Arithmetic

add(x [, y]*)

class calibre.utils.formatter_functions.BuiltinAdd

add(x [, y]*) – zwraca sumę swoich argumentów. Zgłasza wyjątek, jeśli argument nie jest liczbą. Funkcję tę często można zastąpić operatorem +.

ceiling(x)**class** calibre.utils.formatter_functions.**BuiltinCeiling**

ceiling(x) – returns the smallest integer greater than or equal to x. Throws an exception if x is not a number.

divide(x, y)**class** calibre.utils.formatter_functions.**BuiltinDivide**

divide(x, y) – returns x / y. Throws an exception if either x or y are not numbers. This function can often be replaced with the / operator.

floor(x)**class** calibre.utils.formatter_functions.**BuiltinFloor**

floor(x) – returns the largest integer less than or equal to x. Throws an exception if x is not a number.

fractional_part(x)**class** calibre.utils.formatter_functions.**BuiltinFractionalPart**

fractional_part(x) – returns the value after the decimal point. For example, fractional_part(3.14) returns 0.14. Throws an exception if x is not a number.

mod(x)**class** calibre.utils.formatter_functions.**BuiltinMod**

mod(x) – returns floor(remainder of x / y). Throws an exception if either x or y is not a number.

multiply(x [, y]*)**class** calibre.utils.formatter_functions.**BuiltinMultiply**

multiply(x [, y]*) – zwraca iloczyn swoich argumentów. Zgłasza wyjątek, jeśli dowolny argument nie jest liczbą. Funkcję tę często można zastąpić operatorem *.

odejmuj(x, y)**class** calibre.utils.formatter_functions.**BuiltinSubtract**

odejmuj(x, y) – zwraca x - y. Zgłasza wyjątek, jeśli x lub y nie są liczbami. Funkcję tę często można zastąpić operatorem -.

round(x)

class calibre.utils.formatter_functions.BuiltinRound

round(x) – zwraca najbliższą liczbę całkowitą x. Zgłasza wyjątek, jeśli x nie jest liczbą.

Boolean

and(value [, value]*)

class calibre.utils.formatter_functions.BuiltinAnd

and(value [, value]*) – returns the string „1” if all values are not empty, otherwise returns the empty string. This function works well with test or first_non_empty. You can have as many values as you want. In many cases the && operator can replace this function.

not(value)

class calibre.utils.formatter_functions.BuiltinNot

not(value) – zwraca ciąg „1”, jeśli wartość jest pusta, w przeciwnym razie zwraca pusty ciąg. Ta funkcja działa dobrze z testem lub first_non_empty. W wielu przypadkach! operator może zastąpić tę funkcję.

or(wartość [, wartość]*)

class calibre.utils.formatter_functions.BuiltinOr

or(wartość [, wartość]*) – zwraca ciąg „1”, jeśli jakkolwiek wartość nie jest pusta, w przeciwnym razie zwraca pusty ciąg. Ta funkcja działa dobrze z testem lub first_non_empty. Możesz mieć tyle wartości, ile chcesz. W wielu przypadkach || operator może zastąpić tę funkcję.

Date functions

date_arithmetic(date, calc_spec, fmt)

class calibre.utils.formatter_functions.BuiltinDateArithmetic

date_arithmetic(date, calc_spec, fmt) – Calculate a new date from «date» using «calc_spec». Return the new date formatted according to optional «fmt»: if not supplied then the result will be in iso format. The calc_spec is a string formed by concatenating pairs of «vW» (valueWhat) where «v» is a possibly-negative number and W is one of the following letters: s: add «v» seconds to «date» m: add «v» minutes to «date» h: add «v» hours to «date» d: add «v» days to «date» w: add «v» weeks to «date» y: add «v» years to «date», where a year is 365 days. Example: «1s3d-1m» will add 1 second, add 3 days, and subtract 1 minute from «date».

days_between(data1, data2)

class calibre.utils.formatter_functions.BuiltinDaysBetween

days_between(data1, data2) – Zwraca liczbę dni między data1 a data2. Liczba jest dodatnia jeśli data1 jest większa niż data2, w przeciwnym przypadku ujemna. Jeśli obie daty są nieprawidłowe (ciągi znaków nie są datami) zwracany jest pusty ciąg znaków.

today()

class calibre.utils.formatter_functions.BuiltinToday

today() – zwraca bieżącą datę w postaci tekstu. Wynik jest przeznaczony do użycia w funkcjach format_date lub days_between, ale może być przetwarzany jak każdy inny łańcuch. Data jest w formacie ISO.

Formatting values**finish_formatting(wartość, fmt, przedrostek, przyrostek)**

class calibre.utils.formatter_functions.BuiltinFinishFormatting

finish_formatting(wartość, fmt, przedrostek, przyrostek) – zastosuj format, przedrostek i przyrostek do wartości tak jak to się dzieje w szablonach takich jak {series_index:05.2f}l - l- }. Na przykład następujący program zwróci wynik taki sam jak powyższy szablon: program: finish_formatting(field(„series_index”), „05.2f”, „ - „, „ - „)

format_date(val, format_string)

class calibre.utils.formatter_functions.BuiltinFormatDate

format_date(val, format_string) – format the value, which must be a date, using the format_string, returning a string. The formatting codes are: d : the day as number without a leading zero (1 to 31) dd : the day as number with a leading zero (01 to 31) ddd : the abbreviated localized day name (e.g. „Mon” to „Sun”). dddd : the long localized day name (e.g. „Monday” to „Sunday”). M : the month as number without a leading zero (1 to 12). MM : the month as number with a leading zero (01 to 12) MMM : the abbreviated localized month name (e.g. „Jan” to „Dec”). MMMM : the long localized month name (e.g. „January” to „December”). yy : the year as two digit number (00 to 99). yyyy : the year as four digit number. h : the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm) hh : the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm) m : the minutes without a leading 0 (0 to 59) mm : the minutes with a leading 0 (00 to 59) s : the seconds without a leading 0 (0 to 59) ss : the seconds with a leading 0 (00 to 59) ap : use a 12-hour clock instead of a 24-hour clock, with „ap” replaced by the localized string for am or pm AP : use a 12-hour clock instead of a 24-hour clock, with „AP” replaced by the localized string for AM or PM iso : the date with time and timezone. Must be the only format present to_number: the date as a floating point number from_number[:fmt]: format the timestamp using fmt if present otherwise iso

format_date_field(field_name, format_string)

class calibre.utils.formatter_functions.BuiltinFormatDateField

format_date_field(field_name, format_string) – format the value in the field «field_name», which must be the lookup name of date field, either standard or custom. See «format_date» for the formatting codes. This function is much faster than format_date and should be used when you are formatting the value in a field (column). It can't be used for computed dates or dates in string variables. Example: format_date_field(«pubdate», «yyyy.MM.dd»)

format_number(v, template)

class calibre.utils.formatter_functions.BuiltinFormatNumber

format_number(v, template) – format the number v using a Python formatting template such as „{0:5.2f}” or „{0:d}” or „\${0:5.2f}”. The field_name part of the template must be a 0 (zero) (the „{0:” in the above examples). See the template language and Python documentation for more examples. You can leave off the leading „{0:” and trailing „}” if the template contains only a format. Returns the empty string if formatting fails.

human_readable(v)

class calibre.utils.formatter_functions.BuiltinHumanReadable

human_readable(v) – zwraca ciąg znaków reprezentujący liczbę v w KB, MB, GB itd.

Get values from metadata

annotation_count()

class calibre.utils.formatter_functions.BuiltinAnnotationCount

annotation_count() – return the total number of annotations of all types attached to the current book. This function works only in the GUI.

approximate_formats()

class calibre.utils.formatter_functions.BuiltinApproximateFormats

approximate_formats() – zwraca oddzielną przecinkami listę formatów, które w pewnym momencie były powiązane z książką. Nie ma gwarancji, że jest to prawidłowa lista, chociaż prawdopodobnie jest. Ta funkcja może zostać wywołana w trybie programowym przy użyciu szablonu „{:>approximate_formats()»”. Formaty są zawsze pisane wielkimi literami np. EPUB. Ta funkcja działa wyłącznie w GUI. Jeśli jej wynik ma być użyty przy zapisywaniu na dysk lub wysyłaniu do urządzenia to trzeba stworzyć kolumnę typu „Kolumna składająca się z innych kolumn”, zastosować funkcję w szablonie tej kolumny, a następnie użyć kolumny w szablonie zapisywania/wysyłania

author_links(val_separator, pair_separator)

class calibre.utils.formatter_functions.**BuiltinAuthorLinks**

author_links(val_separator, pair_separator) – zwraca ciąg znaków, zawierający listę autorów i odsyłaczy tych autorów w formie author1 val_separator author1link pair_separator author2 val_separator author2link itp. Autor jest oddzielony od odsyłacza za pomocą val_separator, bez spacji. Pary autor:odsyłacz są oddzielane za pomocą pair_separator, bez spacji. Od ciebie zależy wybranie takiego separatora, który nie występuje ani w nazwiskach autorów, ani w odsyłaczach. Autor jest uwzględniany nawet jeśli odsyłacz jest pusty.

author_sorts(val_separator)

class calibre.utils.formatter_functions.**BuiltinAuthorSorts**

author_sorts(val_separator) – zwraca ciąg znaków, zawierający wartości sortowania dla autorów w danej książce. Sortowanie to dotyczy metadanych autorów (nie jest tym samym, co author_sort w książkach). Lista ma postać autor 1 val_separator autor 2 itp. Autorzy na tej liście są w tej samej kolejności, co w książce. Spacje przed i po separatorze zostaną dodane tylko wtedy, kiedy będą uwzględnione w ciągu znaków val_separator

booksize()

class calibre.utils.formatter_functions.**BuiltinBooksize**

booksize() – zwraca wartość pola «wielkość». Ta funkcja działa wyłącznie w GUI. Jeśli jej wynik ma być użyty przy zapisywaniu na dysk lub wysyłaniu do urządzenia to trzeba stworzyć kolumnę typu „Kolumna składająca się z innych kolumn”, zastosować funkcję w szablonie tej kolumny, a następnie użyć kolumny w szablonie zapisywania/wysyłania

connected_device_name(storage_location)

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceName**

connected_device_name(storage_location) – if a device is connected then return the device name, otherwise return the empty string. Each storage location on a device can have a different name. The location names are «main», «carda» and «cardb». This function works only in the GUI.

connected_device_uuid(storage_location)

class calibre.utils.formatter_functions.**BuiltinConnectedDeviceUUID**

connected_device_uuid(storage_location) – if a device is connected then return the device uuid (unique id), otherwise return the empty string. Each storage location on a device has a different uuid. The location names are «main», «carda» and «cardb». This function works only in the GUI.

current_library_name()

class calibre.utils.formatter_functions.**BuiltinCurrentLibraryName**

current_library_name() – zwraca ostatnią część ścieżki bieżącej biblioteki. Ta funkcja może być używana trybie programowym przy użyciu szablonu „{:>current_library_name()}”.

current_library_path()

class calibre.utils.formatter_functions.BuiltinCurrentLibraryPath

current_library_path() – zwraca ścieżkę do aktualnie używanej biblioteki. Ta funkcja może być wywołana w szablonie w trybie programu za pomocą „{:>current_library_path()>}”.

current_virtual_library_name()

class calibre.utils.formatter_functions.BuiltinCurrentVirtualLibraryName

current_virtual_library_name() – return the name of the current virtual library if there is one, otherwise the empty string. Library name case is preserved. Example: „program: current_virtual_library_name()”.

field(lookup_name)

class calibre.utils.formatter_functions.BuiltinField

field(lookup_name) – returns the metadata field named by lookup_name

formats_modtimes(date_format)

class calibre.utils.formatter_functions.BuiltinFormatsModtimes

formats_modtimes(date_format) – zwraca oddzielną przecinkami listę, której elementy są parami oddzielonymi dwukropkiem i reprezentują datę ostatniej modyfikacji poszczególnych formatów. Argument date_format określa sposób formatowania daty. Informacje o sposobach formatowania daty znajdziesz w opisie funkcji date_format. Aby wybrać datę modyfikacji konkretnego formatu książki można użyć funkcji select. Należy pamiętać, że nazwy formatów są zawsze pisane wielkimi literami, np. EPUB.

formats_paths()

class calibre.utils.formatter_functions.BuiltinFormatsPaths

formats_paths() – zwraca oddzielną przecinkami listę elementów, które składają się z nazwy formatu i ścieżki, oddzielonych średnikami. Dzięki tej funkcji można uzyskać ścieżkę do konkretnego formatu. Nazwy formatów są zawsze pisane wielkimi literami np. EPUB.

formats_sizes()

class calibre.utils.formatter_functions.BuiltinFormatsSizes

formats_sizes() – zwraca oddzielną przecinkami listę, której elementy są parami oddzielonymi dwukropkiem i reprezentują rozmiar poszczególnych formatów książki w bajtach. Możesz użyć funkcji select żeby wybrać konkretny format. Należy pamiętać, że nazwy formatów są zawsze pisane wielkimi literami np. EPUB.

has_cover()

class calibre.utils.formatter_functions.BuiltinHasCover

has_cover() – zwraca Yes jeśli książka ma okładkę, w przeciwnym wypadku pusty łańcuch

is_marked()

class calibre.utils.formatter_functions.BuiltinIsMarked

is_marked() – check whether the book is «marked» in calibre. If it is then return the value of the mark, either «true» or the comma-separated list of named marks. Returns «» if the book is not marked.

language_codes(lang_strings)

class calibre.utils.formatter_functions.BuiltinLanguageCodes

language_codes(lang_strings) – zwraca kody języka dla podanych ciągów. Ciągi muszą być w języku bieżącej lokalizacji. Lang_strings jest listą rozdzielaną przecinkami.

ondevice()

class calibre.utils.formatter_functions.BuiltinOndevice

ondevice() – zwraca Yes jeśli parametr ondevice jest ustawiony, w przeciwnym przypadku pusty ciąg. Ta funkcja działa wyłącznie w GUI. Jeśli jej wynik ma być użyty przy zapisywaniu na dysk lub wysyłaniu do urządzenia to trzeba stworzyć kolumnę typu „Kolumna składająca się z innych kolumn”, zastosować funkcję w szablonie tej kolumny, a następnie użyć kolumny w szablonie zapisywania/wysyłania

series_sort()

class calibre.utils.formatter_functions.BuiltinSeriesSort

series_sort() – zwraca wartość pola sortowania serii

user_categories()

class calibre.utils.formatter_functions.BuiltinUserCategories

user_categories() – zwraca rozdzielaną przecinkami listę kategorii zdefiniowanych przez użytkownika, zawierających daną książkę. Funkcja działa wyłącznie w GUI. Jeśli jej wynik ma być użyty przy zapisywaniu na dysk lub wysyłaniu do urządzenia to trzeba stworzyć kolumnę typu „Kolumna składająca się z innych kolumn”, zastosować funkcję w szablonie tej kolumny, a następnie użyć kolumny w szablonie zapisywania/wysyłania

If-then-else

check_yes_no(field_name, is_undefined, is_false, is_true)

class calibre.utils.formatter_functions.BuiltinCheckYesNo

check_yes_no(field_name, is_undefined, is_false, is_true) – checks the value of the yes/no field named by the lookup key field_name for a value specified by the parameters, returning „yes” if a match is found, otherwise returning an empty string. Set the parameter is_undefined, is_false, or is_true to 1 (the number) to check that condition, otherwise set it to 0. Example: check_yes_no(„#bool”, 1, 0, 1) returns „yes” if the yes/no field „#bool” is either undefined (neither True nor False) or True. More than one of is_undefined, is_false, or is_true can be set to 1. This function is usually used by the test() or is_empty() functions.

contains(val, pattern, text if match, text if not match)

class calibre.utils.formatter_functions.BuiltinContains

contains(val, pattern, text if match, text if not match) – checks if val contains matches for the regular expression pattern. Returns text if match if matches are found, otherwise it returns text if no match

field_exists(field_name)

class calibre.utils.formatter_functions.BuiltinFieldExists

field_exists(field_name) – checks if a field (column) named field_name exists, returning «1» if so and «» if not.

ifempty(val, tekst jeśli pusty)

class calibre.utils.formatter_functions.BuiltinIfempty

ifempty(val, tekst jeśli pusty) – zwraca val jeśli val nie jest pusty, w przeciwnym wypadku zwraca tekst jeśli pusty

test(val, tekst, jeśli nie jest pusty, tekst, jeśli jest pusty)

class calibre.utils.formatter_functions.BuiltinTest

test(val, tekst, jeśli nie jest pusty, tekst, jeśli jest pusty) – zwróć tekst, jeśli nie jest pusty, jeśli val nie jest pusty, w przeciwnym razie zwróć tekst, jeśli jest pusty

Iterating over values

first_non_empty(value [, value]*)

class calibre.utils.formatter_functions.BuiltinFirstNonEmpty

first_non_empty(value [, value]*) – returns the first value that is not empty. If all values are empty, then the empty string is returned. You can have as many values as you want.

lookup(val, [pattern, field,]+ else_field)

class calibre.utils.formatter_functions.**BuiltinLookup**

lookup(val, [pattern, field,]+ else_field) – like switch, except the arguments are field (metadata) names, not text. The value of the appropriate field will be fetched and used. Note that because composite columns are fields, you can use this function in one composite field to use the value of some other composite field. This is extremely useful when constructing variable save paths

switch(val, [wzorzec, wartość,]+ else_value)

class calibre.utils.formatter_functions.**BuiltinSwitch**

switch(val, [wzorzec, wartość,]+ else_value) – dla każdej pary *pattern*, *value* sprawdza, czy *val* pasuje do wyrażenia regularnego *pattern*, a jeśli tak, zwraca tę *value*. Jeśli żaden wzorzec nie pasuje, zwracana jest wartość *else_value*. Możesz mieć dowolną liczbę par *pattern*, *value*

switch_if([test_expression, value_expression,]+ else_expression)

class calibre.utils.formatter_functions.**BuiltinSwitchIf**

switch_if([test_expression, value_expression,]+ else_expression) – dla każdej pary „test_expression, value_expression” sprawdza, czy test_expression ma wartość True (nie jest pusta), a jeśli tak, zwraca wynik value_expression. Jeśli żadne wyrażenie_testowe nie ma wartości True, zwracany jest wynik wyrażenia else_expression. Możesz mieć dowolną liczbę par „test_expression, value_expression”.

List lookup**identifier_in_list(val, id_name [, found_val, not_found_val])**

class calibre.utils.formatter_functions.**BuiltinIdentifierInList**

identifier_in_list(val, id_name [, found_val, not_found_val]) – treat val as a list of identifiers separated by commas. An identifier has the format „id_name:value”. The id_name parameter is the id_name text to search for, either „id_name” or „id_name:regexp”. The first case matches if there is any identifier matching that id_name. The second case matches if id_name matches an identifier and the regexp matches the identifier’s value. If found_val and not_found_val are provided then if there is a match then return found_val, otherwise return not_found_val. If found_val and not_found_val are not provided then if there is a match then return the identifier:value pair, otherwise the empty string.

in_list(val, separator, [pattern, found_val,]+ not_found_val)

class calibre.utils.formatter_functions.**BuiltinInList**

in_list(val, separator, [pattern, found_val,]+ not_found_val) – treating val as a list of items separated by separator, if the pattern matches any of the list values then return found_val. If the pattern matches no list value then return not_found_val. The pattern and found_value pairs can be repeated as many times as desired. The patterns are checked in order. The found_val for the first match is returned. Aliases: in_list(), list_contains()

list_item(val, index, separator)

class calibre.utils.formatter_functions.**BuiltinListitem**

list_item(val, index, separator) – interpretuje wartość jako listę oddzielną *separator*em, zwraca *index* elementu. Pierwszy element to numer zero. Ostatni element może zostać zwrócony przy użyciu *list_item(-1, separator)*. Jeśli element nie jest w liście, wówczas zwrócona zostaje pusta wartość. Separator ma takie samo znaczenie jak w funkcji count.

str_in_list(val, separator, [string, found_val,]+ not_found_val)

class calibre.utils.formatter_functions.**BuiltinStrInList**

str_in_list(val, separator, [string, found_val,]+ not_found_val) – traktując val jako listę elementów oddzielonych separator, jeśli łańcuch pasuje do którejkolwiek z wartości listy, zwróć found_val. Jeśli ciąg nie pasuje do żadnej wartości z listy, zwróć not_found_val. Porównanie jest dokładnym dopasowaniem (nie zawiera) i nie uwzględnia wielkości liter. Pary łańcuch i found_val można powtarzać dowolną liczbę razy. Wzory są sprawdzane w kolejności. Zwracana jest wartość found_val dla pierwszego dopasowania.

List manipulation

count(val, separator)

class calibre.utils.formatter_functions.**BuiltinCount**

count(val, separator) – interprets the value as a list of items separated by *separator*, returning the number of items in the list. Most lists use a comma as the separator, but authors uses an ampersand. Examples: {tags:count(,)}, {authors:count(&)}. Aliases: count(), list_count()

list_count_matching(list, pattern, separator)

class calibre.utils.formatter_functions.**BuiltinListCountMatching**

list_count_matching(list, pattern, separator) – interprets «list» as a list of items separated by «separator», returning the number of items in the list that match the regular expression «pattern». Aliases: list_count_matching(), count_matching()

list_difference(list1, list2, separator)

class calibre.utils.formatter_functions.**BuiltinListDifference**

list_difference(list1, list2, separator) – return a list made by removing from list1 any item found in list2, using a case-insensitive comparison. The items in list1 and list2 are separated by separator, as are the items in the returned list.

list_equals(list1, sep1, list2, sep2, yes_val, no_val)

class calibre.utils.formatter_functions.BuiltinListEquals

list_equals(list1, sep1, list2, sep2, yes_val, no_val) – return yes_val if list1 and list2 contain the same items, otherwise return no_val. The items are determined by splitting each list using the appropriate separator character (sep1 or sep2). The order of items in the lists is not relevant. The comparison is case insensitive.

list_intersection(list1, list2, separator)

class calibre.utils.formatter_functions.BuiltinListIntersection

list_intersection(list1, list2, separator) – return a list made by removing from list1 any item not found in list2, using a case-insensitive comparison. The items in list1 and list2 are separated by separator, as are the items in the returned list.

list_join(with_separator, list1, separator1 [, list2, separator2]*)

class calibre.utils.formatter_functions.BuiltinListJoin

list_join(with_separator, list1, separator1 [, list2, separator2]*) – return a list made by joining the items in the source lists (list1, etc) using with_separator between the items in the result list. Items in each source list[123...] are separated by the associated separator[123...]. A list can contain zero values. It can be a field like publisher that is single-valued, effectively a one-item list. Duplicates are removed using a case-insensitive comparison. Items are returned in the order they appear in the source lists. If items on lists differ only in letter case then the last is used. All separators can be more than one character. Example:

program:

```
list_join(«#@#», $authors, «&», $tags, «,»)
```

Możesz użyć list_join na wynikach poprzednich wywołań list_join w następujący sposób:

program:

```
a = list_join(«#@#», $authors, «&», $tags, «,»); b = list_join(«#@#», a, «#@#», $genre, «,»,  
$people, «&»)
```

Możesz użyć wyrażeń do wygenerowania listy. Załóżmy na przykład, że chcesz mieć pozycje dla autorów i #gatunek, ale po zmianie gatunku na słowo „Genre:” z pierwszą literą gatunku, tj. gatunek „Fiction” staje się „Genre: F”. Zrobią to następujące:

program:

```
list_join(«#@#», $authors, «&», list_re($genre, «,», «^(.)*$», «Genre: 1»), «,»)
```

list_re(src_list, separator, include_re, opt_replace)

class calibre.utils.formatter_functions.BuiltinListRe

list_re(src_list, separator, include_re, opt_replace) – konstruuje listę, dzieląc src_list przy użyciu znaku separator. Dla każdego elementu listy sprawdza, czy pasuje on do include_re. Jeśli tak - dodaje go do zwracanej listy. Jeżeli opt_replace nie jest pustym ciągiem, następuje zamiana przed umieszczeniem elementu na liście.

list_re_group(src_list, separator, include_re, search_re [, group_template]+)

class calibre.utils.formatter_functions.BuiltinListReGroup

list_re_group(src_list, separator, include_re, search_re [, group_template]+) – Like list_re except replacements are not optional. It uses re_group(list_item, search_re, group_template, ...) when doing the replacements on the resulting list.

list_remove_duplicates(list, separator)

class calibre.utils.formatter_functions.BuiltinListRemoveDuplicates

list_remove_duplicates(list, separator) – return a list made by removing duplicate items in the source list. If items differ only in case, the last of them is returned. The items in source list are separated by separator, as are the items in the returned list.

list_sort(lista, kolejność, separator)

class calibre.utils.formatter_functions.BuiltinListSort

list_sort(lista, kolejność, separator) – zwraca posortowaną listę. Wielkość znaków nie ma znaczenia. Jeśli kolejność = 0 sortowanie jest rosnąco, w przeciwnym przypadku malejąco. Elementy listy wynikowej są oddzielane tym samym separatorem co listy źródłowej.

list_split(list_val, sep, id_prefix)

class calibre.utils.formatter_functions.BuiltinListSplit

list_split(list_val, sep, id_prefix) – splits the list_val into separate values using «sep», then assigns the values to variables named «id_prefix_N» where N is the position of the value in the list. The first item has position 0 (zero). The function returns the last element in the list. Example: split(«one:two:foo», «:», «var») is equivalent to var_0 = «one»; var_1 = «two»; var_2 = «foo».

list_union(list1, list2, separator)

class calibre.utils.formatter_functions.BuiltinListUnion

list_union(list1, list2, separator) – return a list made by merging the items in list1 and list2, removing duplicate items using a case-insensitive comparison. If items differ in case, the one in list1 is used. The items in list1 and list2 are separated by separator, as are the items in the returned list. Aliases: list_union(), merge_lists()

range(start, stop, step, limit)

class calibre.utils.formatter_functions.BuiltinRange

range(start, stop, step, limit) – returns a list of numbers generated by looping over the range specified by the parameters start, stop, and step, with a maximum length of limit. The first value produced is «start». Subsequent values next_v are current_v+step. The loop continues while next_v < stop assuming step is positive, otherwise while next_v > stop. An empty list is produced if start fails the test: start>=stop if step is positive. The limit sets the maximum length of the list and has a default of 1000. The parameters start, step, and limit are optional. Calling range() with one argument specifies stop. Two arguments specify start and stop. Three arguments specify start, stop, and step. Four arguments specify start, stop, step and limit. Examples: range(5) -> «0,1,2,3,4». range(0,5) ->

«0,1,2,3,4». range(-1,5) -> «-1,0,1,2,3,4». range(1,5) -> «1,2,3,4». range(1,5,2) -> «1,3». range(1,5,2,5) -> «1,3». range(1,5,2,1) -> error(limit exceeded).

subitems(val, start_index, end_index)

class calibre.utils.formatter_functions.BuiltinSubitems

subitems(val, start_index, end_index) – Ta funkcja ma za zadanie podzielenie listy takiej jak gatunki literackie. Funkcja interpretuje val jako oddzielną przecinkami listę elementów, z których każdy jest oddzielną, oddzielną kropkami listą. Zwraca nową listę utworzoną poprzez odnalezienie wszystkich oddzielanych kropkami elementów, wyodrębnienie z nich elementów od start_index do end_index, a następnie złożenie ich z powrotem. Pierwszy element listy oddzielanej kropkami ma numer zero. Jeśli indeks początkowy jest ujemny to liczenie odbywa się od końca listy. W szczególnym przypadku kiedy end_index jest równy zero zakłada się, że brana jest pod uwagę cała lista. Na przykład, przy użyciu szablonu w trybie podstawowym i zakładając, że #gatunek ma wartość „A.B.C” to {#gatunek:subitems(0,1)} zwróci „A”. {#gatunek:subitems(0,2)} zwróci „A.B”. {#gatunek:subitems(1,0)} zwróci „B.C”. Przy założeniu, że #gatunek ma wartość „A.B.C, D.E.F” {#gatunek:subitems(0,1)} zwróci „A, D”, a {#gatunek:subitems(0,2)} zwróci „A.B, D.E”.

sublist(val, start_index, end_index, separator)

class calibre.utils.formatter_functions.BuiltinSublist

sublist(val, start_index, end_index, separator) – interpret the value as a list of items separated by *separator*, returning a new list made from the *start_index* to the *end_index* item. The first item is number zero. If an index is negative, then it counts from the end of the list. As a special case, an end_index of zero is assumed to be the length of the list. Examples using basic template mode and assuming that the tags column (which is comma-separated) contains „A, B, C”: {tags:sublist(0,1,\,)} returns „A”. {tags:sublist(-1,0,\,)} returns „C”. {tags:sublist(0,-1,\,)} returns „A, B”.

Other

arguments(id[=expression] [, id[=expression]]*)

class calibre.utils.formatter_functions.BuiltinArguments

arguments(id[=expression] [, id[=expression]]*) – Used in a stored template to retrieve the arguments passed in the call. It both declares and initializes local variables, effectively parameters. The variables are positional; they get the value of the parameter given in the call in the same position. If the corresponding parameter is not provided in the call then arguments assigns that variable the provided default value. If there is no default value then the variable is set to the empty string.

assign(id, val)

class calibre.utils.formatter_functions.BuiltinAssign

assign(id, val) – assigns val to id, then returns val. id must be an identifier, not an expression. This function can often be replaced with the = operator.

globals(id[=expression] [, id[=expression]]*)

class calibre.utils.formatter_functions.**BuiltinGlobals**

globals(id[=expression] [, id[=expression]]*) – Retrieves „global variables” that can be passed into the formatter. It both declares and initializes local variables with the names of the global variables passed in. If the corresponding variable is not provided in the passed-in globals then it assigns that variable the provided default value. If there is no default value then the variable is set to the empty string.

print(a[, b]*)

class calibre.utils.formatter_functions.**BuiltinPrint**

print(a[, b]*) – prints the arguments to standard output. Unless you start calibre from the command line (calibre-debug -g), the output will go to a black hole.

Recursion

eval(szablon)

class calibre.utils.formatter_functions.**BuiltinEval**

eval(szablon) – rozwiązuje szablon, podstawiając lokalne zmienne zamiast metadanych książki. Pozwala to na tworzenie nawet skomplikowanych wyników z lokalnych zmiennych. Ponieważ znaki { i } są znakami specjalnymi należy użyć zamiast nich [[i]]. Są one automatycznie konwertowane. Należy pamiętać, że przedrostki i przyrostki (*lpředrostek|przyrostek*) nie mogą być użyte w argumentach funkcji w trybie programowym.

template(x)

class calibre.utils.formatter_functions.**BuiltinTemplate**

template(x) – rozwiązuje x jako szablon. Szablon jest rozwiązywany we własnym kontekście, co oznacza, że zmienne wywołującego i szablonu nie są wspólne. Ponieważ znaki { i } są znakami specjalnymi musisz użyć [[zamiast { i]] zamiast }; są one automatycznie konwertowane. Na przykład template(«[[title_sort]]») zostanie rozwiązany jako szablon {title_sort} i zwróci wartość tego pola. Należy pamiętać, że przedrostki i przyrostki (składnia *lpředrostek|przyrostek*) nie może być użyta w argumentach tej funkcji w trybie programowym.

Relational

cmp(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.**BuiltinCmp**

cmp(x, y, lt, eq, gt) – compares x and y after converting both to numbers. Returns lt if x < y. Returns eq if x == y. Otherwise returns gt. In many cases the numeric comparison operators (>#, <#, ==# etc) can replace this function.

first_matching_cmp(val, [cmp1, result1,]+, else_result)

class calibre.utils.formatter_functions.**BuiltinFirstMatchingCmp**

first_matching_cmp(val, [cmp1, result1,]+, else_result) – compares „val < cmpN” in sequence, returning resultN for the first comparison that succeeds. Returns else_result if no comparison succeeds. Example: first_matching_cmp(10,5,”small”,10,”middle”,15,”large”,,”giant”) returns „large”. The same example with a first value of 16 returns „giant”.

strcmp(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.**BuiltinStrcmp**

strcmp(x, y, lt, eq, gt) – does a case-insensitive comparison of x and y as strings. Returns lt if x < y. Returns eq if x == y. Otherwise returns gt. In many cases the lexical comparison operators (>, <, == etc) can replace this function.

strcmpcase(x, y, lt, eq, gt)

class calibre.utils.formatter_functions.**BuiltinStrcmpcase**

strcmpcase(x, y, lt, eq, gt) – does a case-sensitive comparison of x and y as strings. Returns lt if x < y. Returns eq if x == y. Otherwise returns gt. Note: This is NOT the default behavior used by calibre, for example, in the lexical comparison operators (==, >, <, etc.). This function could cause unexpected results, preferably use strcmp() whenever possible.

String case changes**capitalize(val)**

class calibre.utils.formatter_functions.**BuiltinCapitalize**

capitalize(val) – return val capitalized

lowercase(val)

class calibre.utils.formatter_functions.**BuiltinLowercase**

lowercase(val) – zwraca wartość pola małymi literami

titlecase(val)

class calibre.utils.formatter_functions.**BuiltinTitlecase**

titlecase(val) – zwraca wartość pola jako każdy wyraz z wielkiej litery

uppercase(val)

class calibre.utils.formatter_functions.BuiltinUppercase

uppercase(val) – zwraca wartość pola dużymi literami

String manipulation

character(character_name)

class calibre.utils.formatter_functions.BuiltinCharacter

character(character_name) – returns the character named by character_name. For example, character(«newline») returns a newline character (»n«). The supported character names are «newline», «return», «tab», and «backslash».

re(val, pattern, replacement)

class calibre.utils.formatter_functions.BuiltinRe

re(val, pattern, replacement) – return val after applying the regular expression. All instances of *pattern* are replaced with *replacement*. As in all of calibre, these are Python-compatible regular expressions

re_group(val, pattern [, template_for_group]*)

class calibre.utils.formatter_functions.BuiltinReGroup

re_group(val, pattern [, template_for_group]*) – return a string made by applying the regular expression pattern to the val and replacing each matched instance with the string computed by replacing each matched group by the value returned by the corresponding template. The original matched value for the group is available as \$. In template program mode, like for the template and the eval functions, you use [[for { and]] for }. The following example in template program mode looks for series with more than one word and uppercases the first word: {series:>re_group(\$, „(S*)(.*)\”, „[[\${uppercase()}]]”, „[[\${}]”)>}

shorten(val, left chars, middle text, right chars)

class calibre.utils.formatter_functions.BuiltinShorten

shorten(val, left chars, middle text, right chars) – Return a shortened version of val, consisting of *left chars* characters from the beginning of val, followed by *middle text*, followed by *right chars* characters from the end of the string. *Left chars* and *right chars* must be integers. For example, assume the title of the book is *Ancient English Laws in the Times of Ivanhoe*, and you want it to fit in a space of at most 15 characters. If you use {title:shorten(9,-,5)}, the result will be *Ancient E-anhoe*. If the field's length is less than left chars + right chars + the length of *middle text*, then the field will be used intact. For example, the title *The Dome* would not be changed.

strcat_max(max, string1 [, prefix2, string2]*)**class** calibre.utils.formatter_functions.**BuiltinStrcatMax**

strcat_max(max, string1 [, prefix2, string2]*) – Returns a string formed by concatenating the arguments. The returned value is initialized to string1. *Prefix, string* pairs are added to the end of the value as long as the resulting string length is less than *max*. String1 is returned even if string1 is longer than max. You can pass as many *prefix, string* pairs as you wish.

strlen(a)**class** calibre.utils.formatter_functions.**BuiltinStrlen**

strlen(a) – zwraca długość łańcucha znaków, przekazanego jako parametr

substr(str, startowy, końcowy)**class** calibre.utils.formatter_functions.**BuiltinSubstr**

substr(str, startowy, końcowy) – zwraca startowy poprzez końcowy znak ciągu. Pierwszy znak str to zerowy znak. Jeśli końcowy jest ujemny, wówczas wskazuje tylko znaków licząc od prawej. Jeśli końcowy jest zerem, wówczas wskazuje ostatni znak. Na przykład, substr(«12345», 1, 0) zwraca «2345», a substr(«12345», 1, -1) zwraca «234».

swap_around_articles(val, separator)**class** calibre.utils.formatter_functions.**BuiltinSwapAroundArticles**

swap_around_articles(val, separator) – zwraca wartość z artykułami przesuniętymi na koniec. Wartością może być lista, w którym to przypadku przetwarzany jest każdy element listy. Jeśli wartością jest lista, musisz podać separator wartości listy. Jeśli nie podano separatora, wartość jest traktowana jako pojedyncza wartość, a nie lista.

swap_around_comma(val)**class** calibre.utils.formatter_functions.**BuiltinSwapAroundComma**

swap_around_comma(val) – parametr jest ciągiem znaków w postaci „B, A”, zwracany jest ciąg w postaci „A B”. Przydatne najbardziej przy konwersji nazwisk z postaci LN, FN do FN LN. Jeśli parametr funkcji nie zawiera przecinka jest zwracany bez zmian

to_hex(val)**class** calibre.utils.formatter_functions.**BuiltinToHex**

to_hex(val) – zwraca łańcuch zakodowany w systemie szesnastkowym. Jest to przydatne podczas konstruowania adresów URL calibre.

transliterate(a)

class calibre.utils.formatter_functions.**BuiltinTransliterate**

transliterate(a) – Zwraca ciąg znaków w alfabecie łacińskim utworzony na podstawie przybliżonego brzmienia ciągu źródłowego. Na przykładzie jeśli źródłem jest „Фёдор Михайлович Достоевский” to funkcja zwróci „Fiodor Mikhailovich Dostoevskii”.

Template database functions

book_count(query, use_vl)

class calibre.utils.formatter_functions.**BuiltinBookCount**

book_count(query, use_vl) – returns the count of books found by searching for query. If use_vl is 0 (zero) then virtual libraries are ignored. This function can be used only in the GUI.

book_values(column, query, sep, use_vl)

class calibre.utils.formatter_functions.**BuiltinBookValues**

book_values(column, query, sep, use_vl) – returns a list of the values contained in the column „column”, separated by „sep”, in the books found by searching for „query”. If use_vl is 0 (zero) then virtual libraries are ignored. This function can be used only in the GUI.

extra_file_modtime(file_name, format_string)

class calibre.utils.formatter_functions.**BuiltinExtraFileModtime**

extra_file_modtime(file_name, format_string) – returns the modification time of the extra file «file_name» in the book’s «data/» folder if it exists, otherwise -1.0. The modtime is formatted according to «format_string» (see format_date()). If «format_string» is empty, returns the modtime as the floating point number of seconds since the epoch. The epoch is OS dependent. This function can be used only in the GUI.

extra_file_names(sep [, pattern])

class calibre.utils.formatter_functions.**BuiltinExtraFileNames**

extra_file_names(sep [, pattern]) – returns a sep-separated list of extra files in the book’s «data/» folder. If the optional parameter «pattern», a regular expression, is supplied then the list is filtered to files that match pattern. The pattern match is case insensitive. This function can be used only in the GUI.

extra_file_size(file_name)

class calibre.utils.formatter_functions.**BuiltinExtraFileSize**

extra_file_size(file_name) – returns the size in bytes of the extra file «file_name» in the book’s «data/» folder if it exists, otherwise -1. This function can be used only in the GUI.

get_link(field_name, field_value)

class calibre.utils.formatter_functions.**BuiltinGetLink**

get_link(field_name, field_value) – fetch the link for field «field_name» with value «field_value». If there is no attached link, return «». Example: get_link(«tags», «Fiction») returns the link attached to the tag «Fiction».

get_note(field_name, field_value, plain_text)

class calibre.utils.formatter_functions.**BuiltinGetNote**

get_note(field_name, field_value, plain_text) – fetch the note for field «field_name» with value «field_value». If «plain_text» is empty, return the note's HTML. If «plain_text» is non-empty, return the note's plain text. If the note doesn't exist, return «» in both cases. Example: get_note(«tags», «Fiction», «») returns the HTML of the note attached to the tag «Fiction».

has_extra_files([pattern])

class calibre.utils.formatter_functions.**BuiltinHasExtraFiles**

has_extra_files([pattern]) – returns the count of extra files, otherwise «» (the empty string). If the optional parameter «pattern» (a regular expression) is supplied then the list is filtered to files that match pattern before the files are counted. The pattern match is case insensitive. This function can be used only in the GUI.

has_note(field_name, field_value)

class calibre.utils.formatter_functions.**BuiltinHasNote**

has_note(field_name, field_value) – return «1» if the value «field_value» in the field «field_name» has an attached note, «» otherwise. Example: has_note(«tags», «Fiction») returns «1» if the tag «fiction» has an attached note, «» otherwise.

other**set_globals(id[=expression] [, id[=expression]]*)**

class calibre.utils.formatter_functions.**BuiltinSetGlobals**

set_globals(id[=expression] [, id[=expression]]*) – Sets „global variables” that can be passed into the formatter. The globals are given the name of the id passed in. The value of the id is used unless an expression is provided.

API of the Metadata objects

The python implementation of the template functions is passed in a Metadata object. Knowing it's API is useful if you want to define your own template functions.

class calibre.ebooks.metadata.book.base.**Metadata** (*title, authors=('Nieznany',), other=None, template_cache=None, formatter=None*)

A class representing all the metadata for a book. The various standard metadata fields are available as attributes of this object. You can also stick arbitrary attributes onto this object.

Metadata from custom columns should be accessed via the get() method, passing in the lookup name for the column, for example: „#mytags”.

Use the `is_null()` (strona 210) method to test if a field is null.

This object also has functions to format fields into strings.

The list of standard metadata fields grows with time in `STANDARD_METADATA_FIELDS` (strona 211).

Please keep the method based API of this class to a minimum. Every method becomes a reserved field name.

`is_null (field)`

Return True if the value of field is null in this object. «null» means it is unknown or evaluates to False. So a title of _ («Unknown») is null or a language of «und» is null.

Be careful with numeric fields since this will return True for zero as well as None.

Also returns True if the field does not exist.

`deepcopy (class_generator=<function Metadata.<lambda>>)`

Do not use this method unless you know what you are doing, if you want to create a simple clone of this object, use `deepcopy_metadata()` instead. `Class_generator` must be a function that returns an instance of `Metadata` or a subclass of it.

`get_identifiers ()`

Return a copy of the identifiers dictionary. The dict is small, and the penalty for using a reference where a copy is needed is large. Also, we don't want any manipulations of the returned dict to show up in the book.

`set_identifiers (identifiers)`

Set all identifiers. Note that if you previously set ISBN, calling this method will delete it.

`set_identifier (typ, val)`

If val is empty, deletes identifier of type typ

`standard_field_keys ()`

return a list of all possible keys, even if this book doesn't have them

`custom_field_keys ()`

return a list of the custom fields in this book

`all_field_keys ()`

All field keys known by this instance, even if their value is None

`metadata_for_field (key)`

return metadata describing a standard or custom field.

`all_non_none_fields ()`

Return a dictionary containing all non-None metadata fields, including the custom ones.

`get_standard_metadata (field, make_copy)`

return field metadata from the field if it is there. Otherwise return None. field is the key name, not the label. Return a copy if requested, just in case the user wants to change values in the dict.

`get_all_standard_metadata (make_copy)`

return a dict containing all the standard field metadata associated with the book.

`get_all_user_metadata (make_copy)`

return a dict containing all the custom field metadata associated with the book.

`get_user_metadata (field, make_copy)`

return field metadata from the object if it is there. Otherwise return None. field is the key name, not the label. Return a copy if requested, just in case the user wants to change values in the dict.

set_all_user_metadata (*metadata*)

store custom field metadata into the object. Field is the key name not the label

set_user_metadata (*field, metadata*)

store custom field metadata for one column into the object. Field is the key name not the label

remove_stale_user_metadata (*other_mi*)

Remove user metadata keys (custom column keys) if they don't exist in «other_mi», which must be a metadata object

template_to_attribute (*other, ops*)

Takes a list [(src,dest), (src,dest)], evaluates the template in the context of other, then copies the result to self[dest]. This is on a best-efforts basis. Some assignments can make no sense.

smart_update (*other, replace_metadata=False*)

Merge the information in *other* into self. In case of conflicts, the information in *other* takes precedence, unless the information in *other* is NULL.

format_field (*key, series_with_index=True*)

Returns the tuple (display_name, formatted_value)

to_html ()

A HTML representation of this object.

calibre.ebooks.metadata.book.base.**STANDARD_METADATA_FIELDS**

The set of standard metadata fields.

```
'''
All fields must have a NULL value represented as None for simple types,
an empty list/dictionary for complex types and (None, None) for cover_data
'''

SOCIAL_METADATA_FIELDS = frozenset((
    'tags',          # Ordered list
    'rating',        # A floating point number between 0 and 10
    'comments',      # A simple HTML enabled string
    'series',        # A simple string
    'series_index',  # A floating point number
    # Of the form { scheme1:value1, scheme2:value2}
    # For example: {'isbn':'123456789', 'doi':'xxxx', ... }
    'identifiers',
))

'''
The list of names that convert to identifiers when in get and set.
'''

TOP_LEVEL_IDENTIFIERS = frozenset((
    'isbn',
))

PUBLICATION_METADATA_FIELDS = frozenset((
    'title',          # title must never be None. Should be _('Unknown')
    # Pseudo field that can be set, but if not set is auto generated
    # from title and languages
    'title_sort',
    'authors',        # Ordered list. Must never be None, can be [_('Unknown')]
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

    'author_sort_map', # Map of sort strings for each author
    # Pseudo field that can be set, but if not set is auto generated
    # from authors and languages
    'author_sort',
    'book_producer',
    'timestamp',      # Dates and times must be timezone aware
    'pubdate',
    'last_modified',
    'rights',
    # So far only known publication type is periodical:calibre
    # If None, means book
    'publication_type',
    'uuid',           # A UUID usually of type 4
    'languages',      # ordered list of languages in this publication
    'publisher',      # Simple string, no special semantics
    # Absolute path to image file encoded in filesystem_encoding
    'cover',
    # Of the form (format, data) where format is, e.g. 'jpeg', 'png', 'gif'...
    'cover_data',
    # Either thumbnail data, or an object with the attribute
    # image_path which is the path to an image file, encoded
    # in filesystem_encoding
    'thumbnail',
))

BOOK_STRUCTURE_FIELDS = frozenset((
    # These are used by code, Null values are None.
    'toc', 'spine', 'guide', 'manifest',
))

USER_METADATA_FIELDS = frozenset((
    # A dict of dicts similar to field_metadata. Each field description dict
    # also contains a value field with the key #value#.
    'user_metadata',
))

DEVICE_METADATA_FIELDS = frozenset((
    'device_collections', # Ordered list of strings
    'lpath',              # Unicode, / separated
    'size',                # In bytes
    'mime',                # Mimetype of the book file being represented
))

CALIBRE_METADATA_FIELDS = frozenset((
    'application_id',     # An application id, currently set to the db_id.
    'db_id',              # the calibre primary key of the item.
    'formats',            # list of formats (extensions) for this book
    # a dict of user category names, where the value is a list of item names
    # from the book that are in that category
    'user_categories',
    # a dict of items to associated hyperlink
    'link_maps',
))

ALL_METADATA_FIELDS = SOCIAL_METADATA_FIELDS.union(
    PUBLICATION_METADATA_FIELDS).union(
    BOOK_STRUCTURE_FIELDS).union(

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

        USER_METADATA_FIELDS).union(
        DEVICE_METADATA_FIELDS).union(
        CALIBRE_METADATA_FIELDS)

# All fields except custom fields
STANDARD_METADATA_FIELDS = SOCIAL_METADATA_FIELDS.union(
        PUBLICATION_METADATA_FIELDS).union(
        BOOK_STRUCTURE_FIELDS).union(
        DEVICE_METADATA_FIELDS).union(
        CALIBRE_METADATA_FIELDS)

# Metadata fields that smart update must do special processing to copy.
SC_FIELDS_NOT_COPIED = frozenset(('title', 'title_sort', 'authors',
                                   'author_sort', 'author_sort_map',
                                   'cover_data', 'tags', 'languages',
                                   'identifiers'))

# Metadata fields that smart update should copy only if the source is not None
SC_FIELDS_COPY_NOT_NULL = frozenset(('device_collections', 'lpath', 'size', 'comments'
→, 'thumbnail'))

# Metadata fields that smart update should copy without special handling
SC_COPYABLE_FIELDS = SOCIAL_METADATA_FIELDS.union(
        PUBLICATION_METADATA_FIELDS).union(
        BOOK_STRUCTURE_FIELDS).union(
        DEVICE_METADATA_FIELDS).union(
        CALIBRE_METADATA_FIELDS) - \
        SC_FIELDS_NOT_COPIED.union(
        SC_FIELDS_COPY_NOT_NULL)

SERIALIZABLE_FIELDS = SOCIAL_METADATA_FIELDS.union(
        USER_METADATA_FIELDS).union(
        PUBLICATION_METADATA_FIELDS).union(
        CALIBRE_METADATA_FIELDS).union(
        DEVICE_METADATA_FIELDS) - \
        frozenset(('device_collections', 'formats',
        'cover_data'))

# these are rebuilt when needed

```

10.4 Wszystko, co chcielibyście wiedzieć o wyrażeniach regularnych w calibre

Wyrażenia regularne są stosowane w wielu miejscach w calibre, umożliwiając skomplikowane nawet manipulacje zawartością i metadanymi książki. Ten podręcznik wprowadzi cię łagodnie w podstawy użycia wyrażeń regularnych w calibre.

Spis treści

- *Na początek ostrzeżenie i zachęta.* (strona 214)
- *Gdzie w calibre można użyć wyrażeń regularnych?* (strona 214)
- *Czym właściwie jest wyrażenie regularne?* (strona 215)

- *Możesz to wyjaśnić?* (strona 215)
- *Nie brzmi źle. Co dalej?* (strona 215)
- *Hej, nieźle! To zaczyna mieć sens!* (strona 215)
- *No tak, te znaki specjalne są bardzo fajne i w ogóle, ale co jeśli chcemy dopasować kropkę albo znak zapytania?* (strona 216)
- *A jakie są najbardziej przydatne zestawy?* (strona 216)
- *A gdybym miał kilka różnych ciągów, które chciałbym dopasować, to czy byłoby to skomplikowane?* (strona 217)
- *Zapomniałem...* (strona 217)
- *Na początku powiedziałeś, że w wyrażeniach regularnych wielkość liter może nie mieć znaczenia?* (strona 217)
- *Zaczynam chyba rozumieć, o co chodzi w wyrażeniach regularnych, a teraz... jak ich użyć w calibre?* (strona 218)
 - *Konwertowanie* (strona 218)
 - *Dodawanie książek* (strona 218)
 - *Masowa edycja metadanych* (strona 219)
- *Szybkie odniesienie* (strona 219)
- *Uznanie* (strona 224)

10.4.1 Na początek ostrzeżenie i zachęta.

Niestety, będzie to trochę techniczna część - w końcu wyrażenia regularne są technicznym narzędziem do wykonywania technicznych zadań. Znajdzie się tu trochę żargonu i pojęć, które mogą wydawać się skomplikowane i zawiłe. Postaram się je wyjaśnić jak najlepiej jak mogę, ale, wiercie mi, bez tego się nie da. Mimo wszystko nie zniechęcaj się żargonem, postaram się wyjaśniać wszystkie nowe pojęcia. I choć same wyrażenia regularne mogą wydawać się wiedzą tajemną, czarną magią (albo, bardziej prozaicznie, losowym ciągiem znaków), obiecuję, że nie będzie to aż tak skomplikowane. Nawet ci, którzy rozumieją wyrażenia regularne miewają kłopoty z odczytaniem tych bardziej skomplikowanych, ale pisanie ich nie jest takie trudne - konstruujesz wyrażenie krok po kroku. Tak więc - zrób pierwszy krok i - podążaj za białym królikiem...

10.4.2 Gdzie w calibre można użyć wyrażen regularnych?

There are a few places calibre uses regular expressions. There's the *Search & replace* in conversion options, metadata detection from filenames in the import settings and Search & replace when editing the metadata of books in bulk. The calibre book editor can also use regular expressions in its *Search and replace* feature. Finally, you can use regular expressions when searching the calibre book list and when searching inside the calibre E-book viewer.

10.4.3 Czym właściwie *jest* wyrażenie regularne?

Wyrażenie regularne jest sposobem na opisanie ciągu znaków. Pojedyncze wyrażenie może *pasować* do wielu różnych ciągów. To właśnie czyni wyrażenia regularne tak potężnym – to zwięzły sposób na opisanie potencjalnie dużej liczby różnych wersji.

Informacja: I'm using string here in the sense it is used in programming languages: a string of one or more characters, characters including actual characters, numbers, punctuation and so-called whitespace (linebreaks, tabulators etc.). Please note that generally, uppercase and lowercase characters are not considered the same, thus „a” being a different character from „A” and so forth. In calibre, regular expressions are case insensitive in the Search bar, but not in the conversion options. There's a way to make every regular expression case insensitive, but we'll discuss that later. It gets complicated because regular expressions allow for variations in the strings it matches, so one expression can match multiple strings, which is why people bother using them at all. More on that in a bit.

10.4.4 Możesz to wyjaśnić?

Otóż dlatego właśnie tu jesteśmy. Po pierwsze najważniejsza koncepcja wyrażeń regularnych: *Ciąg znaków jest wyrażeniem regularnym, pasującym do samego siebie*. To znaczy, jeśli chcę dopasować ciąg "Hello, World!" przy użyciu wyrażeń regularnych, to najprostsze z nich to Hello, World!. Owszem, to jest takie proste. Zauważysz zapewne, że wyrażenie to pasuje *wyłącznie* do "Hello, World!", a nie na przykład "Hello, wOrld!" czy "hello, world!" ani żadnej innej odmiany.

10.4.5 Nie brzmi źle. Co dalej?

Next is the beginning of the really good stuff. Remember where I said that regular expressions can match multiple strings? This is where it gets a little more complicated. Say, as a somewhat more practical exercise, the e-book you wanted to convert had a nasty footer counting the pages, like „Page 5 of 423”. Obviously the page number would rise from 1 to 423, thus you'd have to match 423 different strings, right? Wrong, actually: regular expressions allow you to define sets of characters that are matched: To define a set, you put all the characters you want to be in the set into square brackets. So, for example, the set [abc] would match either the character „a”, „b” or „c”. *Sets will always only match one of the characters in the set*. They „understand” character ranges, that is, if you wanted to match all the lower case characters, you'd use the set [a-z] for lower- and uppercase characters you'd use [a-zA-Z] and so on. Got the idea? So, obviously, using the expression Page [0-9] of 423 you'd be able to match the first 9 pages, thus reducing the expressions needed to three: The second expression Page [0-9][0-9] of 423 would match all two-digit page numbers, and I'm sure you can guess what the third expression would look like. Yes, go ahead. Write it down.

10.4.6 Hej, nieźle! To zaczyna mieć sens!

I was hoping you'd say that. But brace yourself, now it gets even better! We just saw that using sets, we could match one of several characters at once. But you can even repeat a character or set, reducing the number of expressions needed to handle the above page number example to one. Yes, ONE! Excited? You should be! It works like this: Some so-called special characters, „+”, „?” and „*”, *repeat the single element preceding them*. (Element means either a single character, a character set, an escape sequence or a group (we'll learn about those last two later)- in short, any single entity in a regular expression). These characters are called wildcards or quantifiers. To be more precise, „?” matches *0 or 1* of the preceding element, „*” matches *0 or more* of the preceding element and „+” matches *1 or more* of the preceding element. A few examples: The expression a? would match either „” (which is the empty string, not strictly useful in this case) or „a”, the expression a* would match „”, „a”, „aa” or any number of a's in a row, and, finally, the expression a+ would match „a”, „aa” or any number of a's in a row (Note: it wouldn't match the empty string!). Same deal for sets: The expression [0-9]+ would match *every integer number there is!* I know what you're thinking, and you're right: If you use that in the

above case of matching page numbers, wouldn't that be the single one expression to match all the page numbers? Yes, the expression `Page [0-9]+ of 423` would match every page number in that book!

Informacja: A note on these quantifiers: They generally try to match as much text as possible, so be careful when using them. This is called „greedy behaviour”- I'm sure you get why. It gets problematic when you, say, try to match a tag. Consider, for example, the string `<p class="calibre2">Title here</p>` and let's say you'd want to match the opening tag (the part between the first pair of angle brackets, a little more on tags later). You'd think that the expression `<p.*>` would match that tag, but actually, it matches the whole string! (The character „.” is another special character. It matches anything *except* linebreaks, so, basically, the expression `.*` would match any single line you can think of). Instead, try using `<p.*?>` which makes the quantifier `.*` non-greedy. That expression would only match the first opening tag, as intended. There's actually another way to accomplish this: The expression `<p[^>]*>` will match that same opening tag- you'll see why after the next section. Just note that there quite frequently is more than one way to write a regular expression.

10.4.7 No tak, te znaki specjalne są bardzo fajne i w ogóle, ale co jeśli chcemy dopasować kropkę albo znak zapytania?

Możesz oczywiście zrobić tak: poprzedź znak specjalny backslashem i będzie on interpretowany jak każdy inny znak, bez specjalnego znaczenia. Taka para backslasha i znaku specjalnego jest nazywana sekwencją unikową, a umieszczanie backslasha przed znakiem to maskowanie znaków. Sekwencja unikowa jest interpretowana jak jeden element. Rzecz jasna są sekwencje unikowe, które mają znaczenie inne niż tylko maskowanie znaków specjalnych. Na przykład `"\t"` oznacza tabulator. Więcej o sekwencjach unikowych powiemy później. A przy okazji, co do znaków specjalnych: potraktuj te znaki, o których mowa jako znaki mające specjalne funkcje i tym samym konieczne jest maskowanie jeśli chcesz uzyskać literalnie konkretny znak.

10.4.8 A jakie są najbardziej przydatne zestawy?

Knew you'd ask. Some useful sets are `[0-9]` matching a single number, `[a-z]` matching a single lowercase letter, `[A-Z]` matching a single uppercase letter, `[a-zA-Z]` matching a single letter and `[a-zA-Z0-9]` matching a single letter or number. You can also use an escape sequence as shorthand:

`\d`
is equivalent to `[0-9]`

`\w`
is equivalent to `[a-zA-Z0-9_]`

`\s`
is equivalent to any whitespace

Informacja: „Whitespace” is a term for anything that won't be printed. These characters include space, tabulator, line feed, form feed, carriage return, non-breaking spaces, etc.

Informacja: The upper and lower case sets may match both upper and lowercase if the setting to make searches case insensitive is enabled. Such settings are found, for instance in Preferences->Searching in calibre itself and on the Search panel in the calibre *E-book viewer* as well as the calibre *Edit book* tool.

Na koniec informacji o zestawach - możesz również zdefiniować wszystkie znaki *oprócz* tych ujętych w zestawie. Robi się to poprzez dodanie znaku `^` jako *pierwszego znaku w zestawie*. Tak więc `[^a]` dopasuje każdy znak oprócz „a”.

Nazywane jest to uzupełnianiem zestawu. Sekwencje unikowe, o których pisaliśmy wcześniej również mogą być w ten sposób uzupełniane: "`\D`" oznacza dowolny znak nienumeryczny, jest więc ekwiwalentem `[^0-9]`. Inne skróty, jak się zapewne spodziewasz, mogą również być uzupełniane poprzez wielką literę zamiast małej. Powracając do przykładu `<p [^>]*>` z poprzedniego rozdziału rozumiesz już, że w tym przypadku zestaw jest używany do dopasowania dowolnego znaku oprócz zamykającego nawiasu ostrego.

10.4.9 A gdybym miał kilka różnych ciągów, które chciałbym dopasować, to czy byłoby to skomplikowane?

Fear not, life still is good and easy. Consider this example: The book you're converting has „Title” written on every odd page and „Author” written on every even page. Looks great in print, right? But in e-books, it's annoying. You can group whole expressions in normal parentheses, and the character "`|`" will let you match *either* the expression to its right *or* the one to its left. Combine those and you're done. Too fast for you? Okay, first off, we group the expressions for odd and even pages, thus getting `(Title)(Author)` as our two needed expressions. Now we make things simpler by using the vertical bar ("`|`" is called the vertical bar character): If you use the expression `(Title|Author)` you'll either get a match for „Title” (on the odd pages) or you'd match „Author” (on the even pages). Well, wasn't that easy?

Możesz oczywiście użyć pionowej kreski bez grupujących nawiasów. Pamiętaj, że kwantyfikatory powtarzają poprzedzający je znak? Pionowa kreska działa nieco inaczej: wyrażenie „Tytuł|Autor” także dopasuje ciąg „Tytuł” albo ciąg „Autor”, tak jak w poprzednim przykładzie, gdzie użyliśmy nawiasów. *Pionowa kreska obejmuje swoim działaniem całe wyrażenie przed i za nią.* Tak więc jeśli chcesz odnaleźć ciąg „Calibre” i „calibre”, wybierając pomiędzy małym a wielkim „C”, musisz użyć wyrażenia `(c|C)alibre`, w którym grupowanie oznacza, że wybór dotyczy wyłącznie „c”. Gdybyś użył `c|Calibre` to dopasowany zostałby ciąg „c” albo ciąg „Calibre”, co jest nieprawidłowe. W skrócie: jeśli masz wątpliwości, grupuj wyrażenia, jeśli używasz pionowej kreski.

10.4.10 Zapomniałem...

...moment, jest jeszcze jedna, naprawdę użyteczna funkcja grup. Jeśli w wyrażeniu jest grupa, która została dopasowana, możesz się do niej odnieść później w tym wyrażeniu: grupy są numerowane, począwszy od 1, możesz użyć tych numerów, poprzedzając je backslashem, tak więc piąta grupa będzie oznaczona przez `\5`. Jeśli więc szukasz `([^]+) \1` otrzymasz cały ciąg!

10.4.11 Na początku powiedziałeś, że w wyrażeniach regularnych wielkość liter może nie mieć znaczenia?

Yes, I did, thanks for paying attention and reminding me. You can tell calibre how you want certain things handled by using something called flags. You include flags in your expression by using the special construct `(?flags go here)` where, obviously, you'd replace „flags go here” with the specific flags you want. For ignoring case, the flag is `i`, thus you include `(?i)` in your expression. Thus, `(?i)test` would match „Test”, „tEst”, „TEst” and any case variation you could think of.

Another useful flag lets the dot match any character at all, *including* the newline, the flag `s`. If you want to use multiple flags in an expression, just put them in the same statement: `(?is)` would ignore case and make the dot match all. It doesn't matter which flag you state first, `(?si)` would be equivalent to the above.

10.4.12 Zaczynam chyba rozumieć, o co chodzi w wyrażeniach regularnych, a teraz... jak ich użyć w calibre?

Konwertowanie

Zacznijmy od ustawień konwertowania, co jest dość przyjemne. W części :guilabel:„Znajdź i zamień” możesz wprowadzić wyrażenie regularne opisujące napis, który zostanie podmieniony w trakcie konwertowania książki. Tu z pomocą przychodzi kreator. Naciśnij przycisk kreatora, a zobaczysz podgląd tego, jak calibre „widzi” książkę podczas konwertowania. Przewiń tekst do napisu, który chcesz usunąć, zaznacz go, skopiuj i wklej do pola wyrażenia regularnego na górze okna. Jeśli napis zawiera części zmienne, takie jak numery stron itp., możesz użyć zbiorów i kwantyfikatorów, by je wychwycić. Skoro już przy tym jesteśmy, pamiętaj, by poprzedzić jakiegokolwiek znaki specjalne znakiem modyfikacji (tj. odwrotnym ukośnikiem). Naciśnij przycisk :guilabel:„Test”, a calibre wyróżni części, które zastąpiłby podczas konwertowania. Kiedy już będziesz gotowy, naciśnij OK, by rozpocząć konwertowanie. Pamiętaj, że jeżeli źródło konwertowania zawiera znaczniki, jak w poniższym przykładzie:

```
Maybe, but the cops feel like you do, Anita. What's one more dead vampire?
New laws don't change that. </p>
<p class="calibre4"> <b class="calibre2">Generated by ABC Amber LIT Conv
<a href="http://www.processtext.com/abclit.html" class="calibre3">erter,
http://www.processtext.com/abclit.html</a></b></p>
<p class="calibre4"> It had only been two years since Addison v. Clark.
The court case gave us a revised version of what life was
```

(shamelessly ripped out of [this thread](#)⁹³). You'd have to remove some of the tags as well. In this example, I'd recommend beginning with the tag `<b class="calibre2">`, now you have to end with the corresponding closing tag (opening tags are `<tag>`, closing tags are `</tag>`), which is simply the next `` in this case. (Refer to a good HTML manual or ask in the forum if you are unclear on this point). The opening tag can be described using `<b.*?>`, the closing tag using ``, thus we could remove everything between those tags using `<b.*?>.*?`. But using this expression would be a bad idea, because it removes everything enclosed by ``-tags (which, by the way, render the enclosed text in bold print), and it's a fair bet that we'll remove portions of the book in this way. Instead, include the beginning of the enclosed string as well, making the regular expression `<b.*?\s*Generated\s+by\s+ABC\s+Amber\s+LIT.*?`. The `\s` with quantifiers are included here instead of explicitly using the spaces as seen in the string to catch any variations of the string that might occur. Remember to check what calibre will remove to make sure you don't remove any portions you want to keep if you test a new expression. If you only check one occurrence, you might miss a mismatch somewhere else in the text. Also note that should you accidentally remove more or fewer tags than you actually wanted to, calibre tries to repair the damaged code after doing the removal.

Dodawanie książek

Another thing you can use regular expressions for is to extract metadata from filenames. You can find this feature in the „Adding books” part of the settings. There's a special feature here: You can use field names for metadata fields, for example `(?P<title>)` would indicate that calibre uses this part of the string as book title. The allowed field names are listed in the windows, together with another nice test field. An example: Say you want to import a whole bunch of files named like Classical Texts: The Divine Comedy by Dante Alighieri.mobi. (Obviously, this is already in your library, since we all love classical italian poetry) or Science Fiction epics: The Foundation Trilogy by Isaac Asimov.epub. This is obviously a naming scheme that calibre won't extract any meaningful data out of - its standard expression for extracting metadata is `(?P<title>.+)` - `(?P<author>[^_]+)`. A regular expression that works here would be `[a-zA-Z]+: (?P<title>.+)` by `(?P<author>.+)`. Please note that, inside the group for the metadata field, you need to use expressions to describe what the field actually matches. And also note that, when using the test field calibre provides, you need to add the file extension to your testing filename, otherwise you won't get any matches at all, despite using a working expression.

⁹³ <http://www.mobileread.com/forums/showthread.php?t=75594>

Masowa edycja metadanych

The last part is regular expression *Search and replace* in metadata fields. You can access this by selecting multiple books in the library and using bulk metadata edit. Be very careful when using this last feature, as it can do **Very Bad Things** to your library! Doublecheck that your expressions do what you want them to using the test fields, and only mark the books you really want to change! In the regular expression search mode, you can search in one field, replace the text with something and even write the result into another field. A practical example: Say your library contained the books of Frank Herbert's *Dune* series, named after the fashion *Dune 1 - Dune*, *Dune 2 - Dune* *Messiah* and so on. Now you want to get *Dune* into the series field. You can do that by searching for `(.*?) \d+ - .*` in the title field and replacing it with `\1` in the series field. See what I did there? That's a reference to the first group you're replacing the series field with. Now that you have the series all set, you only need to do another search for `.*?` in the title field and replace it with `" "` (an empty string), again in the title field, and your metadata is all neat and tidy. Isn't that great? By the way, instead of replacing the entire field, you can also append or prepend to the field, so, if you *wanted* the book title to be prepended with series info, you could do that as well. As you by now have undoubtedly noticed, there's a checkbox labeled *Case sensitive*, so you won't have to use flags to select behaviour here.

Well, that just about concludes the very short introduction to regular expressions. Hopefully I'll have shown you enough to at least get you started and to enable you to continue learning by yourself- a good starting point would be the [Python documentation for regexps](https://docs.python.org/library/re.html)⁹⁴.

One last word of warning, though: Regexps are powerful, but also really easy to get wrong. calibre provides really great testing possibilities to see if your expressions behave as you expect them to. Use them. Try not to shoot yourself in the foot. (God, I love that expression...). But should you, despite the warning, injure your foot (or any other body parts), try to learn from it.

10.4.13 Szybkie odniesienie

Krótką dokumentacja składni wyrażeń regularnych

This checklist summarizes the most commonly used/hard to remember parts of the regexp engine available in most parts of calibre.

Spis treści

- *Klasy znaków* (strona 220)
- *Shorthand character classes* (strona 220)
- *Kwantyfikatory* (strona 220)
- *Chciwość* (strona 221)
- *Alternation* (strona 221)
- *Exclusion* (strona 221)
- *anchors* (strona 221)
- *Groups* (strona 222)
- *Lookarounds* (strona 222)
- *Recursion* (strona 223)
- *Special characters* (strona 223)

⁹⁴ <https://docs.python.org/library/re.html>

- *Meta-characters* (strona 223)
- *Modes* (strona 224)

Klasy znaków

Character classes are useful to represent different groups of characters, succinctly.

Przykłady:

Reprezentacja	Klasa
[a-z]	Małe litery. Nie zawiera znaków diakrytycznych i ligatur
[a-z0-9]	Małe litery od a do z albo cyfry od 0 do 9
[A-Za-z-]	Wielkie lub małe litery, albo myślnik. Żeby umieścić myślnik w klasie, umieść go na początku lub końcu tak, by nie mylił się ze znakiem łącznika, który określa zakres znaków.
[^0-9]	Any character except a digit. The caret (^) placed at the beginning of the class excludes the characters of the class (complemented class)
[[a-z] -- [ae	Małe spółgłoski. Klasa może być zawarta w klasie. Znaki -- wykluczają to, co następuje po nich.
[\w -- [\	Wszystkie litery (włączając znaki diakrytyczne). Skrócone klasy mogą być użyte wewnątrz klas.
d_]]	

Przykład:

```
<[^<>]+> to select an HTML tag
```

Shorthand character classes

Reprezentacja	Klasa
\d	Cyfra (to samo co [0-9])
\D	Any non-numeric character (same as [^0-9])
\w	Znak alfanumeryczny [a-zA-Z0-9_] włączając znaki diakrytyczne i ligatury
\W	Każdy znak nie będący "słowem"
\s	Spacja, spacja niełamująca, tabulator, znak końca linii
\S	Any "non-whitespace" character
.	Any character except newline. Use the "dot all" checkbox or the (?s) regex modifier to include the newline character.

Kwantyfikatory

Kwantyfikator	Liczba wystąpień wyrażenia poprzedzającego kwantyfikator
?	0 albo 1 wystąpienie wyrażenia. To samo co {0, 1}
+	1 albo więcej wystąpień wyrażenia. To samo co {1, }
*	0, 1 albo więcej wystąpień wyrażenia. to samo co {0, }
{n}	Dokładnie n wystąpień wyrażenia.
{min,max}	Liczba wystąpień pomiędzy podanymi wartościami minimalną i maksymalną.
{min, }	Liczba wystąpień pomiędzy podaną wartością minimalną a nieskończonością
{ , max}	Liczba wystąpień pomiędzy 0 a podaną wartością maksymalną

Chciwość

By default, with quantifiers, the regular expression engine is greedy: it extends the selection as much as possible. This often causes surprises, at first. `?` follows a quantifier to make it lazy. Avoid putting two in the same expression, the result can be unpredictable.

Beware of nesting quantifiers, for example, the pattern `(a*)*`, as it exponentially increases processing time.

Alternation

The `|` character in a regular expression is a logical OR. It means that either the preceding or the following expression can match.

Exclusion

Method 1

```
pattern_to_exclude(*SKIP)(*FAIL)|pattern_to_select
```

Example:

```
"Blabla"(*SKIP)(*FAIL)|Blabla
```

selects Blabla, in the strings Blabla or „Blabla or Blabla”, but not in „Blabla”.

Method 2

```
pattern_to_exclude\K|(pattern_to_select)
```

```
"Blabla"\K|(Blabla)
```

selects Blabla, in the strings Blabla or „Blabla or Blabla”, but not in „Blabla”.

Anchors

An anchor is a way to match a logical location in a string, rather than a character. The most useful anchors for text processing are:

\b

Designates a word boundary, i.e. a transition from space to non-space character. For example, you can use `\bsurd` to match the `surd` but not `absurd`.

^

Matches the start of a line (in multi-line mode, which is the default)

\$

Matches the end of a line (in multi-line mode, which is the default)

\K

Resets the start position of the selection to its position in the pattern. Some regexp engines (but not calibre) do not allow lookbehind of variable length, especially with quantifiers. When you can use `\K` with these engines, it also allows you to get rid of this limit by writing the equivalent of a positive lookbehind of variable length.

Groups

(**expression**)

Capturing group, which stores the selection and can be recalled later in the *search* or *replace* patterns with `\n`, where `n` is the sequence number of the capturing group (starting at 1 in reading order)

(**? : expression**)

Group that does not capture the selection

(**? > expression**)

Atomic Group: As soon as the expression is satisfied, the regexp engine passes, and if the rest of the pattern fails, it will not backtrack to try other combinations with the expression. Atomic groups do not capture.

(**? | expression**)

Branch reset group: the branches of the alternations included in the expression share the same group numbers

(**? < name > expression**)

Group named “name”. The selection can be recalled later in the *search* pattern by `(?P=name)` and in the *replace* by `\g<name>`. Two different groups can use the same name.

Lookarounds

Lookaround	Meaning
<code>?=</code>	Positive lookahead (to be placed after the selection)
<code>?!</code>	Negative lookahead (to be placed after the selection)
<code>?<=</code>	Positive lookbehind (to be placed before the selection)
<code>?<!</code>	Negative lookbehind (to be placed before the selection)

Lookaheads and lookbehinds do not consume characters, they are zero length and do not capture. They are atomic groups: as soon as the assertion is satisfied, the regexp engine passes, and if the rest of the pattern fails, it will not backtrack inside the lookahead to try other combinations.

When looking for multiple matches in a string, at the starting position of each match attempt, a lookbehind can inspect the characters before the current position. Therefore, on the string 123, the pattern `(?<=\d)\d` (a digit preceded by a digit) should, in theory, select 2 and 3. On the other hand, `\d\K\d` can only select 2, because the starting position after the first selection is immediately before 3, and there are not enough digits for a second match. Similarly, `\d(\d)` only captures 2. In calibre’s regexp engine practice, the positive lookbehind behaves in the same way, and selects only 2, contrary to theory.

Groups can be placed inside lookarounds, but capture is rarely useful. Nevertheless, if it is useful, it will be necessary to be very careful in the use of a quantifier in a lookbehind: the greed associated with the absence of backtracking can give a surprising capture. For this reason, use `\K` rather than a positive lookbehind when you have a quantifier (or worse, several) in a capturing group of the positive lookbehind.

Example of negative lookahead:

```
(?! [^<>{ } ] * [ > ] )
```

Placed at the end of the pattern prevents to select within a tag or a style embedded in the file.

Whenever possible, it is always better to „anchor” the lookarounds, to reduce the number of steps necessary to obtain the result.

Recursion

Reprezentacja	Meaning
(?R)	Recursion of the entire pattern
(?1)	Recursion of the only pattern of the numbered capturing group, here group 1

Recursion is calling oneself. This is useful for balanced queries, such as quoted strings, which can contain embedded quoted strings. Thus, if during the processing of a string between double quotation marks, we encounter the beginning of a new string between double quotation marks, well we know how to do, and we call ourselves. Then we have a pattern like:

```
start-pattern(?>atomic sub-pattern|(?R))*end-pattern
```

To select a string between double quotation marks without stopping on an embedded string:

```
"( (?>[^\"]+|(?R))* [^\"]+ )"
```

This template can also be used to modify pairs of tags that can be embedded, such as `<div>` tags.

Special characters

Reprezentacja	Character
\t	tabulation
\n	line break
\x20	(breakable) space
\xa0	no-break space

Meta-characters

Meta-characters are those that have a special meaning for the regexp engine. Of these, twelve must be preceded by an escape character, the backslash (\), to lose their special meaning and become a regular character again:

```
^ . [ ] $ ( ) * + ? | \
```

Seven other meta-characters do not need to be preceded by a backslash (but can be without any other consequence):

```
{ } ! < > = :
```

Special characters lose their status if they are used inside a class (between brackets `[]`). The closing bracket and the dash have a special status in a class. Outside the class, the dash is a simple literal, the closing bracket remains a meta-character.

The slash (/) and the number sign (or hash character) (#) are not meta-characters, they don't need to be escaped.

In some tools, like regex101.com with the Python engine, double quotes have the special status of separator, and must be escaped, or the options changed. This is not the case in the editor of calibre.

Modes

(?s)

Causes the dot (.) to match newline characters as well

(?m)

Makes the ^ and \$ anchors match the start and end of lines instead of the start and end of the entire string.

10.4.14 Uznanie

Dzięki za pomocne rady, korekty itp.:

- Idolse
- kovidgoyal
- chaley
- dwanthny
- kacir
- Starson17
- Orpheu

For more about regexps see [The Python User Manual](#)⁹⁵. The actual regular expression library used by calibre is: `regex`⁹⁶ which supports several useful enhancements over the Python standard library one.

10.5 Pisanie własnych wtyczek, by rozszerzyć funkcjonalność calibre

calibre has a very modular design. Almost all functionality in calibre comes in the form of plugins. Plugins are used for conversion, for downloading news (though these are called recipes), for various components of the user interface, to connect to different devices, to process files when adding them to calibre and so on. You can get a complete list of all the built-in plugins in calibre by going to *Preferences* → *Advanced* → *Plugins*.

Here, we will teach you how to create your own plugins to add new features to calibre.

Spis treści

- *Anatomia wtyczki calibre* (strona 225)
- *A User Interface plugin* (strona 226)
 - `__init__.py` (strona 227)
 - `ui.py` (strona 228)
 - `main.py` (strona 229)
 - *Getting resources from the plugin ZIP file* (strona 232)
 - *Enabling user configuration of your plugin* (strona 232)

⁹⁵ <https://docs.python.org/library/re.html>

⁹⁶ <https://bitbucket.org/mrabarnett/mrab-regex/src/hg/>

- *Edit book plugins* (strona 234)
 - *main.py* (strona 235)
- *Adding translations to your plugin* (strona 237)
- *The plugin API* (strona 238)
- *Debugging plugins* (strona 238)
- *More plugin examples* (strona 239)
- *Sharing your plugins with others* (strona 239)

Informacja: Dotyczy to wyłącznie calibre w wersji >= 0.8.60

10.5.1 Anatomia wtyczki calibre

A calibre plugin is very simple, it's just a ZIP file that contains some Python code and any other resources like image files needed by the plugin. Without further ado, let's see a basic example.

Suppose you have an installation of calibre that you are using to self publish various e-documents in EPUB and MOBI formats. You would like all files generated by calibre to have their publisher set as „Hello world”, here's how to do it. Create a file named `__init__.py` (this is a special name and must always be used for the main file of your plugin) and enter the following Python code into it:

```
from calibre.customize import FileTypePlugin

class HelloWorld(FileTypePlugin):

    name = 'Hello World Plugin' # Name of the plugin
    description = 'Set the publisher to Hello World for all new conversions'
    supported_platforms = ['windows', 'osx', 'linux'] # Platforms this plugin will
    ↪run on
    author = 'Acme Inc.' # The author of this plugin
    version = (1, 0, 0) # The version number of this plugin
    file_types = {'epub', 'mobi'} # The file types that this plugin will be
    ↪applied to
    on_postprocess = True # Run this plugin after conversion is complete
    minimum_calibre_version = (0, 7, 53)

    def run(self, path_to_ebook):
        from calibre.ebooks.metadata.meta import get_metadata, set_metadata
        with open(path_to_ebook, 'r+b') as file:
            ext = os.path.splitext(path_to_ebook)[-1][1:].lower()
            mi = get_metadata(file, ext)
            mi.publisher = 'Hello World'
            set_metadata(file, mi, ext)
        return path_to_ebook
```

That's all. To add this code to calibre as a plugin, simply run the following in the folder in which you created `__init__.py`:

```
calibre-customize -b .
```

Informacja: W systemie macOS narzędzia wiersza poleceń znajdują się w pakiecie kalibru, na przykład, jeśli zainstalowałeś calibre w /Applications, narzędzia wiersza poleceń znajdują się w `:file:/Applications/calibre.app/Contents/MacOS/``.

You can download the Hello World plugin from [helloworld_plugin.zip](#)⁹⁷.

Every time you use calibre to convert a book, the plugin's `run()` method will be called and the converted book will have its publisher set to „Hello World”. This is a trivial plugin, lets move on to a more complex example that actually adds a component to the user interface.

10.5.2 A User Interface plugin

This plugin will be spread over a few files (to keep the code clean). It will show you how to get resources (images or data files) from the plugin ZIP file, allow users to configure your plugin, how to create elements in the calibre user interface and how to access and query the books database in calibre.

You can download this plugin from [interface_demo_plugin.zip](#)⁹⁸

The first thing to note is that this ZIP file has a lot more files in it, explained below, pay particular attention to `plugin-import-name-interface_demo.txt`.

plugin-import-name-interface_demo.txt

An empty text file used to enable the multi-file plugin magic. This file must be present in all plugins that use more than one `.py` file. It should be empty and its filename must be of the form: `plugin-import-name-some_name.txt`. The presence of this file allows you to import code from the `.py` files present inside the ZIP file, using a statement like:

```
from calibre_plugins.some_name.some_module import some_object
```

The prefix `calibre_plugins` must always be present. `some_name` comes from the filename of the empty text file. `some_module` refers to `some_module.py` file inside the ZIP file. Note that this importing is just as powerful as regular Python imports. You can create packages and subpackages of `.py` modules inside the ZIP file, just like you would normally (by defining `__init__.py` in each sub-folder), and everything should „just work”.

The name you use for `some_name` enters a global namespace shared by all plugins, **so make it as unique as possible**. But remember that it must be a valid Python identifier (only alphabets, numbers and the underscore).

__init__.py

As before, the file that defines the plugin class

main.py

This file contains the actual code that does something useful

ui.py

This file defines the interface part of the plugin

images/icon.png

The icon for this plugin

⁹⁷ https://calibre-ebook.com/downloads/helloworld_plugin.zip

⁹⁸ https://calibre-ebook.com/downloads/interface_demo_plugin.zip

about.txt

A text file with information about the plugin

translations

A folder containing .mo files with the translations of the user interface of your plugin into different languages. See below for details.

Now let's look at the code.

__init__.py

First, the obligatory __init__.py to define the plugin metadata:

```
from calibre.customize import InterfaceActionBase

class InterfacePluginDemo(InterfaceActionBase):
    '''
    This class is a simple wrapper that provides information about the actual
    plugin class. The actual interface plugin class is called InterfacePlugin
    and is defined in the ui.py file, as specified in the actual_plugin field
    below.

    The reason for having two classes is that it allows the command line
    calibre utilities to run without needing to load the GUI libraries.
    '''
    name = 'Interface Plugin Demo'
    description = 'An advanced plugin demo'
    supported_platforms = ['windows', 'osx', 'linux']
    author = 'Kovid Goyal'
    version = (1, 0, 0)
    minimum_calibre_version = (0, 7, 53)

    #: This field defines the GUI plugin class that contains all the code
    #: that actually does something. Its format is module_path:class_name
    #: The specified class must be defined in the specified module.
    actual_plugin = 'calibre_plugins.interface_demo.ui:InterfacePlugin'

    def is_customizable(self):
        '''
        This method must return True to enable customization via
        Preferences->Plugins
        '''
        return True

    def config_widget(self):
        '''
        Implement this method and :meth:`save_settings` in your plugin to
        use a custom configuration dialog.

        This method, if implemented, must return a QWidget. The widget can have
        an optional method validate() that takes no arguments and is called
        immediately after the user clicks OK. Changes are applied if and only
        if the method returns True.

        If for some reason you cannot perform the configuration at this time,
        return a tuple of two strings (message, details), these will be
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

displayed as a warning dialog to the user and the process will be
aborted.

The base class implementation of this method raises NotImplementedError
so by default no user configuration is possible.
'''
# It is important to put this import statement here rather than at the
# top of the module as importing the config class will also cause the
# GUI libraries to be loaded, which we do not want when using calibre
# from the command line
from calibre_plugins.interface_demo.config import ConfigWidget
return ConfigWidget()

def save_settings(self, config_widget):
    '''
    Save the settings specified by the user with config_widget.

    :param config_widget: The widget returned by :meth:`config_widget`.
    '''
    config_widget.save_settings()

    # Apply the changes
    ac = self.actual_plugin_
    if ac is not None:
        ac.apply_settings()

```

The only noteworthy feature is the field `actual_plugin`. Since calibre has both command line and GUI interfaces, GUI plugins like this one should not load any GUI libraries in `__init__.py`. The `actual_plugin` field does this for you, by telling calibre that the actual plugin is to be found in another file inside your ZIP archive, which will only be loaded in a GUI context.

Remember that for this to work, you must have a `plugin-import-name-some_name.txt` file in your plugin ZIP file, as discussed above.

Also there are a couple of methods for enabling user configuration of the plugin. These are discussed below.

ui.py

Now let's look at `ui.py` which defines the actual GUI plugin. The source code is heavily commented and should be self explanatory:

```

from calibre.gui2.actions import InterfaceAction
from calibre_plugins.interface_demo.main import DemoDialog

class InterfacePlugin(InterfaceAction):

    name = 'Interface Plugin Demo'

    # Declare the main action associated with this plugin
    # The keyboard shortcut can be None if you dont want to use a keyboard
    # shortcut. Remember that currently calibre has no central management for
    # keyboard shortcuts, so try to use an unusual/unused shortcut.
    action_spec = ('Interface Plugin Demo', None,

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

        'Run the Interface Plugin Demo', 'Ctrl+Shift+F1')

def genesis(self):
    # This method is called once per plugin, do initial setup here

    # Set the icon for this interface action
    # The get_icons function is a builtin function defined for all your
    # plugin code. It loads icons from the plugin zip file. It returns
    # QIcon objects, if you want the actual data, use the analogous
    # get_resources builtin function.
    #
    # Note that if you are loading more than one icon, for performance, you
    # should pass a list of names to get_icons. In this case, get_icons
    # will return a dictionary mapping names to QIcons. Names that
    # are not found in the zip file will result in null QIcons.
    icon = get_icons('images/icon.png', 'Interface Demo Plugin')

    # The qaction is automatically created from the action_spec defined
    # above
    self.qaction.setIcon(icon)
    self.qaction.triggered.connect(self.show_dialog)

def show_dialog(self):
    # The base plugin object defined in __init__.py
    base_plugin_object = self.interface_action_base_plugin
    # Show the config dialog
    # The config dialog can also be shown from within
    # Preferences->Plugins, which is why the do_user_config
    # method is defined on the base plugin class
    do_user_config = base_plugin_object.do_user_config

    # self.gui is the main calibre GUI. It acts as the gateway to access
    # all the elements of the calibre user interface, it should also be the
    # parent of the dialog
    d = DemoDialog(self.gui, self.qaction.icon(), do_user_config)
    d.show()

def apply_settings(self):
    from calibre_plugins.interface_demo.config import prefs
    # In an actual non trivial plugin, you would probably need to
    # do something based on the settings in prefs
    prefs

```

main.py

The actual logic to implement the Interface Plugin Demo dialog.

```

from qt.core import QDialog, QLabel, QMessageBox, QPushButton, QVBoxLayout

class DemoDialog(QDialog):

    def __init__(self, gui, icon, do_user_config):
        QDialog.__init__(self, gui)
        self.gui = gui

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

self.do_user_config = do_user_config

# The current database shown in the GUI
# db is an instance of the class LibraryDatabase from db/legacy.py
# This class has many, many methods that allow you to do a lot of
# things. For most purposes you should use db.new_api, which has
# a much nicer interface from db/cache.py
self.db = gui.current_db

self.l = QVBoxLayout()
self.setLayout(self.l)

self.label = QLabel(prefs['hello_world_msg'])
self.l.addWidget(self.label)

self.setWindowTitle('Interface Plugin Demo')
self.setWindowIcon(icon)

self.about_button = QPushButton('About', self)
self.about_button.clicked.connect(self.about)
self.l.addWidget(self.about_button)

self.marked_button = QPushButton(
    'Show books with only one format in the calibre GUI', self)
self.marked_button.clicked.connect(self.marked)
self.l.addWidget(self.marked_button)

self.view_button = QPushButton(
    'View the most recently added book', self)
self.view_button.clicked.connect(self.view)
self.l.addWidget(self.view_button)

self.update_metadata_button = QPushButton(
    'Update metadata in a book\'s files', self)
self.update_metadata_button.clicked.connect(self.update_metadata)
self.l.addWidget(self.update_metadata_button)

self.conf_button = QPushButton(
    'Configure this plugin', self)
self.conf_button.clicked.connect(self.config)
self.l.addWidget(self.conf_button)

self.resize(self.sizeHint())

def about(self):
    # Get the about text from a file inside the plugin zip file
    # The get_resources function is a builtin function defined for all your
    # plugin code. It loads files from the plugin zip file. It returns
    # the bytes from the specified file.
    #
    # Note that if you are loading more than one file, for performance, you
    # should pass a list of names to get_resources. In this case,
    # get_resources will return a dictionary mapping names to bytes. Names that
    # are not found in the zip file will not be in the returned dictionary.
    text = get_resources('about.txt')
    QMessageBox.about(self, 'About the Interface Plugin Demo',
        text.decode('utf-8'))

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

def marked(self):
    ''' Show books with only one format '''
    db = self.db.new_api
    matched_ids = {book_id for book_id in db.all_book_ids() if len(db.
↪formats(book_id)) == 1}
    # Mark the records with the matching ids
    # new_api does not know anything about marked books, so we use the full
    # db object
    self.db.set_marked_ids(matched_ids)

    # Tell the GUI to search for all marked records
    self.gui.search.setText('marked:true')
    self.gui.search.do_search()

def view(self):
    ''' View the most recently added book '''
    most_recent = most_recent_id = None
    db = self.db.new_api
    for book_id, timestamp in db.all_field_for('timestamp', db.all_book_ids()).
↪items():
        if most_recent is None or timestamp > most_recent:
            most_recent = timestamp
            most_recent_id = book_id

    if most_recent_id is not None:
        # Get a reference to the View plugin
        view_plugin = self.gui.iactions['View']
        # Ask the view plugin to launch the viewer for row_number
        view_plugin._view_calibre_books([most_recent_id])

def update_metadata(self):
    '''
    Set the metadata in the files in the selected book's record to
    match the current metadata in the database.
    '''
    from calibre.ebooks.metadata.meta import set_metadata
    from calibre.gui2 import error_dialog, info_dialog

    # Get currently selected books
    rows = self.gui.library_view.selectionModel().selectedRows()
    if not rows or len(rows) == 0:
        return error_dialog(self.gui, 'Cannot update metadata',
                            'No books selected', show=True)

    # Map the rows to book ids
    ids = list(map(self.gui.library_view.model().id, rows))
    db = self.db.new_api
    for book_id in ids:
        # Get the current metadata for this book from the db
        mi = db.get_metadata(book_id, get_cover=True, cover_as_data=True)
        fmts = db.formats(book_id)
        if not fmts:
            continue
        for fmt in fmts:
            fmt = fmt.lower()
            # Get a python file object for the format. This will be either
            # an in memory file or a temporary on disk file

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

        ffile = db.format(book_id, fmt, as_file=True)
        ffile.seek(0)
        # Set metadata in the format
        set_metadata(ffile, mi, fmt)
        ffile.seek(0)
        # Now replace the file in the calibre library with the updated
        # file. We dont use add_format_with_hooks as the hooks were
        # already run when the file was first added to calibre.
        db.add_format(book_id, fmt, ffile, run_hooks=False)

    info_dialog(self, 'Updated files',
                'Updated the metadata in the files of %d book(s)' % len(ids),
                show=True)

    def config(self):
        self.do_user_config(parent=self)
        # Apply the changes
        self.label.setText(prefs['hello_world_msg'])

```

Getting resources from the plugin ZIP file

calibre's plugin loading system defines a couple of built-in functions that allow you to conveniently get files from the plugin ZIP file.

get_resources(name_or_list_of_names)

This function should be called with a list of paths to files inside the ZIP file. For example to access the file `icon.png` in the folder `images` in the ZIP file, you would use: `images/icon.png`. Always use a forward slash as the path separator, even on Windows. When you pass in a single name, the function will return the raw bytes of that file or `None` if the name was not found in the ZIP file. If you pass in more than one name then it returns a dictionary mapping the names to bytes. If a name is not found, it will not be present in the returned dictionary.

get_icons(name_or_list_of_names, plugin_name=»»»)

A wrapper for `get_resources()` that creates `QIcon` objects from the raw bytes returned by `get_resources`. If a name is not found in the ZIP file the corresponding `QIcon` will be `null`. In order to support icon theme-ing, pass in the human friendly name of your plugin as `plugin_name`. If the user is using an icon theme with icons for your plugin, they will be loaded preferentially.

Enabling user configuration of your plugin

To allow users to configure your plugin, you must define three methods in your base plugin class, **is_customizable**, **config_widget** and **save_settings** as shown below:

```

def is_customizable(self):
    """
    This method must return True to enable customization via
    Preferences->Plugins
    """
    return True

```

```

def config_widget(self):
    """
    Implement this method and :meth:`save_settings` in your plugin to

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

use a custom configuration dialog.

This method, if implemented, must return a `QWidget`. The widget can have an optional method `validate()` that takes no arguments and is called immediately after the user clicks OK. Changes are applied if and only if the method returns `True`.

If for some reason you cannot perform the configuration at this time, return a tuple of two strings (message, details), these will be displayed as a warning dialog to the user and the process will be aborted.

The base class implementation of this method raises `NotImplementedError` so by default no user configuration is possible.

```
'''
# It is important to put this import statement here rather than at the
# top of the module as importing the config class will also cause the
# GUI libraries to be loaded, which we do not want when using calibre
# from the command line
from calibre_plugins.interface_demo.config import ConfigWidget
return ConfigWidget()
```

```
def save_settings(self, config_widget):
    '''
    Save the settings specified by the user with config_widget.

    :param config_widget: The widget returned by :meth:`config_widget`.
    '''
    config_widget.save_settings()

    # Apply the changes
    ac = self.actual_plugin_
    if ac is not None:
        ac.apply_settings()
```

calibre has many different ways to store configuration data (a legacy of its long history). The recommended way is to use the **JSONConfig** class, which stores your configuration information in a .json file.

The code to manage configuration data in the demo plugin is in `config.py`:

```
from qt.core import QHBoxLayout, QLabel, QLineEdit, QWidget

# This is where all preferences for this plugin will be stored
# Remember that this name (i.e. plugins/interface_demo) is also
# in a global namespace, so make it as unique as possible.
# You should always prefix your config file name with plugins/,
# so as to ensure you dont accidentally clobber a calibre config file
prefs = JSONConfig('plugins/interface_demo')

# Set defaults
prefs.defaults['hello_world_msg'] = 'Hello, World!'

class ConfigWidget(QWidget):

    def __init__(self):
        QWidget.__init__(self)
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

self.l = QHBoxLayout()
self.setLayout(self.l)

self.label = QLabel('Hello world &message:')
self.l.addWidget(self.label)

self.msg = QLineEdit(self)
self.msg.setText(prefs['hello_world_msg'])
self.l.addWidget(self.msg)
self.label.setBuddy(self.msg)

def save_settings(self):
    prefs['hello_world_msg'] = self.msg.text()

```

The prefs object is now available throughout the plugin code by a simple:

```
from calibre_plugins.interface_demo.config import prefs
```

You can see the prefs object being used in main.py:

```

self.do_user_config(parent=self)
# Apply the changes
self.label.setText(prefs['hello_world_msg'])

```

10.5.3 Edit book plugins

Now let's change gears for a bit and look at creating a plugin to add tools to the calibre book editor. The plugin is available here: [editor_demo_plugin.zip](#)⁹⁹.

The first step, as for all plugins is to create the import name empty txt file, as described *above* (strona 226). We shall name the file plugin-import-name-editor_plugin_demo.txt.

Now we create the mandatory `__init__.py` file that contains metadata about the plugin – its name, author, version, etc.

```

from calibre.customize import EditBookToolPlugin

class DemoPlugin(EditBookToolPlugin):

    name = 'Edit Book plugin demo'
    version = (1, 0, 0)
    author = 'Kovid Goyal'
    supported_platforms = ['windows', 'osx', 'linux']
    description = 'A demonstration of the plugin interface for the ebook editor'
    minimum_calibre_version = (1, 46, 0)

```

A single editor plugin can provide multiple tools each tool corresponds to a single button in the toolbar and entry in the *Plugins* menu in the editor. These can have sub-menus in case the tool has multiple related actions.

The tools must all be defined in the file `main.py` in your plugin. Every tool is a class that inherits from the `calibre.gui2.tweak_book.plugin.Tool` (strona 370) class. Let's look at `main.py` from the demo plugin, the source code is heavily commented and should be self-explanatory. Read the API documents of the `calibre.gui2.tweak_book.plugin.Tool` (strona 370) class for more details.

⁹⁹ https://calibre-ebook.com/downloads/editor_demo_plugin.zip

main.py

Here we will see the definition of a single tool that will multiply all font sizes in the book by a number provided by the user. This tool demonstrates various important concepts that you will need in developing your own plugins, so you should read the (heavily commented) source code carefully.

```
import re

from calibre import force_unicode
from calibre.ebooks.oeb.polish.container import OEB_DOCS, OEB_STYLES, serialize
from calibre.gui2 import error_dialog

# The base class that all tools must inherit from
from calibre.gui2.tweak_book.plugin import Tool
from css_parser.css import CSSRule
from qt.core import QAction, QInputDialog

class DemoTool(Tool):

    #: Set this to a unique name it will be used as a key
    name = 'demo-tool'

    #: If True the user can choose to place this tool in the plugins toolbar
    allowed_in_toolbar = True

    #: If True the user can choose to place this tool in the plugins menu
    allowed_in_menu = True

    def create_action(self, for_toolbar=True):
        # Create an action, this will be added to the plugins toolbar and
        # the plugins menu
        ac = QAction(get_icons('images/icon.png'), 'Magnify fonts', self.gui) # noqa
        if not for_toolbar:
            # Register a keyboard shortcut for this toolbar action. We only
            # register it for the action created for the menu, not the toolbar,
            # to avoid a double trigger
            self.register_shortcut(ac, 'magnify-fonts-tool', default_keys=(
↳ 'Ctrl+Shift+Alt+D',))
        ac.triggered.connect(self.ask_user)
        return ac

    def ask_user(self):
        # Ask the user for a factor by which to multiply all font sizes
        factor, ok = QInputDialog.getDouble(
            self.gui, 'Enter a magnification factor', 'Allow font sizes in the book_
↳ will be multiplied by the specified factor',
            value=2, min=0.1, max=4
        )
        if ok:
            # Ensure any in progress editing the user is doing is present in the_
↳ container
            self.boss.commit_all_editors_to_container()
            try:
                self.magnify_fonts(factor)
            except Exception:
                # Something bad happened report the error to the user
                import traceback
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

        error_dialog(self.gui, _('Failed to magnify fonts'), _('
            Failed to magnify fonts, click "Show details" for more info'),
            det_msg=traceback.format_exc(), show=True)
        # Revert to the saved restore point
        self.boss.revert_requested(self.boss.global_undo.previous_container)
    else:
        # Show the user what changes we have made, allowing her to
        # revert them if necessary
        self.boss.show_current_diff()
        # Update the editor UI to take into account all the changes we
        # have made
        self.boss.apply_container_update_to_gui()

    def magnify_fonts(self, factor):
        # Magnify all font sizes defined in the book by the specified factor
        # First we create a restore point so that the user can undo all changes
        # we make.
        self.boss.add_savepoint('Before: Magnify fonts')

        container = self.current_container # The book being edited as a container_
↪object

        # Iterate over all style declarations in the book, this means css
        # stylesheets, <style> tags and style="" attributes
        for name, media_type in container.mime_map.items():
            if media_type in OEB_STYLES:
                # A stylesheet. Parsed stylesheets are css_parser CSSStyleSheet
                # objects.
                self.magnify_stylesheet(container.parsed(name), factor)
                container.dirty(name) # Tell the container that we have changed the_
↪stylesheet
            elif media_type in OEB_DOCS:
                # A HTML file. Parsed HTML files are lxml elements

                for style_tag in container.parsed(name).xpath('//*[local-name="style"]
↪'):
                    if style_tag.text and style_tag.get('type', None) in {None, 'text/
↪css'}:
                        # We have an inline CSS <style> tag, parse it into a
                        # stylesheet object
                        sheet = container.parse_css(style_tag.text)
                        self.magnify_stylesheet(sheet, factor)
                        style_tag.text = serialize(sheet, 'text/css', pretty_
↪print=True)
                        container.dirty(name) # Tell the container that we have_
↪changed the stylesheet
                        for elem in container.parsed(name).xpath('//*[style]'):
                            # Process inline style attributes
                            block = container.parse_css(elem.get('style'), is_
↪declaration=True)
                            self.magnify_declaration(block, factor)
                            elem.set('style', force_unicode(block.getCssText(separator=' '),
↪'utf-8'))

    def magnify_stylesheet(self, sheet, factor):
        # Magnify all fonts in the specified stylesheet by the specified
        # factor.

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

for rule in sheet.cssRules.rulesOfType(CSSRule.STYLE_RULE):
    self.magnify_declaration(rule.style, factor)

def magnify_declaration(self, style, factor):
    # Magnify all fonts in the specified style declaration by the specified
    # factor
    val = style.getPropertyValue('font-size')
    if not val:
        return
    # see if the font-size contains a number
    num = re.search(r'[0-9.]', val)
    if num is not None:
        num = num.group()
        val = val.replace(num, '%f' % (float(num) * factor))
        style.setProperty('font-size', val)
    # We should also be dealing with the font shorthand property and
    # font sizes specified as non numbers, but those are left as exercises
    # for the reader

```

Let's break down `main.py`. We see that it defines a single tool, named *Magnify fonts*. This tool will ask the user for a number and multiply all font sizes in the book by that number.

The first important thing is the tool name which you must set to some relatively unique string as it will be used as the key for this tool.

The next important entry point is the `calibre.gui2.tweak_book.plugin.Tool.create_action()` (strona 371). This method creates the `QAction` objects that appear in the plugins toolbar and plugin menu. It also, optionally, assigns a keyboard shortcut that the user can customize. The triggered signal from the `QAction` is connected to the `ask_user()` method that asks the user for the font size multiplier, and then runs the magnification code.

The magnification code is well commented and fairly simple. The main things to note are that you get a reference to the editor window as `self.gui` and the editor *Boss* as `self.boss`. The *Boss* is the object that controls the editor user interface. It has many useful methods, that are documented in the `calibre.gui2.tweak_book.boss.Boss` (strona 372) class.

Finally, there is `self.current_container` which is a reference to the book being edited as a `calibre.ebooks.oeb.polish.container.Container` (strona 363) object. This represents the book as a collection of its constituent HTML/CSS/image files and has convenience methods for doing many useful things. The container object and various useful utility functions that can be reused in your plugin code are documented in *Dokumentacja API narzędzi do edytowania książek* (strona 363).

10.5.4 Adding translations to your plugin

You can have all the user interface strings in your plugin translated and displayed in whatever language is set for the main calibre user interface.

The first step is to go through your plugin's source code and mark all user visible strings as translatable, by surrounding them in `_()`. For example:

```
action_spec = (_('My plugin'), None, _('My plugin is cool'), None)
```

Then use some program to generate `.po` files from your plugin source code. There should be one `.po` file for every language you want to translate into. For example: `de.po` for German, `fr.po` for French and so on. You can use the [Poedit](https://poedit.net/)¹⁰⁰ program for this.

¹⁰⁰ <https://poedit.net/>

Send these .po files to your translators. Once you get them back, compile them into .mo files. You can again use Poedit for that, or just do:

```
calibre-debug -c "from calibre.translations.msgfmt import main; main()" filename.po
```

Put the .mo files into the `translations` folder in your plugin.

The last step is to simply call the function `load_translations()` at the top of your plugin's .py files. For performance reasons you should only call this function in those .py files that actually have translatable strings. So in a typical User Interface plugin you would call it at the top of `ui.py` but not `__init__.py`.

You can test the translations of your plugins by changing the user interface language in calibre under *Preferences* → *Interface* → *Look & feel* or by running calibre with the `CALIBRE_OVERRIDE_LANG` environment variable set. For example:

```
CALIBRE_OVERRIDE_LANG=de
```

Replace `de` with the language code of the language you want to test.

For translations with plurals, use the `ngettext()` function instead of `_()`. For example:

```
ngettext('Delete a book', 'Delete {} books', num_books).format(num_books)
```

10.5.5 The plugin API

As you may have noticed above, a plugin in calibre is a class. There are different classes for the different types of plugins in calibre. Details on each class, including the base class of all plugins can be found in [Dokumentacja API dla wtyczek](#) (strona 256).

Your plugin is almost certainly going to use code from calibre. To learn how to find various bits of functionality in the calibre code base, read the section on the calibre [Code layout](#) (strona 346).

10.5.6 Debugging plugins

The first, most important step is to run calibre in debug mode. You can do this from the command line with:

```
calibre-debug -g
```

Or from within calibre by right-clicking the *Preferences* button or using the `Ctrl+Shift+R` keyboard shortcut.

When running from the command line, debug output will be printed to the console, when running from within calibre the output will go to a txt file.

You can insert print statements anywhere in your plugin code, they will be output in debug mode. Remember, this is Python, you really shouldn't need anything more than print statements to debug ;) I developed all of calibre using just this debugging technique.

You can quickly test changes to your plugin by using the following command line:

```
calibre-debug -s; calibre-customize -b /path/to/your/plugin/folder; calibre
```

This will shutdown a running calibre, wait for the shutdown to complete, then update your plugin in calibre and relaunch calibre.

10.5.7 More plugin examples

You can find a list of many sophisticated calibre plugins [here](#)¹⁰¹.

10.5.8 Sharing your plugins with others

If you would like to share the plugins you have created with other users of calibre, post your plugin in a new thread in the [calibre plugins forum](#)¹⁰².

10.6 Skład matematyki w e-bookach

Calibre przeglądarka e-booków ma możliwość wyświetlania matematyki osadzonej w e-bookach (pliki EPUB i HTML). Możesz składać matematykę bezpośrednio za pomocą TeX, MathML lub AsciiMath. Calibre przeglądarka e-booków wykorzystuje do tego doskonałą bibliotekę [MathJax](#)¹⁰³. To jest krótki samouczek dotyczący tworzenia e-booków z matematyką, które dobrze współpracują z przeglądarką calibre e-booków.

10.6.1 Prosty plik HTML ze wzorami matematycznymi

Możesz pisać matematykę w tekście w prostym pliku HTML, a przeglądarka e-booków kalibru przekształci ją w odpowiednio złożoną matematykę. W poniższym przykładzie używamy notacji TeX do matematyki. Zobaczysz, że możesz używać normalnych poleceń TeX, z małym zastrzeżeniem, że znaki „&” oraz „mniejsze niż” i „większe niż” muszą być zapisane jako `&` i `<` i `>`; odpowiednio.

Pierwszym krokiem jest poinformowanie kalibru, że ten testament zawiera matematykę. Robisz to, dodając następujący fragment kodu do sekcji `<head>` pliku HTML:

```
<script type="text/x-mathjax-config"></script>
```

Teraz można już wpisać wzory dokładnie tak, jak to się robi w pliku `.tex`. Oto, dla przykładu, równanie Lorentza:

```
<h2>The Lorenz Equations</h2>

<p>
\begin{align}
\dot{x} \ &= \sigma(y-x) \ \backslash
\dot{y} \ &= \rho x - y - xz \ \backslash
\dot{z} \ &= -\beta z + xy
\end{align}
</p>
```

Ten fragment wygląda jak poniższy zrzut ekranu w przeglądarce e-booków calibre.

Pełny plik HTML z większą liczbą równań i wbudowaną matematyką został przedstawiony poniżej. Możesz przekonwertować ten plik HTML do formatu EPUB w kalibrze, aby otrzymać e-book, który możesz łatwo rozpowszechniać wśród innych osób.

```
<!DOCTYPE html>
<html>
<!-- Copyright (c) 2012 Design Science, Inc. -->
```

(ciąg dalszy na następnej stronie)

¹⁰¹ <https://www.mobileread.com/forums/showthread.php?t=118764>

¹⁰² <https://www.mobileread.com/forums/forumdisplay.php?f=237>

¹⁰³ <https://www.mathjax.org>

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= \rho x - y - xz \\ \dot{z} &= -\beta z + xy\end{aligned}$$

Rys. 1: *The Lorenz Equations*

(kontynuacja poprzedniej strony)

```

<head>
<title>Math Test Page</title>
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />

<!-- This script tag is needed to make calibre's ebook-viewer recognize that this_
    ↪file needs math typesetting -->
<script type="text/x-mathjax-config">
    // This line adds numbers to all equations automatically, unless explicitly_
    ↪suppressed.
    MathJax.tex = {tags: 'all'};
</script>

<style>
h1 {text-align:center}
h2 {
    font-weight: bold;
    background-color: #DDDDDD;
    padding: .2em .5em;
    margin-top: 1.5em;
    border-top: 3px solid #666666;
    border-bottom: 2px solid #999999;
}
</style>
</head>
<body>

<h1>Sample Equations</h1>

<h2>The Lorenz Equations</h2>

<p>
\begin{align}
\dot{x} \text{ \& } &= \sigma(y-x) \text{ \label{lorenz}} \\
\dot{y} \text{ \& } &= \rho x - y - xz \\
\dot{z} \text{ \& } &= -\beta z + xy
\end{align}
</p>

<h2>The Cauchy-Schwarz Inequality</h2>

<p>\[
\left( \sum_{k=1}^n a_k b_k \right)^{\{\!\!\!2\}} \leq
\left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

\]
```

A Cross Product Formula

```

<p>\[
\mathbf{V}_1 \times \mathbf{V}_2 =
\begin{vmatrix}
\mathbf{i} & \mathbf{j} & \mathbf{k} \\
\frac{\partial X}{\partial u} & \frac{\partial Y}{\partial u} & 0 \\
\frac{\partial X}{\partial v} & \frac{\partial Y}{\partial v} & 0
\end{vmatrix}
\]
```

The probability of getting k heads when flipping n coins is:

```

<p>P(E) = {n \choose k} p^k (1-p)^{n-k}
```

An Identity of Ramanujan

```

<p>\[
\frac{1}{\sqrt{\phi \sqrt{5}} - \phi} e^{\frac{25}{\pi}} =
1 + \frac{e^{-2\pi}}{1 + \frac{e^{-4\pi}}{1 + \frac{e^{-6\pi}}{1 + \frac{e^{-8\pi}}{1 + \dots}}}}
\]
```

A Rogers-Ramanujan Identity

```

<p>\[
1 + \frac{q^2}{(1-q)} + \frac{q^6}{(1-q)(1-q^2)} + \dots =
\prod_{j=0}^{\infty} \frac{1}{(1-q^{5j+2})(1-q^{5j+3})},
\quad \text{for } |q| < 1.
\]
```

Maxwell's Equations

```

<p>
\begin{align}
\nabla \times \vec{\mathbf{B}} &= \frac{1}{c} \frac{\partial \vec{\mathbf{E}}}{\partial t} \\
\nabla \cdot \vec{\mathbf{E}} &= 4 \pi \rho \\
\nabla \times \vec{\mathbf{E}} &= -\frac{1}{c} \frac{\partial \vec{\mathbf{B}}}{\partial t} \\
\nabla \cdot \vec{\mathbf{B}} &= 0
\end{align}
</p>

```

In-line Mathematics

While display equations look good for a page of samples, the ability to mix math and text in a paragraph is also important. This expression $\sqrt{3x-1} + (1+x)^2$ is an example of an inline equation. As you see, equations can be used this way as well, without unduly disturbing the spacing between lines.

References to equations

Here is a reference to the Lorenz Equations ([\ref{lorenz}](#)). Clicking on the

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
↪equation number will take you back to the equation.</p>
</body>
</html>
```

10.6.2 Więcej informacji

Ponieważ przeglądarka e-booków Calibre korzysta z biblioteki MathJax do renderowania matematyki, najlepszym miejscem, aby dowiedzieć się więcej o matematyce w e-bookach i uzyskać pomoc, jest strona internetowa «MathJax <<https://www.mathjax.org>>».

10.7 Tworzenie katalogów AZW3, EPUB, MOBI

Opcja tworzenia katalogów calibre umożliwia stworzenie katalogu książek w różnych formatach. Ten rozdział opisuje opcje tworzenia katalogów w formatach AZW3, EPUB i MOBI

- *Wybór książek do katalogu* (strona 242)
- *Włączone sekcje* (strona 243)
- *Przedrostki* (strona 244)
- *Wykluczone książki* (strona 244)
- *Wykluczone rodzaje* (strona 245)
- *Inne opcje* (strona 245)
- *Własna okładka katalogu* (strona 246)
- *Dodatkowe informacje* (strona 246)

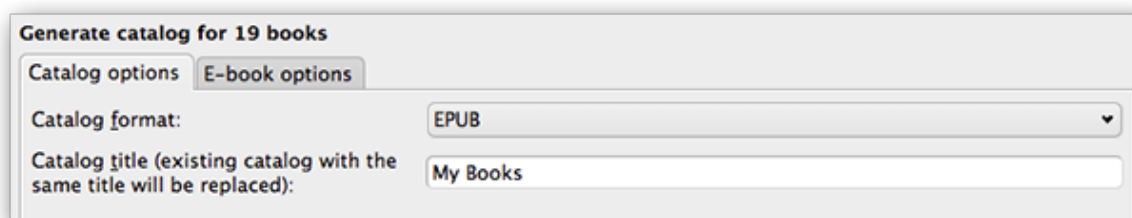
10.7.1 Wybór książek do katalogu

Jeśli chcesz skatalogować *cały* księgozbiór, usuń wszystkie filtry i kryteria wyszukiwania w głównym oknie. Jeśli tylko jedna książka jest zaznaczona na liście, wszystkie książki z biblioteki będą brane pod uwagę przy tworzeniu katalogu. Można określić dodatkowe kryteria, na podstawie których niektóre z nich zostaną wykluczone. Więcej informacji znajdziesz w sekcji *Wykluczone rodzaje* (strona 245) poniżej.

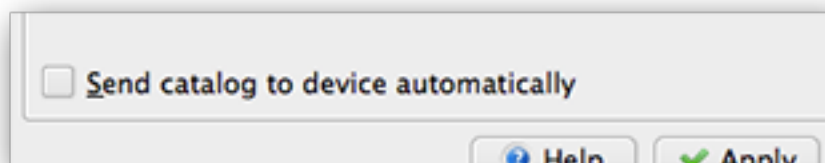
Jeśli chcesz umieścić w katalogu tylko *niektóre* książki, masz dwie możliwości:

- Zaznacz na liście te książki, które chcesz umieścić w katalogu. Jeśli w głównym oknie jest zaznaczona więcej niż jedna książka to w katalogu znajdą się wyłącznie te zaznaczone.
- Użyj pola Wyszukiwania lub przeglądarki Etykiet by filtrować wyświetlane książki. Tylko wyświetlone książki mogą być katalogowane.

Aby rozpocząć generowanie katalogu, wybierz pozycję menu :guilabel: *Konwertuj książki > Utwórz katalog książek w swojej bibliotece calibre*. Możesz także dodać przycisk *Utwórz katalog* do paska narzędzi w *Preferencje > Interfejs > Paski narzędzi i menu*, aby uzyskać łatwiejszy dostęp do okna dialogowego *Generuj katalog*.

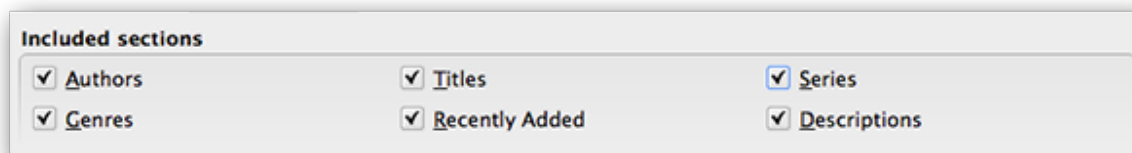


Na karcie *Opcje katalogu* wybierz **AZW3, EPUB lub MOBI** jako format katalogu. W polu *Tytuł katalogu* podaj nazwę, która zostanie użyta przy generowaniu katalogu. Jeżeli istnieje już katalog o tej samej nazwie i formacie, zostanie on zastąpiony nowo tworzonemu.



Zaznaczenie opcji *Automatycznie wyślij katalog do urządzenia* spowoduje, przesłanie katalogu do aktualnie podłączonego urządzenia zaraz po jego wygenerowaniu.

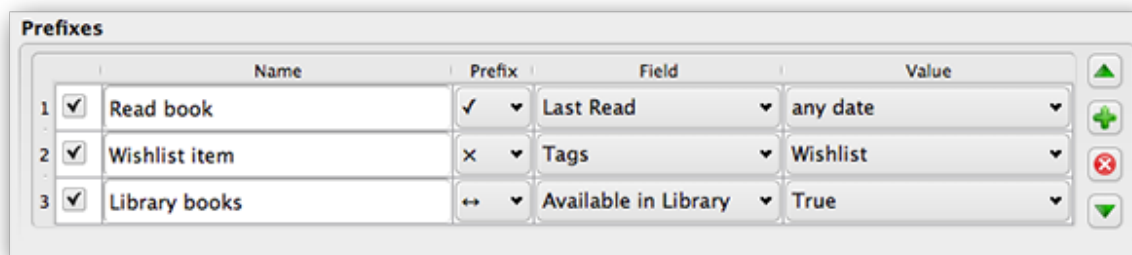
10.7.2 Włączone sekcje



Oznaczone sekcje zostaną włączone do katalogu:

- *Autorzy* - wszystkie książki, posortowane wg autora, w postaci listy. Książki będące częścią żadnego cyklu są umieszczane przed połączonymi w cykle.
- *Titles* - wszystkie książki, posortowane wg tytułu, w postaci listy.
- *Serie* - wszystkie książki, które są częścią cyklu, posortowane wg cyklu, w postaci listy.
- *Rodzaje* - poszczególne gatunki literackie w postaci listy, posortowane wg autora i cyklu
- :guilabel:"Ostatnio dodane" - wszystkie książki, posortowane w kolejności odwrotnej do chronologicznej. Lista zawiera książki dodane w ciągu ostatnich 30 dni, dalej zaś miesiąc po miesiącu.
- *Opisy* - szczegółowy opis każdej książki, z uwzględnieniem okładki i komentarza. Posortowane wg autora, książki nie będące częścią cyklu są umieszczane przed cyklami.

10.7.3 Przedrostki



Reguły przedrostków pozwalają dodać przedrostek do książek, które spełniają określone kryteria. Na przykład możesz oznaczyć książki, które już przeczytałeś, albo te, które chcesz nabyć.

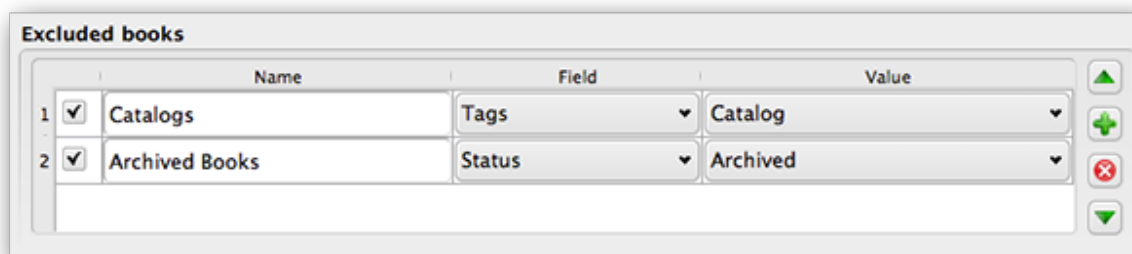
Zaznaczenia pola w pierwszej kolumnie aktywuje regułę. *Nazwa* to nazwa reguły, nadawana przez ciebie. W kolumnie *Pole* może być albo *Etykieta* albo twoja własna kolumna, utworzona w bibliotece. *Wartość* to treść *Pola*, która musi zostać odnaleziona. Jeśli warunek zostanie spełniony, książka zostanie oznaczona wybranym *Przedrostkiem*.

Trzy takie reguły zostały przedstawione w powyższym przykładzie:

1. *Read book* określa, że książka z dowolną datą w kolumnie *Last Read* zostanie oznaczona „ptaszkiem”
2. Dzięki regule *Wishlist* każda książka, zawierająca etykietę *Wishlist* zostanie oznaczona symbolem X.
3. Reguła *Library* pozwala oznaczyć podwójną strzałką każdą książkę, która ma wartość True (albo Yes) w kolumnie *Available in Library*.

Przedrostek jest określany przez pierwszą pasującą regułę. Reguły nieaktywne lub niekompletne są pomijane.

10.7.4 Wykluczone książki



Reguły wykluczające pozwalają pominąć wybrane książki.

Zaznaczenia pola w pierwszej kolumnie aktywuje regułę. *Nazwa* to nazwa reguły, nadawana przez ciebie. W kolumnie *Pole* może być albo *Etykieta* albo twoja własna kolumna, utworzona w bibliotece. *Wartość* to treść *Pola*, która musi zostać odnaleziona. Jeśli warunek zostanie spełniony, książka zostanie wykluczona z katalogu.

Powyżej zostały przedstawione dwie reguły wykluczające:

1. Reguła *Catalogs* wyłącza z katalogu książki z etykietą *Catalog*.
2. Dzięki regule *Archived Books* z katalogu zostaną wykluczone książki, które w kolumnie *Status* mają wartość *Archived*.

Wszystkie reguły są dopasowywane do każdej książki. Nieaktywne i niekompletne reguły są ignorowane.

10.7.5 Wykluczone rodzaje

Kiedy katalog jest generowany, etykiety są traktowane jak rodzaje (gatunki literackie). Jeżeli, na przykład, używasz etykiet `Fikcja i Fakt`. Etykiety te staną się osobnymi rodzajami w katalogu, z przynależnymi do nich książkami. Każda książka zostanie umieszczona we wszystkich rodzajach, zgodnie z tym, jakie miała przypisane etykiety.

Czasem stosuje się etykiety, do oznaczenia pewnych specjalnych cech na przykład `+` dla oznaczenia przeczytanych książek, albo nawiasy kwadratowe, ot takie: `[Amazon Freebie]` do oznaczenia pochodzenia książek. Wyrażenie w polu *Wykluczone rodzaje* pozwala wykluczyć te, których nie chcesz użyć w katalogu. Domyślna wartość `\\[.+\\]\\++` wyklucza z katalogu wszystkie etykiety w postaci `[tag]` oraz `+`, domyślny dla przeczytanych książek.

Można również użyć konkretnych etykiet jako wyrażenia. Na przykład `[Amazon Freebie]` lub `[Project Gutenberg]`. Jeśli chcesz umieścić kilka takich etykiet, umieść między nimi pionową kreskę: `[Amazon Freebie]|[Project Gutenberg]`.

Wynik wyrażenia pokazuje, jakie etykiety zostaną wykluczone w czasie budowania katalogu, porównując etykiety z twojej biblioteki z wyrażeniem wpisanym w pole powyżej. Wynik jest uaktualniany w miarę wpisywania wyrażenia.

10.7.6 Inne opcje

`:guilabel:„Okładka katalogu”` określa, czy wygenerować nową okładkę, czy użyć istniejącej. Możliwe jest stworzenie własnej okładki do swoich katalogów - zobacz *Własna okładka katalogu* (strona 246), aby uzyskać więcej informacji. Jeśli stworzyłeś niestandardową okładkę, którą chcesz ponownie wykorzystać, wybierz *Użyj istniejącej okładki*. W przeciwnym razie wybierz *Generuj nową okładkę*.

Extra Description note specifies a custom column's contents to be inserted into the Description page, next to the cover thumbnail. For example, you might want to display the date you last read a book using a *Last Read* custom column. For advanced use of the Description note feature, see [this post in the calibre forum](https://www.mobileread.com/forums/showpost.php?p=1335767&postcount=395)¹⁰⁴.

Thumb width specifies a width preference for cover thumbnails included with Descriptions pages. Thumbnails are cached to improve performance. To experiment with different widths, try generating a catalog with just a few books until you've determined your preferred width, then generate your full catalog. The first time a catalog is generated with a new thumbnail width, performance will be slower, but subsequent builds of the catalog will take advantage of the thumbnail cache.

¹⁰⁴ <https://www.mobileread.com/forums/showpost.php?p=1335767&postcount=395>

Merge with comments specifies a custom column whose content will be non-destructively merged with the comments metadata during catalog generation. For example, you might have a custom column *Author bio* that you'd like to append to the comments metadata. You can choose to insert the custom column contents *before or after* the comments section, and optionally separate the appended content with a horizontal rule separator. Eligible custom column types include `text`, `comments`, and `composite`.

10.7.7 Własna okładka katalogu



With the [Generate Cover plugin](#)¹⁰⁵ installed, you can create custom covers for your catalog. To install the plugin, go to *Preferences > Advanced > Plugins > Get new plugins*.

10.7.8 Dodatkowe informacje

Aby uzyskać więcej informacji na temat funkcji katalogu calibre, zobacz przyklejone forum MobileRead „Tworzenie katalogów — Rozpocznij tutaj <<https://www.mobileread.com/forums/showthread.php?t=118556>>”, gdzie można znaleźć informacje o tym, jak aby dostosować szablony katalogu i jak przesłać raport o błędzie.

Aby zadać pytania lub porozmawiać o funkcji katalogu Calibre z innymi użytkownikami, odwiedź forum MobileRead [Zarządzanie biblioteką](#)¹⁰⁶.

¹⁰⁵ <https://www.mobileread.com/forums/showthread.php?t=124219>

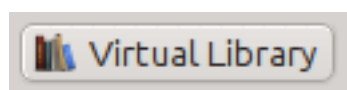
¹⁰⁶ <https://www.mobileread.com/forums/forumdisplay.php?f=236>

10.8 Wirtualne biblioteki

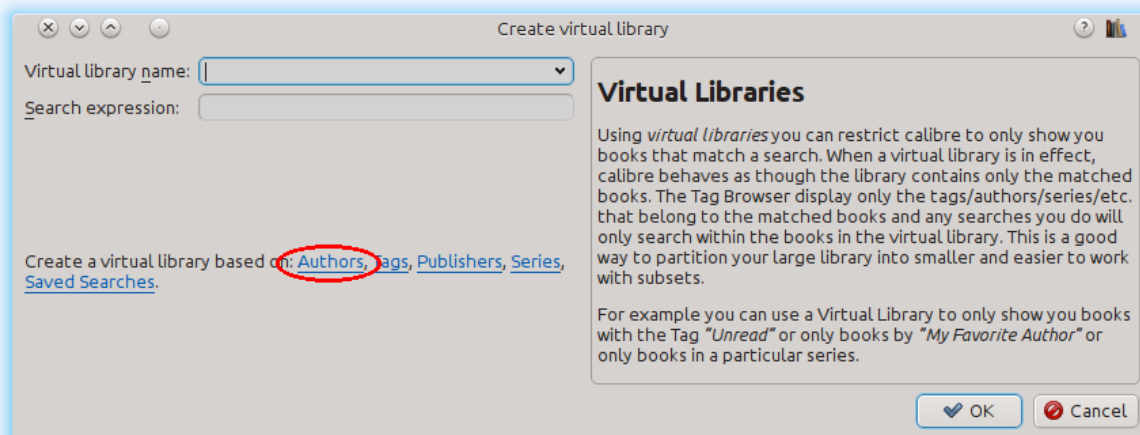
In calibre, a Virtual library is a way to tell calibre to open only a subset of a normal library. For example, you might want to only work with books by a certain author, or books having only a certain tag. Using Virtual libraries is the preferred way of partitioning your large book collection into smaller sub collections. It is superior to splitting up your library into multiple smaller libraries as, when you want to search through your entire collection, you can simply go back to the full library. There is no way to search through multiple separate libraries simultaneously in calibre.

A Virtual library is different from a simple search. A search will only restrict the list of books shown in the book list. A Virtual library does that, and in addition it also restricts the entries shown in the *Tag browser* to the left. The Tag browser will only show tags, authors, series, publishers, etc. that come from the books in the Virtual library. A Virtual library thus behaves as though the actual library contains only the restricted set of books.

10.8.1 Tworzenie wirtualnych bibliotek



To use a Virtual library click the *Virtual library* button located to the left of the Search bar and select the *Create Virtual library* option. As a first example, let's create a Virtual library that shows us only the books by a particular author. Click the *Authors* link as shown in the image below and choose the author you want to use and click OK.



Okno dialogowe Utwórz bibliotekę wirtualną zostało wypełnione za Ciebie. Kliknij OK, a zobaczysz, że nowa wirtualna biblioteka została utworzona i automatycznie przełączona, która wyświetla tylko książki wybranego autora. Jeśli chodzi o calibre, to tak, jakby twoja biblioteka zawierała tylko książki wybranego autora.

You can switch back to the full library at any time by once again clicking the *Virtual library* and selecting the entry named *<None>*.

Virtual libraries are based on *searches*. You can use any search as the basis of a Virtual library. The Virtual library will contain only the books matched by that search. First, type in the search you want to use in the Search bar or build a search using the *Tag browser*. When you are happy with the returned results, click the *Virtual library* button, choose *Create library* and enter a name for the new Virtual library. The Virtual library will then be created based on the search you just typed in. Searches are very powerful, for examples of the kinds of things you can do with them, see [The search interface](#) (strona 12).

Przykłady przydatnych wirtualnych bibliotek

- **Książki dodane wczoraj:**
date:>1daysago
- **Książki dodane w ostatnim miesiącu:**
date:>30daysago
- **Książki z oceną 5 gwiazdek:**
rating:5
- **Książki z oceną co najmniej 4 gwiazdki:**
rating:>=4
- **Książki bez oceny:**
rating:false
- **Periodyki pobrane przez calibre za pomocą funkcji pobierania wiadomości**
tags:=News and author:=calibre
- **Książki bez etykiet:**
tags:false
- **Książki bez okładek:**
cover:false

10.8.2 Posługiwanie się wirtualnymi bibliotekami

Możesz edytować wcześniej utworzoną bibliotekę wirtualną lub ją usunąć, klikając *Biblioteka wirtualna* i wybierając odpowiednią akcję.

You can tell calibre that you always want to apply a particular Virtual library when the current library is opened, by going to *Preferences* → *Interface* → *Behavior*.

You can quickly use the current search as a temporary Virtual library by clicking the *Virtual library* button and choosing the **current search* entry.

Możesz wyświetlić wszystkie dostępne biblioteki wirtualne jako zakładki nad listą książek. Jest to szczególnie przydatne, jeśli lubisz bardzo często przełączać się między bibliotekami wirtualnymi. Kliknij przycisk *Biblioteka wirtualna* i wybierz *Pokaż biblioteki wirtualne jako zakładki*. Możesz zmienić kolejność kart, przeciągając i upuszczając i zamykając te, których nie chcesz widzieć. Zamknięte karty można przywrócić, klikając prawym przyciskiem myszy pasek kart.

10.8.3 Korzystanie z wirtualnych bibliotek w wyszukiwaniu

You can search for books that are in a Virtual library using the `vl:` prefix. For example, `vl:Read` will find all the books in the *Read* Virtual library. The search `vl:Read` and `vl:"Science Fiction"` will find all the books that are in both the *Read* and *Science Fiction* Virtual libraries.

The value following `vl:` must be the name of a Virtual library. If the Virtual library name contains spaces then surround it with quotes.

One use for a Virtual library search is in the Content server. In *Preferences* → *Sharing over the net* → *Require username and password* you can limit the calibre libraries visible to a user. For each visible library you can specify a search expression to further limit which books are seen. Use `vl:"Virtual library name"` to limit the books to those in a Virtual library.

10.8.4 Dodatkowe ograniczenia

You can further restrict the books shown in a Virtual library by using *Additional restrictions*. An additional restriction is a saved search you previously created that can be applied to the current Virtual library to further restrict the books shown in a Virtual library. For example, say you have a Virtual library for books tagged as *Historical Fiction* and a saved search that shows you unread books, you can click the *Virtual Library* button and choose the *Additional restriction* option to show only unread Historical Fiction books. To learn about saved searches, see [Zapisywanie wyszukiwań](#) (strona 17).

Schemat adresu URL calibre://

calibre registers itself as the handler program for calibre:// URLs. So you can use these to perform actions like opening books, searching for books, etc from other programs/documents or via the command line. For example, running the following at the command line:

```
calibre calibre://switch-library/Some_Library
```

Otworzy calibre z biblioteką o nazwie `Some Library`. Nazwy bibliotek to nazwa folderu biblioteki ze spacjami zastąpionymi podkreśleniami. Specjalna wartość `_` oznacza bieżącą bibliotekę. Poniżej opisano różne typy adresów URL.

Możesz nawet umieścić te linki w plikach HTML, dokumentach Word lub podobnych, a system operacyjny automatycznie uruchomi kaliber, aby wykonać określoną akcję.

- *Przełącz się na konkretną bibliotekę* (strona 252)
- *Pokaż konkretną książkę w calibre* (strona 252)
- *Otwórz określoną książkę w przeglądarce e-booków w określonej pozycji* (strona 252)
- *Poszukiwanie książek* (strona 253)
- *Open a book details window on a book in some library* (strona 253)
- *Open the notes associated with an author/series/etc.* (strona 253)
- *Kodowanie szesnastkowe parametrów adresu URL* (strona 254)

11.1 Przełącz się na konkretną bibliotekę

Składnia adresu URL to:

```
calibre://switch-library/Library_Name
```

Library names are the folder name of the library with spaces replaced by underscores. The special value `_` means the current library. You can also use *hex encoding* (strona 254) for the library names, useful if the library names have special characters that would otherwise require URL encoding. Hex encoded library names look like:

```
_hex_-AD23F4BC
```

Where the part after the `_hex_-` prefix is the library name encoded as UTF-8 and every byte represented by two hexadecimal characters.

11.2 Pokaż konkretną książkę w calibre

Składnia adresu URL to:

```
calibre://show-book/Library_Name/book_id
```

This will show the book with `book_id` (a number) in calibre. The ids for books can be seen in the calibre interface by hovering over the *Click to open* link in the *Book details* panel, it is the number in brackets at the end of the path to the book folder.

You can copy a link to the current book displayed in calibre by right clicking the *Book details* panel and choosing *Copy link to book*.

If a search is active and the book is not matched by the search then the search is cleared.

If a Virtual library is selected, calibre will use it when showing the book. If the book isn't found in that virtual library then the virtual library is cleared.

If you want to switch to a particular Virtual library when showing the book, use:

```
calibre://show-book/Library_Name/book_id?virtual_library=Library%20Name
or
calibre://show-book/Library_Name/book_id?encoded_virtual_library=hex_encoded_virtual_
↳library_name
```

replacing spaces in the Virtual library name by `%20`. If the book isn't found in that virtual library then the virtual library is ignored.

11.3 Otwórz określoną książkę w przeglądarce e-booków w określonej pozycji

Składnia adresu URL to:

```
calibre://view-book/Library_Name/book_id/book_format?open_at=location
```

Here, `book_format` is the format of the book, for example, EPUB or MOBI and the `location` is an optional location inside the book. The easiest way to get these links is to open a book in the viewer, then in the viewer controls select *Go to* → *Location* and there such a link will be given that you can copy/paste elsewhere.

11.4 Poszukiwanie książek

Składnia adresu URL to:

```
calibre://search/Library_Name?q=query
calibre://search/Library_Name?eq=hex_encoded_query
```

Here query is any valid *search expression* (strona 12). If the search expression is complicated, *encode it as a hex string* (strona 254) and use eq instead. Leaving out the query will cause the current search to be cleared.

Domyślnie, jeśli wybrana jest wirtualna biblioteka, calibre wyczyści ją przed wykonaniem wyszukiwania, aby upewnić się, że wszystkie książki zostaną znalezione. Jeśli chcesz zachować wirtualną bibliotekę, użyj:

```
calibre://search/Library_Name?q=query&virtual_library=_
```

If you want to switch to a particular Virtual library, use:

```
calibre://search/Library_Name?virtual_library=Library%20Name
or
calibre://search/Library_Name?encoded_virtual_library=hex_encoded_virtual_library_name
```

replacing spaces in the Virtual library name by %20.

If you perform a search in calibre and want to generate a link for it you can do so by right clicking the search bar and choosing *Copy search as URL*.

11.5 Open a book details window on a book in some library

Składnia adresu URL to:

```
calibre://book-details/Library_Name/book_id
```

This opens a book details window on the specified book from the specified library without changing the current library or the selected book.

11.6 Open the notes associated with an author/series/etc.

Składnia adresu URL to:

```
calibre://book-details/Library_Name/Field_Name/id_Item_Id
```

This opens a window showing the notes of the specified item. The easiest way to create such URLs is to show the notes you want in calibre and click the *Copy URL* button to copy the URL to the clipboard and paste it wherever you need.

Here Field_Name is the name of the columns such as authors or tags. For user created columns, replace the leading # in the field name with an underscore, so #mytags becomes _mytags.

In addition to specifying items by id using Item_Id you can also specify them by name using either val_Item_Name or hex_Hex_Encoded_Item_Name. For example:

```
calibre://book-details/Library_Name/authors/val_John%20Doe
```

11.7 Kodowanie szesnastkowe parametrów adresu URL

Hex encoding of URL parameters is done by first encoding the parameter as UTF-8 bytes, and then replacing each byte by two hexadecimal characters representing the byte. For example, the string `abc` is the bytes `0x61` `0x62` and `0x63` in UTF-8, so the encoded version is the string: `616263`.

Dostosowywanie calibre

calibre ma wysoce modułową konstrukcję. Różne jego części można dostosować. Tutaj dowiesz się:

- jak używać zmiennych środowiskowych i *ulepszeń*, aby dostosować zachowanie calibre,
- jak określić własne zasoby statyczne, takie jak ikony i szablony, aby zastąpić wartości domyślne
- jak używać *wtyczek*, aby dodać funkcjonalność do calibre.
- jak udostępniać motywy ikon i wtyczki innym użytkownikom calibre.
- aby zobaczyć jak tworzyć *przepisy* na dodawanie nowych źródeł treści online do calibre odwiedź Sekcję *Dodawanie ulubionej strony z wiadomościami* (strona 31).

Informacja: Należy pamiętać, że chociaż motywy ikon i wtyczki są indeksowane i można je pobrać za pomocą wbudowanego aktualizatora calibre, nie są one częścią calibre, a ich kanoniczne lokalizacje wsparcia i kodu źródłowego znajdują się na [forach Mobileread](https://www.mobileread.com/forums/)¹⁰⁷ w ich wątkach wsparcia.

- *Zmienne środowiskowe* (strona 289)
- *Parametry* (strona 290)
- *Własne ikony, szablony itp.* (strona 301)
- *Tworzenie własnego motywu ikon dla calibre* (strona 301)
- *Dostosowywanie calibre za pomocą wtyczek* (strona 302)

¹⁰⁷ <https://www.mobileread.com/forums/forumdisplay.php?f=166>

12.1 Dokumentacja API dla wtyczek

Definiuje różne abstrakcyjne klasy bazowe, które można podzielić na podklasy w celu tworzenia potężnych wtyczek. Przydatnymi klasami są:

- *Wtyczka* (strona 256)
- *FileTypePlugin* (strona 258)
- *Wtyczki metadanych* (strona 260)
- *Katalog wtyczek* (strona 261)
- *Wtyczki do pobierania metadanych* (strona 262)
- *Wtyczki do konwersji* (strona 266)
- *Sterowniki urządzeń* (strona 269)
- *Akcje interfejsu użytkownika* (strona 283)
- *Preferencje wtyczki* (strona 287)

12.1.1 Wtyczka

class calibre.customize.**Plugin** (*plugin_path*)

Wtyczka calibre. Przydatni członkowie to:

- `self.installation_type`: Przechowuje sposób instalacji wtyczki.
- **`self.plugin_path`**: Przechowuje ścieżkę do pliku ZIP, który zawiera ta wtyczka lub None jeśli jest to wtyczka wbudowana
- **`self.site_customization`**: Przechowuje wprowadzony ciąg dostosowywania przez użytkownika

Metody, które powinny być nadpisane w podklasach:

- `initialize()` (strona 257)
- `customization_help()` (strona 258)

Przydatne metody:

- `temporary_file()` (strona 258)
- `__enter__()`
- `load_resources()` (strona 257)

supported_platforms = []

Lista platform, na których pracuje ta wtyczka. Na przykład:

name = 'Trivial Plugin'

Nazwa wtyczki. Musi to być coś innego niż Trivial Plugin, by wtyczka zadziałała.

version = (1, 0, 0)

Wersja wtyczki jako krotka (major, minor, revision)

description = 'Zupełnie nic nie zmienia'

Krótki string, opisujący, co robi ta wtyczka

author = 'Nieznany'

Autor wtyczki

priority = 1

Jeśli dla danego typu pliku istnieje więcej niż jedna wtyczka, wtyczki są uruchamiane w kolejności malejącego priorytetu. Wtyczki o wyższym priorytecie będą uruchamiane jako pierwsze. Najwyższym możliwym priorytetem jest `sys.maxsize`. Domyślny priorytet to 1.

minimum_calibre_version = (0, 4, 118)

Najwcześniejsza wersja calibre, wymagana przez wtyczkę.

installation_type = None

Sposób instalacji tej wtyczki

can_be_disabled = True

Jeśli False, użytkownik nie będzie mógł wyłączyć wtyczki. Używać z rozwagą.

type = 'Baza'

Typ wtyczki. Używane w celi kategoryzacji wtyczek w GUI.

initialize()

Wywoływane raz, gdy inicjowane są wtyczki Calibre. Wtyczki są ponownie inicjowane za każdym razem, gdy dodawana jest nowa wtyczka. Należy również zauważyć, że jeśli wtyczka jest uruchamiana w procesie roboczym, na przykład w celu dodawania książek, wtyczka zostanie zainicjowana dla każdego nowego procesu roboczego.

Wykonaj tutaj inicjalizację specyficzną dla wtyczki, na przykład wyodrębnij zasoby z pliku ZIP wtyczki. Ścieżka do pliku ZIP jest dostępna jako `self.plugin_path`.

Zauważ, że `self.site_customization` **nie** jest dostępna w tym miejscu.

config_widget()

Zaimplementuj tę metodę i `save_settings()` (strona 257) w swojej wtyczce, aby użyć niestandardowego okna konfiguracji, zamiast polegać na prostym domyślnym dostosowaniu opartym na ciągach znaków.

Ta metoda, jeśli jest zaimplementowana, musi zwracać `QWidget`. Widżet może mieć opcjonalną metodę `validate()`, która nie przyjmuje żadnych argumentów i jest wywoływana natychmiast po kliknięciu przez użytkownika przycisku OK. Zmiany są stosowane wtedy i tylko wtedy, gdy metoda zwraca True.

Jeśli z jakiegoś powodu nie możesz teraz przeprowadzić konfiguracji, zwróć krotkę dwóch ciągów (wiadomość, szczegóły), zostaną one wyświetlone jako okno dialogowe ostrzeżenia dla użytkownika, a proces zostanie przerwany.

save_settings(config_widget)

Zapisz ustawienia określone przez użytkownika za pomocą `config_widget`.

Parametry

config_widget – Widżet zwrócony przez `config_widget()` (strona 257).

do_user_config(parent=None)

Ta metoda pokazuje okno dialogowe konfiguracji tej wtyczki. Zwraca True, jeśli użytkownik kliknie OK, w przeciwnym razie False. Zmiany są stosowane automatycznie.

load_resources (*names*)

Jeśli ta wtyczka znajduje się w pliku ZIP (wtyczka dodana przez użytkownika), ta metoda pozwoli Ci załadować zasoby z pliku ZIP.

Na przykład, aby załadować obraz:

```

pixmap = QPixmap()
pixmap.loadFromData(self.load_resources(['images/icon.png'])['images/icon.png
↪'])
icon = QIcon(pixmap)

```

Parametry

names – Lista ścieżek do zasobów w pliku ZIP przy użyciu / jako separatora

Zwraca

Słownik w postaci {name: file_contents}. Wszelkie nazwy, które nie zostały znalezione w pliku ZIP, nie będą obecne w słowniku.

customization_help (*gui=False*)

Zwróć ciąg, który pomoże Ci dostosować tę wtyczkę. Domyślnie podnieś `NotImplementedError`, co oznacza, że wtyczka nie wymaga dostosowywania.

Jeśli ponownie zaimplementujesz tę metodę w swojej podklasie, użytkownik zostanie poproszony o wprowadzenie ciągu jako dostosowania tej wtyczki. Ciąg dostosowywania będzie dostępny jako `self.site_customization`.

Dostosowanie witryny może być dowolne, na przykład ścieżką do potrzebnego pliku binarnego na komputerze użytkownika.

Parametry

gui – Jeśli `True` zwróć pomoc HTML, w przeciwnym razie zwróć pomoc w postaci zwykłego tekstu.

temporary_file (*suffix*)

Zwróć obiekt podobny do pliku, który jest plikiem tymczasowym w systemie plików. Ten plik pozostanie dostępny nawet po zamknięciu i zostanie usunięty dopiero po zamknięciu tłumacza. Użyj elementu `name` zwróconego obiektu, aby uzyskać dostęp do pełnej ścieżki do utworzonego pliku tymczasowego.

Parametry

suffix – Przyrostek, który będzie miał plik tymczasowy.

cli_main (*args*)

Ta metoda jest głównym punktem wejścia dla interfejsu wiersza poleceń wtyczek. Jest wywoływana, gdy użytkownik to zrobi: `calibre-debug -r „Nazwa wtyczki”`. Wszystkie przekazane argumenty są obecne w zmiennej `args`.

12.1.2 FileTypePlugin

class calibre.customize.**FileTypePlugin** (*plugin_path*)

Klasy bazowe: *Plugin* (strona 256)

Wtyczka powiązana z określonym zestawem typów plików.

file_types = {}

Zestaw typów plików, dla których ta wtyczka powinna zostać uruchomiona. Użyj „*” dla wszystkich typów plików. Na przykład: {'lit', 'mobi', 'prc'}

on_import = False

Jeśli True, ta wtyczka jest uruchamiana, gdy książki są dodawane do bazy danych

on_postimport = False

Jeśli True, ta wtyczka jest uruchamiana po dodaniu książek do bazy danych. W tym przypadku wywoływane są metody `postimport` i `postadd` wtyczki.

on_postconvert = False

If True, this plugin is run after a book is converted. In this case the `postconvert` method of the plugin is called.

on_postdelete = False

If True, this plugin is run after a book file is deleted from the database. In this case the `postdelete` method of the plugin is called.

on_preprocess = False

Jeśli True, ta wtyczka jest uruchamiana tuż przed konwersją

on_postprocess = False

Jeśli True, ta wtyczka jest uruchamiana po konwersji na końcowym pliku utworzonym przez wtyczkę konwersji wyjściowej.

type = 'Typ pliku'

Typ wtyczki. Używane w celu kategoryzacji wtyczek w GUI.

run (path_to_ebook)

Uruchom wtyczkę. Musi być zaimplementowana w podklasach. Powinien wykonywać wszelkie modyfikacje wymagane w e-booku i zwracać bezwzględną ścieżkę do zmodyfikowanego e-booka. Jeśli nie są potrzebne żadne modyfikacje, powinien zwrócić ścieżkę do oryginalnego e-booka. Jeśli zostanie napotkany błąd, powinien zgłosić wyjątek. Domyślna implementacja po prostu zwraca ścieżkę do oryginalnego e-booka. Zwróć uwagę, że ścieżka do oryginalnego pliku (przed uruchomieniem wtyczek typu pliku jest dostępna jako `self.original_path_to_file`).

Zmodyfikowany plik e-booka należy utworzyć za pomocą metody `temporary_file()`.

Parametry

path_to_ebook – Absolutna ścieżka do e-booka.

Zwraca

Absolutna ścieżka do zmodyfikowanego e-booka.

postimport (book_id, book_format, db)

Nazywany post importem, czyli po dodaniu pliku książki do bazy danych. Zauważ, że różni się to od `postadd()` (strona 260), które jest wywoływane, gdy rekord książki jest tworzony po raz pierwszy. Ta metoda jest wywoływana za każdym razem, gdy do rekordu księgowego dodawany jest nowy plik. Przydaje się do modyfikowania ewidencji księgowej na podstawie zawartości nowo dodanego pliku.

Parametry

- **book_id** – Identyfikator bazy dodanej książki.
- **book_format** – Typ pliku dodanej książki.
- **db** – Biblioteczna baza danych.

postconvert (book_id, book_format, db)

Called post conversion, i.e., after the conversion output book file has been added to the database. Note that it is run after a conversion only, not after a book is added. It is useful for modifying the book record based on the contents of the newly added file.

Parametry

- **book_id** – Identyfikator bazy dodanej książki.
- **book_format** – Typ pliku dodanej książki.
- **db** – Biblioteczna baza danych.

postdelete (*book_id, book_format, db*)

Called post deletion, i.e., after the book file has been deleted from the database. Note that it is not run when a book record is deleted, only when one or more formats from the book are deleted. It is useful for modifying the book record based on the format of the deleted file.

Parametry

- **book_id** – Identyfikator bazy dodanej książki.
- **book_format** – Typ pliku dodanej książki.
- **db** – Biblioteczna baza danych.

postadd (*book_id, fmt_map, db*)

Nazwany post add, czyli po dodaniu książki do bazy. Zauważ, że różni się to od `postimport()` (strona 259), który jest wywoływany po dodaniu pojedynczego pliku książki do książki. `postadd()` jest wywoływana tylko wtedy, gdy cały rekord książki z prawdopodobnie więcej niż jednym plikiem książki został utworzony po raz pierwszy. Jest to przydatne, jeśli chcesz zmodyfikować rekord książki w bazie danych, gdy książka jest po raz pierwszy dodawana do calibre.

Parametry

- **book_id** – Identyfikator bazy dodanej książki.
- **fmt_map** – Mapa formatu pliku na ścieżkę, z której dodano format pliku. Zwróć uwagę, że może to wskazywać, ale nie musi, na rzeczywisty istniejący plik, ponieważ czasami pliki są dodawane jako strumienie. W takim przypadku może to być fikcyjna wartość lub nieistniejąca ścieżka.
- **db** – Baza danych biblioteki

12.1.3 Wtyczki metadanych

```
class calibre.customize.MetadataReaderPlugin(*args, **kwargs)
```

Klasy bazowe: `Plugin` (strona 256)

Wtyczka implementująca odczytywanie metadanych z zestawu typów plików.

```
file_types = {}
```

Zestaw typów plików, dla których ta wtyczka powinna zostać uruchomiona. Na przykład: `set(['lit', 'mobi', 'prc'])`

```
supported_platforms = ['windows', 'osx', 'linux']
```

Lista platform, na których pracuje ta wtyczka. Na przykład:

```
version = (7, 9, 0)
```

Wersja wtyczki jako krotka (major, minor, revision)

```
author = 'Kovid Goyal'
```

Autor wtyczki

```
type = 'Czytnik metadanych'
```

Typ wtyczki. Używane w celi kategoryzacji wtyczek w GUI.

get_metadata (*stream, type*)

Zwróć metadane dla pliku reprezentowanego przez strumień (plik podobny do obiektu, który obsługuje odczyt). Zgłoś wyjątek, gdy wystąpi błąd w danych wejściowych.

Parametry

type – Typ pliku. Gwarantowany jeden z wpisów w *file_types* (strona 260).

Zwraca

A `calibre.ebooks.metadata.book.Metadata` obiekt

class `calibre.customize.MetadataWriterPlugin` (**args, **kwargs*)

Klasy bazowe: *Plugin* (strona 256)

Wtyczka implementująca odczytywanie metadanych z zestawu typów plików.

file_types = {}

Zestaw typów plików, dla których ta wtyczka powinna zostać uruchomiona. Na przykład: `set(['lit', 'mobi', 'prc'])`

supported_platforms = ['windows', 'osx', 'linux']

Lista platform, na których pracuje ta wtyczka. Na przykład:

version = (7, 9, 0)

Wersja wtyczki jako krotka (major, minor, revision)

author = 'Kovid Goyal'

Autor wtyczki

type = 'Zapisywanie metadanych'

Typ wtyczki. Używane w celi kategoryzacji wtyczek w GUI.

set_metadata (*stream, mi, type*)

Ustaw metadane dla pliku reprezentowanego przez strumień (plik podobny do obiektu, który obsługuje odczyt). Zgłoś wyjątek, gdy wystąpi błąd w danych wejściowych.

Parametry

- **type** – Typ pliku. Gwarantowany jeden z wpisów w *file_types* (strona 261).
- **mi** – A `calibre.ebooks.metadata.book.Metadata` obiekt

12.1.4 Katalog wtyczek

class `calibre.customize.CatalogPlugin` (*plugin_path*)

Klasy bazowe: *Plugin* (strona 256)

Wtyczka implementująca generator katalogów.

file_types = {}

Typ pliku wyjściowego, dla którego ta wtyczka powinna zostać uruchomiona. Na przykład: „epub” lub „xml”

type = 'Generator katalogu'

Typ wtyczki. Używane w celi kategoryzacji wtyczek w GUI.

cli_options = []

Opcje parsera CLI specyficzne dla tej wtyczki, zadeklarowane jako *namedtuple Option*:

```
from collections import namedtuple
Option = namedtuple('Option', 'option default dest help')
cli_options = [Option('catalog-title', default = 'My Catalog', dest = 'catalog_title', help =
```

```
(_(«Title of generated catalog. nDefault:») + „ „» + «%default» + „»”)) cli_options parsed in  
calibre.db.cli.cmd_catalog:option_parser()
```

initialize()

Jeśli wtyczka nie jest wbudowana, skopiuj pliki .ui i .py wtyczki z pliku ZIP do \$TMPDIR. Karta zostanie dynamicznie wygenerowana i dodana do okna dialogowego Opcje katalogu w calibre.gui2.dialogs.catalog.py:Catalog

run (*path_to_output*, *opts*, *db*, *ids*, *notification=None*)

Uruchom wtyczkę. Musi być zaimplementowana w podklasach. Powinien generować katalog w formacie określonym w *file_types*, zwracając bezwzględną ścieżkę do wygenerowanego pliku katalogu. Jeśli zostanie napotkany błąd, powinien zgłosić wyjątek.

Wygenerowany plik katalogu należy utworzyć za pomocą metody `temporary_file()`.

Parametry

- **path_to_output** – Ścieżka bezwzględna do wygenerowanego pliku katalogu.
- **opts** – Słownik argumentów słów kluczowych
- **db** – Obiekt `LibraryDatabase2`

12.1.5 Wtyczki do pobierania metadanych

```
class calibre.ebooks.metadata.sources.base.Source (*args, **kwargs)
```

Klasy bazowe: *Plugin* (strona 256)

type = 'Źródło metadanych'

Typ wtyczki. Używane w celu kategoryzacji wtyczek w GUI.

author = 'Kovid Goyal'

Autor wtyczki

supported_platforms = ['windows', 'osx', 'linux']

Lista platform, na których pracuje ta wtyczka. Na przykład:

capabilities = frozenset({})

Zestaw możliwości obsługiwanych przez tę wtyczkę. Przydatne możliwości to: «identify», «cover»

touched_fields = frozenset({})

Lista pól metadanych, które potencjalnie mogą zostać pobrane przez tę wtyczkę podczas fazy identyfikacji

has_html_comments = False

Ustaw to na True, jeśli twoja wtyczka zwraca komentarze w formacie HTML

supports_gzip_transfer_encoding = False

Ustawienie tego na True oznacza, że obiekt przeglądarki wskaże, że obsługuje kodowanie transferu gzip. Może to przyspieszyć pobieranie, ale najpierw upewnij się, że źródło faktycznie obsługuje poprawnie kodowanie transferu gzip

ignore_ssl_errors = False

Ustaw to na True, aby ignorować błędy certyfikatu HTTPS podczas łączenia się z tym źródłem.

cached_cover_url_is_reliable = True

Buforowane adresy URL okładki mogą czasami być niewiarygodne (tj. pobieranie może się nie powieść lub zwrócony obraz może być fałszywy). Jeśli często tak jest w przypadku tego źródła, ustaw wartość False

options = ()

Lista obiektów `Option`. Zostaną one użyte do automatycznego skonstruowania widżetu konfiguracyjnego dla tej wtyczki

config_help_message = None

Ciąg, który jest wyświetlany u góry widżetu konfiguracyjnego dla tej wtyczki

can_get_multiple_covers = False

Jeśli `True`, to źródło może zwrócić wiele okładek dla danego zapytania

auto_trim_covers = False

Jeśli ustawione na `True`, okładki pobrane przez tę wtyczkę są automatycznie przycinane.

prefer_results_with_isbn = True

Jeśli ustawione na `True`, a to źródło zwróci wiele wyników dla zapytania, z których niektóre mają numery ISBN, a inne nie, wyniki bez numerów ISBN zostaną zignorowane

is_configured()

Zwróć `False`, jeśli wtyczka musi zostać skonfigurowana przed użyciem. Na przykład może potrzebować nazwy użytkownika/hasła/klucza API.

customization_help()

Zwróć ciąg, który pomoże Ci dostosować tę wtyczkę. Domyślnie podnieś `NotImplementedError`, co oznacza, że wtyczka nie wymaga dostosowywania.

Jeśli ponownie zaimplementujesz tę metodę w swojej podklasie, użytkownik zostanie poproszony o wprowadzenie ciągu jako dostosowania tej wtyczki. Ciąg dostosowywania będzie dostępny jako `self.site_customization`.

Dostosowanie witryny może być dowolne, na przykład ścieżką do potrzebnego pliku binarnego na komputerze użytkownika.

Parametry

gui – Jeśli `True` zwróć pomoc HTML, w przeciwnym razie zwróć pomoc w postaci zwykłego tekstu.

config_widget()

Zaimplementuj tę metodę i `save_settings()` (strona 263) w swojej wtyczce, aby użyć niestandardowego okna konfiguracji, zamiast polegać na prostym domyślnym dostosowaniu opartym na ciągach znaków.

Ta metoda, jeśli jest zaimplementowana, musi zwracać `QWidget`. Widżet może mieć opcjonalną metodę `validate()`, która nie przyjmuje żadnych argumentów i jest wywoływana natychmiast po kliknięciu przez użytkownika przycisku OK. Zmiany są stosowane wtedy i tylko wtedy, gdy metoda zwraca `True`.

Jeśli z jakiegoś powodu nie możesz teraz przeprowadzić konfiguracji, zwróć krotkę dwóch ciągów (wiadomość, szczegóły), zostaną one wyświetlone jako okno dialogowe ostrzeżenia dla użytkownika, a proces zostanie przerwany.

save_settings(config_widget)

Zapisz ustawienia określone przez użytkownika za pomocą `config_widget`.

Parametry

config_widget – Widżet zwrócony przez `config_widget()` (strona 263).

get_author_tokens(authors, only_first_author=True)

Weź listę autorów i zwróć listę tokenów przydatnych dla zapytania wyszukiwania AND. Ta funkcja próbuje zwracać tokeny w kolejności imion, drugich imion i nazwisk, zakładając, że jeśli w nazwisku autora występuje przecinek, to imię jest w nazwisku, pozostałe nazwiska tworzą się.

get_title_tokens (*title, strip_joiners=True, strip_subtitle=False*)

Weź tytuł i zwróć listę tokenów przydatnych dla zapytania wyszukiwania AND. Nie obejmuje spójników (opcjonalnie) i interpunkcji.

split_jobs (*jobs, num*)

Podziel listę zadań na maksymalnie num grupy, tak równomiernie, jak to możliwe

test_fields (*mi*)

Zwróć pierwsze pole z self.touched_fields, które ma wartość null w obiekcie mi

clean_downloaded_metadata (*mi*)

Wywołaj tę metodę w metodzie identyfikacji wtyczki, aby znormalizować metadane przed umieszczeniem obiektu Metadata w kolejce wyników. Możesz oczywiście użyć niestandardowego algorytmu dopasowanego do Twojego źródła metadanych.

get_book_url (*identifiers*)

Zwróć krotkę 3 lub Brak. Trójka ma postać: (identifier_type, identifier_value, URL). URL to adres URL książki identyfikowany przez identyfikatory w tym źródle. identifier_type, identifier_value określa identyfikator odpowiadający adresowi URL. Ten adres URL musi być możliwy do przeglądania przez człowieka za pomocą przeglądarki. Ma on na celu udostępnienie użytkownikowi klikalnego linku do łatwego odwiedzania strony książek w tym źródle. Jeśli nie zostanie znaleziony żaden adres URL, zwróć Brak. Ta metoda musi być szybka i spójna, więc zaimplementuj ją tylko wtedy, gdy możliwe jest skonstruowanie adresu URL ze znanego schematu z podanymi identyfikatorami.

get_book_url_name (*idtype, idval, url*)

Zwraca nazwę czytelną dla człowieka z wartości zwracanej przez get_book_url().

get_book_urls (*identifiers*)

Override this method if you would like to return multiple URLs for this book. Return a list of 3-tuples. By default this method simply calls `get_book_url()` (strona 264).

get_cached_cover_url (*identifiers*)

Return cached cover URL for the book identified by the identifiers dictionary or None if no such URL exists.

Pamiętaj, że ta metoda musi zwracać tylko zweryfikowane adresy URL, tj. nie adresy URL, które mogą skutkować ogólnym obrazem okładki lub błędem „nie znaleziono”.

id_from_url (*url*)

Przeanalizuj adres URL i zwróć krotkę postaci: (identifier_type, identifier_value). Jeśli adres URL nie pasuje do wzorca źródła metadanych, zwróć Brak.

identify_results_keygen (*title=None, authors=None, identifiers={}*)

Zwraca funkcję używaną do wygenerowania klucza, który może sortować obiekty metadanych według ich trafności na podstawie zapytania wyszukiwania (title, authors, identifiers).

Te klucze są używane do sortowania wyników wywołania `identify()` (strona 264).

Aby uzyskać szczegółowe informacje na temat domyślnego algorytmu, zobacz `InternalMetadataCompareKeyGen` (strona 265). Ponownie zaimplementuj tę funkcję we wtyczce, jeśli domyślny algorytm nie jest odpowiedni.

identify (*log, result_queue, abort, title=None, authors=None, identifiers={}, timeout=30*)

Zidentyfikuj książkę według tytułu/autora/ISBN/itd.

Jeśli określono identyfikatory i nie zostanie znalezione żadne dopasowanie, a to źródło metadanych nie przechowuje wszystkich powiązanych identyfikatorów (na przykład wszystkich numerów ISBN książki), ta metoda powinna ponowić próbę tylko z tytułem i autorem (zakładając, że zostały one określone).

Jeśli to źródło metadanych udostępnia również okładki, adres URL okładki powinien być buforowany, aby kolejne wywołanie interfejsu API pobierania okładek z tym samym ISBN/specjalnym identyfikatorem nie wymagało ponownego pobrania adresu URL okładki. Użyj do tego buforowania API.

Każdy obiekt Metadata umieszczony w kolejce wyników za pomocą tej metody musi mieć atrybut *source_relevance* będący liczbą całkowitą wskazującą kolejność, w jakiej wyniki zostały zwrócone przez źródło metadanych dla tego zapytania. Ta liczba będzie używana przez `compare_identify_results()`. Jeśli kolejność jest nieistotna, ustaw ją na zero dla każdego wyniku.

Upewnij się, że wszelkie informacje o mapowaniu okładki/ISBN są buforowane, zanim obiekt Metadata zostanie umieszczony w `result_queue`.

Parametry

- **log** – Obiekt dziennika, użyj go do wyprowadzenia informacji/błędów debugowania
- **result_queue** – Kolejka wyników, wyniki powinny być w niej umieszczone. Każdy wynik jest obiektem metadanych
- **abort** – Jeśli `abort.is_set()` zwraca True, przerwij dalsze przetwarzanie i wróć tak szybko, jak to możliwe
- **title** – Tytuł książki może brzmieć Brak
- **authors** – Lista autorów książki, może być Brak
- **identifiers** – Słownik innych identyfikatorów, najczęściej {«isbn»:»1234...»}
- **timeout** – Limit czasu w sekundach, żadne żądanie sieciowe nie powinno zawieszać się dłużej niż limit czasu.

Zwraca

Brak, jeśli nie wystąpiły żadne błędy, w przeciwnym razie reprezentacja błędu w kodzie Unicode odpowiednia do wyświetlenia użytkownikowi

download_cover (*log, result_queue, abort, title=None, authors=None, identifiers={}, timeout=30, get_best_cover=False*)

Pobierz okładkę i wstaw ją do kolejki wyników. Wszystkie parametry mają takie samo znaczenie jak dla `identify()` (strona 264). Umieść (`self, cover_data`) w `result_queue`.

Ta metoda powinna wykorzystywać buforowane adresy URL okładki, aby zapewnić wydajność, gdy tylko jest to możliwe. Gdy dane w pamięci podręcznej nie są obecne, większość wtyczek po prostu wywołuje identyfikację i wykorzystuje jej wyniki.

Jeśli parametr `get_best_cover` ma wartość True, a ta wtyczka może otrzymać wiele okładek, powinna otrzymać tylko „najlepszą”.

```
class calibre.ebooks.metadata.sources.base.InternalMetadataCompareKeyGen (mi,
                                                                    source_plugin,
                                                                    title,
                                                                    authors,
                                                                    identifiers)
```

Wygeneruj klucz sortowania w celu porównania istotności obiektów metadanych dla danego zapytania wyszukiwania. Służy tylko do porównywania wyników z tego samego źródła metadanych, a nie z różnych źródeł.

Klucz sortowania zapewnia, że sortowanie w porządku rosnącym to sortowanie według kolejności malejącej istotności.

Algorytm to:

- Preferuj wyniki, które mają co najmniej jeden identyfikator taki sam jak dla zapytania

- Preferuj wyniki z adresem URL okładki w pamięci podręcznej
- Preferuj wyniki z wypełnionymi wszystkimi dostępnymi polami
- Preferuj wyniki w tym samym języku, co bieżący język interfejsu użytkownika
- Preferuj wyniki, które dokładnie odpowiadają tytułowi zapytania
- Preferuj wyniki z dłuższymi komentarzami (dłuższymi o ponad 10%)
- **Użyj trafności wyniku zgodnie z wynikami wyszukiwania w źródle metadanych**
silnik

12.1.6 Wtyczki do konwersji

class calibre.customize.conversion.InputFormatPlugin(*args)

Klasy bazowe: *Plugin* (strona 256)

InputFormatPlugins są odpowiedzialne za konwersję dokumentu do formatu HTML+OPF+CSS+etc. Wyniki konwersji *muszą* być zakodowane w UTF-8. Główna akcja odbywa się w *convert()* (strona 267).

type = 'Źródłowy format do konwersji'

Typ wtyczki. Używane w celi kategoryzacji wtyczek w GUI.

can_be_disabled = False

Jeśli False, użytkownik nie będzie mógł wyłączyć wtyczki. Używać z rozważą.

supported_platforms = ['windows', 'osx', 'linux']

Lista platform, na których pracuje ta wtyczka. Na przykład:

file_types = {}

Zestaw typów plików, dla których ta wtyczka powinna być uruchamiana. Na przykład: `set(['azw', 'mobi', 'prc'])`

is_image_collection = False

Jeśli True, ta wtyczka wejściowa generuje kolekcję obrazów, po jednym na plik HTML. Można to ustawić dynamicznie w metodzie konwersji, jeśli pliki wejściowe mogą być zarówno kolekcjami obrazów, jak i kolekcjami innymi niż obrazy. Jeśli ustawisz to na True, musisz zaimplementować metodę *get_images()*, która zwraca listę obrazów.

core_usage = 1

Liczba rdzeni procesora używanych przez tę wtyczkę. Wartość -1 oznacza, że wykorzystuje wszystkie dostępne rdzenie

for_viewer = False

Jeśli ustawiono na True, wtyczka wejściowa wykona specjalne przetwarzanie, aby dane wyjściowe nadawały się do przeglądania

output_encoding = 'utf-8'

Kodowanie, w którym ta wtyczka wejściowa tworzy pliki. Wartość Brak oznacza, że kodowanie jest niezdefiniowane i musi być wykrywane indywidualnie

common_options = {<calibre.customize.conversion.OptionRecommendation object>}

Opcje wspólne dla wszystkich wtyczek formatu wejściowego. Nie zastępuj w podklasach. Zamiast tego użyj *options* (strona 266). Każda opcja musi być instancją *OptionRecommendation*.

options = {}

Opcje dostosowania zachowania tej wtyczki. Każda opcja musi być instancją `OptionRecommendation`.

recommendations = {}

Zestaw 3 krotek formularza (`option_name`, `recommended_value`, `recommendation_level`)

get_images ()

Zwróć listę bezwzględnych ścieżek do obrazów, jeśli ta wtyczka wejściowa reprezentuje kolekcję obrazów. Lista obrazów jest w tej samej kolejności co kręgosłup i spis treści.

convert (stream, options, file_ext, log, accelerators)

Ta metoda musi być zaimplementowana w podklasach. Musi zwrócić ścieżkę do utworzonego pliku OPF lub instancji `OEBBook`. Wszystkie dane wyjściowe powinny znajdować się w bieżącym folderze. Jeśli ta wtyczka tworzy pliki poza bieżącym folderem, należy je usunąć/oznaczyć do usunięcia, zanim ta metoda powróci.

Parametry

- **stream** – Plik podobny do obiektu, który zawiera plik wejściowy.
- **options** – Opcje dostosowania procesu konwersji. Gwarantowane posiadanie atrybutów odpowiadających wszystkim opcjom zadeklarowanym przez tę wtyczkę. Ponadto będzie miał atrybut `gadatliwy`, który przyjmuje wartości całkowite od zera w górę. Wyższe liczby oznaczają, że będziesz bardziej gadatliwy. Innym przydatnym atrybutem jest `input_profile`, który jest instancją `calibre.customize.profiles.InputProfile`.
- **file_ext** – Rozszerzenie (bez `.`) pliku wejściowego. Jest to gwarantowane, że jest to jeden z `file_types` obsługiwanych przez tę wtyczkę.
- **log** – Obiekt `calibre.utils.logging.Log`. Wszystkie dane wyjściowe powinny używać tego obiektu.
- **accelarators** – Słownik różnych informacji, które wtyczka wejściowa może łatwo uzyskać, przyspieszając kolejne etapy konwersji.

postprocess_book (oeb, opts, log)

Wywoływana, aby umożliwić wtyczce wejściowej wykonanie przetwarzania końcowego po przeanalizowaniu książki.

specialize (oeb, opts, log, output_fmt)

Wywoływane, aby umożliwić wtyczce wejściowej specjalizację przeanalizowanej książki pod kątem określonego formatu wyjściowego. Wywoływane po `postprocess_book` i przed wykonaniem jakichkolwiek przekształceń na przeanalizowanej książce.

gui_configuration_widget (parent, get_option_by_name, get_option_help, db, book_id=None)

Wywoływany w celu utworzenia widżetu używanego do konfiguracji tej wtyczki w GUI calibre. Widżet musi być instancją klasy `PluginWidget`. Zobacz przykłady wbudowanych wtyczek wejściowych.

class calibre.customize.conversion.OutputFormatPlugin (*args)

Klasy bazowe: `Plugin` (strona 256)

Wtyczki formatu wyjściowego są odpowiedzialne za konwersję dokumentu OEB (OF+HTML) do wyjściowego e-booka.

Można założyć, że dokument OEB jest zakodowany w UTF-8. Główna akcja odbywa się w `convert ()` (strona 268).

type = 'Docelowy format po konwersji'

Typ wtyczki. Używane w celi kategoryzacji wtyczek w GUI.

can_be_disabled = False

Jeśli False, użytkownik nie będzie mógł wyłączyć wtyczki. Używać z rozwagą.

supported_platforms = ['windows', 'osx', 'linux']

Lista platform, na których pracuje ta wtyczka. Na przykład:

file_type = None

Typ pliku (rozszerzenie bez początkowego okresu), który generuje ta wtyczka

common_options = {<calibre.customize.conversion.OptionRecommendation object>}

Opcje wspólne dla wszystkich wtyczek formatu wejściowego. Nie zastępuj w podklasach. Zamiast tego użyj *options* (strona 268). Każda opcja musi być instancją *OptionRecommendation*.

options = {}

Opcje dostosowania zachowania tej wtyczki. Każda opcja musi być instancją *OptionRecommendation*.

recommendations = {}

Zestaw 3 krotek formularza (*option_name*, *recommended_value*, *recommendation_level*)

property description

`str(object=»») -> str str(bytes_or_buffer[, encoding[, errors]]) -> str`

Utwórz nowy obiekt tekstowy z podanego obiektu. Jeśli określono kodowanie lub błędy, obiekt musi uwidaczniać bufor danych, który zostanie zdekodowany przy użyciu danego kodowania i obsługi błędów. W przeciwnym razie zwraca wynik obiektu `__str__()` (jeśli zdefiniowano) lub `repr(object)`. domyślnie kodowanie to `sys.getdefaultencoding()`. błędy domyślnie to «strict».

convert (*oeb_book*, *output*, *input_plugin*, *opts*, *log*)

Wyrenderuj zawartość *oeb_book* (który jest instancją `calibre.ebooks.oeb.OEBBook`) do pliku określonego przez wyjście.

Parametry

- **output** – Albo plik taki jak obiekt lub ciąg. Jeśli jest to ciąg, jest to ścieżka do folderu, który może istnieć lub nie. Wtyczka wyjściowa powinna zapisać swoje dane wyjściowe w tym folderze. Jeśli jest to plik podobny do obiektu, wtyczka wyjścia powinna zapisać swoje dane wyjściowe w pliku.
- **input_plugin** – Wtyczka wejściowa użyta na początku potoku konwersji.
- **opts** – Opcje konwersji. Gwarantowane posiadanie atrybutów odpowiadających *OptionRecommendations* tej wtyczki.
- **log** – Rejestrator. Wydrukuj komunikaty debugowania/informacje itp. za pomocą tego.

specialize_options (*log*, *opts*, *input_fmt*)

Może służyć do zmiany wartości opcji konwersji używanych przez potok konwersji.

specialize_css_for_output (*log*, *opts*, *item*, *stylizer*)

Can be used to make changes to the CSS during the CSS flattening process.

Parametry

- **item** – Przetwarzany element (plik HTML)
- **stylizer** – Obiekt Stylizer zawierający spłaszczone style elementu. Możesz uzyskać styl dla dowolnego elementu przez `stylizer.style(element)`.

gui_configuration_widget (*parent, get_option_by_name, get_option_help, db, book_id=None*)

Wywoływany w celu utworzenia widżetu używanego do konfiguracji tej wtyczki w GUI calibre. Widżet musi być instancją klasy PluginWidget. Zobacz przykłady wbudowanych wtyczek wyjściowych.

12.1.7 Sterowniki urządzeń

Klasa bazowa dla wszystkich sterowników urządzeń to *DevicePlugin* (strona 269). Jednakże, jeśli twoje urządzenie ujawnia się jako dysk USBMS dla systemu operacyjnego, powinieneś zamiast tego użyć klasy USBMS, ponieważ implementuje całą logikę niezbędną do obsługi tego rodzaju urządzeń.

class calibre.devices.interface.**DevicePlugin** (*plugin_path*)

Klasy bazowe: *Plugin* (strona 256)

Definiuje interfejs, który powinien być zaimplementowany przez backendy komunikujące się z czytnikiem e-booków.

type = 'Interfejs urządzenia'

Typ wtyczki. Używane w celi kategoryzacji wtyczek w GUI.

FORMATS = ['lrf', 'rtf', 'pdf', 'txt']

Uporządkowana lista obsługiwanych formatów

VENDOR_ID = 0

VENDOR_ID może być liczbą całkowitą, listą liczb całkowitych lub słownikiem. Jeśli jest to słownik, to musi to być słownik słowników postaci:

```
{
    integer_vendor_id : { product_id : [list of BCDs], ... },
    ...
}
```

PRODUCT_ID = 0

Liczba całkowita lub lista liczb całkowitych

BCD = None

BCD może mieć wartość Brak, aby nie rozróżniać urządzeń opartych na BCD, lub może to być lista numerów BCD wszystkich urządzeń obsługiwanych przez ten sterownik.

THUMBNAIL_HEIGHT = 68

Wysokość miniatur na urządzeniu

THUMBNAIL_COMPRESSION_QUALITY = 75

Jakość kompresji miniatur. Ustaw tę wartość bliżej 100, aby uzyskać miniatury lepszej jakości z mniejszą liczbą artefaktów kompresji. Oczywiście miniatury też się powiększają.

WANTS_UPDATED_THUMBNAILS = False

Ustaw to na True, jeśli urządzenie obsługuje aktualizowanie miniaturek okładek podczas sync_booklists. Ustawienie go na true spowoduje, że device.py odświeży miniatury okładki podczas dopasowywania książki

CAN_SET_METADATA = ['title', 'authors', 'collections']

Czy metadane w książkach można ustawić za pomocą GUI.

CAN_DO_DEVICE_DB_PLUGBOARD = False

Czy urządzenie może obsługiwać wtyczki metadanych device_db?

path_sep = '/'

Separator ścieżek dla ścieżek do książek na urządzeniu

icon = 'reader.png'

Ikona tego urządzenia

UserAnnotation

alias of Annotation

OPEN_FEEDBACK_MESSAGE = None

GUI wyświetla to jako komunikat, jeśli nie Brak na pasku stanu. Przydatne, jeśli otwieranie może zająć dużo czasu

VIRTUAL_BOOK_EXTENSIONS = frozenset({})

Zestaw rozszerzeń, które są „wirtualnymi książkami” na urządzeniu i dlatego nie można ich przeglądać/zapisywać/dodawać do biblioteki. Na przykład: `frozenset(['kobo'])`

VIRTUAL_BOOK_EXTENSION_MESSAGE = None

Wiadomość do wyświetlenia użytkownikowi w przypadku rozszerzeń książek wirtualnych.

NUKE_COMMENTS = None

Czy odrzucać komentarze w kopii książki wysłanej na urządzenie. Jeśli nie Brak, powinien to być krótki ciąg znaków, którym zostaną zastąpione komentarze.

MANAGES_DEVICE_PRESENCE = False

Jeśli True wskazuje, że ten sterownik całkowicie zarządza wykrywaniem urządzenia, wysuwaniem i tak dalej. Jeśli ustawisz to na True, *musisz* zaimplementować metody `detect_managed_devices` i `debug_managed_device_detection`. Sterownik z tym ustawieniem na true jest odpowiedzialny za wykrywanie urządzeń, zarządzanie czarną listą urządzeń, listą usuniętych urządzeń i tak dalej. calibre będzie okresowo wywoływał metodę `detect_managed_devices()`, a jeśli zwróci wykryte urządzenie, calibre wywoła metodę `open()`. `open()` będzie wywoływane za każdym razem, gdy zwracane jest urządzenie, nawet jeśli poprzednie wywołania `open()` nie powiodły się, dlatego sterownik musi utrzymywać własną czarną listę nieudanych urządzeń. Podobnie, podczas wysuwania, kaliber wywoła `eject()`, a następnie zakładając, że następne wywołanie `detect_managed_devices()` zwróci None, wywoła `post_yank_cleanup()`.

SLOW_DRIVEINFO = False

Jeśli ustawione na True, calibre wywoła metodę `get_driveinfo()` (strona 272) po załadowaniu list książek w celu uzyskania informacji o dysku.

ASK_TO_ALLOW_CONNECT = False

Jeśli ustawione na True, calibre zapyta użytkownika, czy chce zarządzać urządzeniem z calibre, przy pierwszym wykryciu. Jeśli ustawisz to na True, *musisz* zaimplementować `get_device_uid()` (strona 275) i `ignore_connected_device()` (strona 275) oraz `get_user_blacklisted_devices()` (strona 275) i `set_user_blacklisted_devices()` (strona 275)

user_feedback_after_callback = None

Ustaw to na słownik w postaci `{«title»:title, «msg»:msg, «det_msg»:detailed_msg}`, aby calibre wyświetlał komunikat dla użytkownika po uruchomieniu niektórych wywołań zwrotnych (obecnie tylko `upload_books`). Uważaj, aby nie spamować użytkownika zbyt dużą liczbą wiadomości. Ta zmienna jest sprawdzana po *każdym* wywołaniu zwrotnym, więc ustawiaj ją tylko wtedy, gdy naprawdę potrzebujesz.

classmethod get_open_popup_message()

GUI wyświetla to jako niemodalne wyskakujące okienko. Powinna być instancją `OpenPopupMessage`

is_usb_connected (*devices_on_system, debug=False, only_presence=False*)

Zwróć True, device_info, jeśli urządzenie obsługiwane przez tę wtyczkę jest aktualnie podłączone.

Parametry

devices_on_system – Lista aktualnie podłączonych urządzeń

detect_managed_devices (*devices_on_system, force_refresh=False*)

Wywoływane tylko wtedy, gdy MANAGES_DEVICE_PRESENCE ma wartość True.

Skanuj w poszukiwaniu urządzeń, które obsługuje ten sterownik. Powinna zwrócić obiekt urządzenia, jeśli urządzenie zostanie znalezione. Ten obiekt zostanie przekazany do metody open() jako connect_device. Jeśli nie zostanie znalezione żadne urządzenie, zwróć Brak. Zwrócony obiekt może być dowolny, kaliber go nie używa, jest przekazywany tylko do open().

Ta metoda jest wywoływana okresowo przez GUI, więc upewnij się, że nie wymaga zbyt dużych zasobów. Użyj pamięci podręcznej, aby uniknąć wielokrotnego skanowania systemu.

Parametry

- **devices_on_system** – Zbiór urządzeń USB znalezionych w systemie.
- **force_refresh** – Jeśli True, a sterownik używa pamięci podręcznej, aby zapobiec powtórnemu skanowaniu, pamięć podręczna musi zostać opróżniona.

debug_managed_device_detection (*devices_on_system, output*)

Wywoływane tylko wtedy, gdy MANAGES_DEVICE_PRESENCE ma wartość True.

Powinien zapisywać informacje o urządzeniach wykrytych w systemie na wyjściu, które jest plikiem podobnym do obiektu.

Powinien zwrócić True, jeśli urządzenie zostało wykryte i pomyślnie otwarte, w przeciwnym razie False.

reset (*key='-1', log_packets=False, report_progress=None, detected_device=None*)

Parametry

- **key** – Klucz do odblokowania urządzenia
- **log_packets** – Jeśli true, strumień pakietów do/z urządzenia jest rejestrowany
- **report_progress** – Funkcja wywoływana z postępem % (liczba od 0 do 100) dla różnych zadań. Jeśli zostanie wywołane z -1, oznacza to, że zadanie nie ma żadnych informacji o postępie
- **detected_device** – Informacje o urządzeniu ze skanera urządzenia

can_handle_windows (*usbdevice, debug=False*)

Opcjonalna metoda wykonywania dalszych kontroli na urządzeniu, aby sprawdzić, czy ten sterownik jest w stanie je obsłużyć. Jeśli tak nie jest, powinno zwrócić False. Ta metoda jest wywoływana dopiero po dopasowaniu dostawcy, identyfikatorów produktów i bcd, dzięki czemu można przeprowadzić stosunkowo czasochłonne kontrole. Domyślna implementacja zwraca True. Ta metoda jest wywoływana tylko w systemie Windows. Zobacz także [can_handle\(\)](#) (strona 271).

Zauważ, że dla urządzeń opartych na USBMS ta metoda domyślnie deleguje do [can_handle\(\)](#) (strona 271). Więc musisz tylko nadpisać [can_handle\(\)](#) (strona 271) w swojej podklasie USBMS.

Parametry

usbdevice – Urządzenie usb zwrócone przez `calibre.devices.winusb.scan_usb_devices()`

can_handle (*device_info, debug=False*)

Unixowa wersja [can_handle_windows\(\)](#) (strona 271).

Parametry

device_info – Jest krotką (vid, pid, bcd, producent, produkt, sn)

open (*connected_device, library_uuid*)

Wykonaj inicjalizację na dowolnym urządzeniu. Wywoływane po wykryciu urządzenia, ale przed innymi funkcjami, które komunikują się z urządzeniem. Na przykład: W przypadku urządzeń, które prezentują się jako urządzenia pamięci masowej USB, ta metoda byłaby odpowiedzialna za zamontowanie urządzenia lub, jeśli urządzenie zostało zmontowane automatycznie, za sprawdzenie, gdzie zostało zamontowane. Metoda `calibre.devices.usbms.device.Device.open()` (strona 280) posiada implementację tej funkcji, która powinna służyć jako dobry przykład dla urządzeń pamięci masowej USB.

Ta metoda może zgłosić wyjątek `OpenFeedback`, aby wyświetlić komunikat dla użytkownika.

Parametry

- **connected_device** – Urządzenie, które próbujemy otworzyć. Jest to krotka z (identyfikator dostawcy, identyfikator produktu, bcd, nazwa producenta, nazwa produktu, numer seryjny urządzenia). Jednak niektóre urządzenia nie mają numeru seryjnego, a w systemie Windows obecne są tylko pierwsze trzy pola, pozostałe to Brak.
- **library_uuid** – UUID bieżącej biblioteki kalibrów. Może mieć wartość Brak, jeśli nie ma biblioteki (na przykład, gdy jest używana z wiersza poleceń).

eject ()

Odmontuj/wysuń urządzenie z systemu operacyjnego. Nie sprawdza to, czy istnieją oczekujące zadania GUI, które wymagają komunikacji z urządzeniem.

UWAGA: Ta metoda nie może być wywoływana w tym samym wątku, co pozostałe metody urządzeń.

post_yank_cleanup ()

Wywoływane, jeśli użytkownik wyciągnie urządzenie bez jego uprzedniego usunięcia.

set_progress_reporter (*report_progress*)

Ustaw funkcję raportowania informacji o postępie.

Parametry

report_progress – Funkcja wywoływana z postępu `%` (liczba od 0 do 100) dla różnych zadań. Jeśli zostanie wywołane z -1, oznacza to, że zadanie nie ma żadnych informacji o postępie

get_device_information (*end_session=True*)

Zapytaj urządzenie o informacje o urządzeniu. Zobacz `L{DeviceInfoQuery}`.

Zwraca

(nazwa urządzenia, wersja urządzenia, wersja oprogramowania na urządzeniu, typ MIME)

Krotka może opcjonalnie mieć piąty element, który jest słownikiem informacji o napędzie.

Zobacz `usbms.driver` na przykład.

get_driveinfo ()

Zwróć słownik `driveinfo`. Zwykle wywoływane z `get_device_information()`, ale jeśli ładowanie informacji o dysku jest powolne dla tego sterownika, należy ustawić `SLOW_DRIVEINFO`. W takim przypadku ta metoda zostanie wywołana przez kaliber po załadowaniu list książek. Zauważ, że nie jest on wywoływany w wątku urządzenia, więc sterownik powinien buforować informacje o dysku w metodzie `books()` i ta funkcja powinna zwracać dane z pamięci podręcznej.

card_prefix (*end_session=True*)

Zwróć dwuelementową listę prefiksu do ścieżek na kartach. Jeśli nie ma karty Brak jest ustawiony dla prefiksu karty. np. (`«/place»`, `«/place2»`) (`None`, `«/place2»`) (`«/place»`, `None`) (`None`, `None`)

total_space (*end_session=True*)

Uzyskaj całkowitą dostępną przestrzeń na punktach montowania:

1. Pamięć główna
2. Karta pamięci A
3. Karta pamięci B

Zwraca

Lista 3-elementowa z całkowitą przestrzenią w bajtach (1, 2, 3). Jeśli konkretne urządzenie nie ma żadnej z tych lokalizacji, powinno zwrócić 0.

free_space (*end_session=True*)

Uzyskaj wolne miejsce na punktach montażowych:

1. Pamięć główna
2. Karta A
3. Karta B

Zwraca

Lista 3 elementów z wolnym miejscem w bajtach (1, 2, 3). Jeśli dane urządzenie nie ma żadnej z tych lokalizacji, powinno zwrócić -1.

books (*oncard=None, end_session=True*)

Zwróć listę e-booków na urządzeniu.

Parametry

oncard – Jeśli «carda» lub «cardb» zwraca listę e-książek na określonej karcie pamięci, w przeciwnym razie zwraca listę e-książek w pamięci głównej urządzenia. Jeśli określono kartę i nie ma na niej żadnych książek, zwróć pustą listę.

Zwraca

Lista książek.

upload_books (*files, names, on_card=None, end_session=True, metadata=None*)

Prześlij listę książek na urządzenie. Jeśli plik już istnieje na urządzeniu, należy go zastąpić. Ta metoda powinna wywołać `FreeSpaceError`, jeśli na urządzeniu nie ma wystarczającej ilości wolnego miejsca. Tekst `FreeSpaceError` musi zawierać słowo „karta”, jeśli „na_karcie” nie ma wartości `Brak`, w przeciwnym razie musi zawierać słowo „pamięć”.

Parametry

- **files** – Lista ścieżek
- **names** – Lista nazw plików, które książki powinny mieć po przesłaniu na urządzenie. `dł(nazwy) == dł(pliki)`
- **metadata** – Jeśli nie `Brak`, jest to lista obiektów `Metadata`. Pomysł polega na tym, aby wykorzystać metadane do określenia, gdzie na urządzeniu umieścić książkę. `len(metadane) == len(pliki)`. Oprócz zwykłej okładki (ścieżka do okładki) może istnieć również atrybut miniatury, którego należy używać w pierwszej kolejności. Atrybut miniatury ma postać (`width, height, cover_data` jako `jpeg`).

Zwraca

Lista krotek 3-elementowych. Lista ma być przekazana do `add_books_to_metadata()` (strona 273).

classmethod add_books_to_metadata (*locations, metadata, booklists*)

Dodaj lokalizacje do list książek. Ta funkcja nie może komunikować się z urządzeniem.

Parametry

- **locations** – Wynik rozmowy z `L{upload_books}`
- **metadata** – Lista obiektów Metadata, taka sama jak dla `upload_books()` (strona 273).
- **booklists** – Krotka zawierająca wynik wywołań do (`books(ocard=None)()`, `books(ocard='carda')()`, `:meth`books(ocard=>cardb)``)).

delete_books (*paths, end_session=True*)

Usuń książki ze ścieżek na urządzeniu.

classmethod remove_books_from_metadata (*paths, booklists*)

Usuń książki z listy metadanych. Ta funkcja nie może komunikować się z urządzeniem.

Parametry

- **paths** – ścieżki do książek na urządzeniu.
- **booklists** – Krotka zawierająca wynik wywołań do (`books(ocard=None)()`, `books(ocard='carda')()`, `:meth`books(ocard=>cardb)``)).

sync_booklists (*booklists, end_session=True*)

Zaktualizuj metadane na urządzeniu.

Parametry

booklists – Krotka zawierająca wynik wywołań do (`books(ocard=None)()`, `books(ocard='carda')()`, `:meth`books(ocard=>cardb)``)).

get_file (*path, outfile, end_session=True*)

Przeczytaj plik w ścieżce na urządzeniu i zapisz go w pliku wyjściowym.

Parametry

outfile – obiekt plikowy, taki jak `sys.stdout` lub wynik wywołania `open()` (strona 272).

classmethod config_widget ()

Powinien zwrócić QWidget. QWidget zawiera ustawienia interfejsu urządzenia

classmethod save_settings (*settings_widget*)

Powinien zapisać ustawienia na dysku. Pobiera widżet utworzony w `config_widget()` (strona 274) i zapisuje wszystkie ustawienia na dysku.

classmethod settings ()

Powinna zwrócić obiekt opts. Obiekt opts powinien mieć co najmniej jeden atrybut „format_map”, który jest uporządkowaną listą formatów urządzenia.

set_plugboards (*plugboards, pb_func*)

zapewnić sterownikowi aktualny zestaw wtyczek i funkcję wyboru konkretnej wtyczek. Ta metoda jest wywoływana bezpośrednio przed `add_books` i `sync_booklists`.

pb_func jest wywoływalny z następującą sygnaturą:

```
def pb_func(device_name, format, plugboards)
```

Podajesz mu aktualną nazwę urządzenia (albo nazwę klasy lub `DEVICE_PLUGBOARD_NAME`), format, który Cię interesuje («prawdziwy» format lub «device_db») oraz wtyczki (otrzymałeś je przez `set_plugboards`, to samo miejsce, w którym masz Ta metoda).

Zwraca

Brak lub pojedyncza instancja wtyczki.

set_driveinfo_name (*location_code, name*)

Ustaw nazwę urządzenia w pliku driveinfo na „nazwa”. To ustawienie będzie obowiązywać do momentu ponownego utworzenia pliku lub ponownej zmiany nazwy.

Urządzenia niedyskowe powinny implementować tę metodę w oparciu o kody lokalizacji zwracane przez metodę `get_device_information()`.

prepare_addable_books (*paths*)

Biorąc pod uwagę listę ścieżek, zwraca inną listę ścieżek. Te ścieżki wskazują na możliwe do dodania wersje książek.

Jeśli wystąpi błąd podczas przygotowywania książki, to zamiast ścieżki, pozycja na zwróconej liście dla tej książki powinna być krotką trzech: (original_path, the exception instance, traceback)

startup ()

Wywoływane, gdy calibre uruchamia urządzenie. Wykonaj wszelkie wymagane inicjalizacje. Zauważ, że można utworzyć wiele instancji klasy, a zatem `__init__` może być wywoływany wiele razy, ale tylko jedna instancja będzie miała wywołaną tę metodę. Ta metoda jest wywoływana w wątku urządzenia, a nie w wątku GUI.

shutdown ()

Wywoływane, gdy calibre się wyłącza, na dobre lub w trakcie przygotowań do ponownego uruchomienia. Czy wymagane czyszczenie. Ta metoda jest wywoływana w wątku urządzenia, a nie w wątku GUI.

get_device_uid ()

Musi zwrócić unikalny identyfikator dla aktualnie podłączonego urządzenia (jest to wywoływane natychmiast po udanym wywołaniu funkcji `open()`). Musisz zaimplementować tę metodę, jeśli ustawisz `ASK_TO_ALLOW_CONNECT=True`

ignore_connected_device (*uid*)

Powinna ignorować urządzenie identyfikowane przez uid (wynik wywołania `get_device_uid()`) w przyszłości. Musisz zaimplementować tę metodę, jeśli ustawisz `ASK_TO_ALLOW_CONNECT=True`. Zauważ, że ta funkcja jest wywoływana natychmiast po `open()`, więc jeśli `open()` buforuje jakiś stan, sterownik powinien zresetować ten stan.

get_user_blacklisted_devices ()

Zwróć mapę identyfikatora urządzenia do przyjaznej nazwy dla wszystkich urządzeń, które użytkownik poprosił o zignorowanie.

set_user_blacklisted_devices (*devices*)

Ustaw listę identyfikatorów urządzeń, które powinny być ignorowane przez ten sterownik.

specialize_global_preferences (*device_prefs*)

Zastosuj tę metodę, jeśli Twoje urządzenie chce zastąpić określoną preferencję. Musisz upewnić się, że wszystkie strony wywołujące, które chcą ustawienia, które można nadpisać, używają `device_prefs[«coś»]` zamiast `prefs[«coś»]`. Twoja metoda powinna wywołać `device_prefs.set_overrides(pref=val, pref=val, ...)`. Obecnie używany do: zarządzania metadanymi (`prefs[«manage_device_metadata»]`)

set_library_info (*library_name, library_uuid, field_metadata*)

Zaimplementuj tę metodę, jeśli chcesz uzyskać informacje o bieżącej bibliotece kalibrów. Ta metoda jest wywoływana podczas uruchamiania i gdy biblioteka kalibru zmienia się podczas połączenia.

is_dynamically_controllable ()

Wywoływane przez menedżera urządzeń podczas uruchamiania wtyczek. Jeśli ta metoda zwraca ciąg znaków,

to a) obsługuje interfejs dynamicznego sterowania menedżera urządzeń oraz b) ta nazwa ma być używana podczas komunikacji z wtyczką.

Tę metodę można wywołać w wątku GUI. Sterownik, który implementuje tę metodę, musi być bezpieczny wątkowo.

start_plugin()

Ta metoda jest wywoływana w celu uruchomienia wtyczki. Wtyczka powinna zacząć akceptować połączenia urządzeń, jednak robi to. Jeśli wtyczka już akceptuje połączenia, nie rób nic.

Tę metodę można wywołać w wątku GUI. Sterownik, który implementuje tę metodę, musi być bezpieczny wątkowo.

stop_plugin()

Ta metoda jest wywoływana, aby zatrzymać wtyczkę. Wtyczka nie powinna już akceptować połączeń i powinna sama sprzątać. Jest prawdopodobne, że ta metoda powinna wywołać zamknięcie. Jeśli wtyczka już nie akceptuje połączeń, nie rób nic.

Tę metodę można wywołać w wątku GUI. Sterownik, który implementuje tę metodę, musi być bezpieczny wątkowo.

get_option(opt_string, default=None)

Zwraca wartość opcji wskazanej przez *opt_string*. Tę metodę można wywołać, gdy wtyczka nie jest uruchomiona. Zwróć Brak, jeśli opcja nie istnieje.

Tę metodę można wywołać w wątku GUI. Sterownik, który implementuje tę metodę, musi być bezpieczny wątkowo.

set_option(opt_string, opt_value)

Ustaw wartość opcji wskazanej przez *opt_string*. Tę metodę można wywołać, gdy wtyczka nie jest uruchomiona.

Tę metodę można wywołać w wątku GUI. Sterownik, który implementuje tę metodę, musi być bezpieczny wątkowo.

is_running()

Zwróci True, jeśli wtyczka jest uruchomiona, w przeciwnym razie false

Tę metodę można wywołać w wątku GUI. Sterownik, który implementuje tę metodę, musi być bezpieczny wątkowo.

synchronize_with_db(db, book_id, book_metadata, first_call)

Wywoływane podczas dopasowywania książek, gdy książka na urządzeniu zostanie dopasowana do książki w calibre db. Metoda odpowiada za synchronizację danych z urządzenia z bazą danych calibre (w razie potrzeby).

Metoda musi zwrócić krotkę o dwóch wartościach. Pierwsza wartość to zestaw identyfikatorów księgi calibre zmienionych, jeśli baza danych calibre została zmieniona, lub Brak, jeśli baza danych nie została zmieniona. Jeśli pierwsza wartość jest pustym zestawem, metadane książki na urządzeniu są aktualizowane za pomocą metadanych calibre i zwracane do urządzenia, ale nie jest wykonywane odświeżanie GUI tej książki. Jest to przydatne, gdy dane calibre są poprawne, ale muszą zostać przesłane do urządzenia.

Druga wartość sama w sobie jest krotką o dwóch wartościach. Pierwsza wartość w krotce określa, czy format książki powinien zostać wysłany do urządzenia. Celem jest umożliwienie sprawdzenia, czy książka na urządzeniu jest taka sama jak książka w calibre. Ta wartość musi wynosić None, jeśli żadna książka nie ma być wysłana, w przeciwnym razie zwróć nazwę pliku bazowego na urządzeniu (łańcuch, taki jak foobar.epub). Pamiętaj, aby w nazwie uwzględnić rozszerzenie. Podsystem urządzenia utworzy zadanie `send_books` dla wszystkich książek z wartościami, które nie zostały zwrócone. Uwaga: oprócz późniejszego pobrania rozszerzenia, nazwa jest ignorowana w przypadkach, gdy urządzenie używa szablonu do wygenerowania nazwy pliku, co najczęściej robi. Druga wartość w zwróconej krotce wskazuje, czy format jest datowany na

przyszłość. Zwróć True, jeśli tak, w przeciwnym razie zwróć False. Calibre wyświetli użytkownikowi okno dialogowe z listą wszystkich przyszłych datowanych książek.

Niezwykle ważne: ta metoda jest wywoływana w wątku GUI. Musi być bezpieczny wątkowo w odniesieniu do wątku menedżera urządzeń.

`book_id`: identyfikator calibre książki w bazie danych. `book_metadata`: obiekt Metadata dla książki pochodzącej z urządzenia. `first_call`: prawda, jeśli jest to pierwsze wywołanie podczas synchronizacji, w przeciwnym razie fałsz

class `calibre.devices.interface.BookList` (*oncard, prefix, settings*)

Klasy bazowe: `list`

Lista książek. Każdy obiekt książki musi mieć pola

1. tytuł
2. autorzy
3. rozmiar (rozmiar pliku książki)
4. data i godzina (czas UTC)
5. ścieżka (ścieżka na urządzeniu do książki)
6. miniatura (może być Brak) miniatura jest albo obiektem str/bytes z danymi obrazu, albo powinna mieć atrybut `image_path`, który przechowuje bezwzględną (natywną platformę) ścieżkę do obrazu
7. tagi (lista ciągów, może być pusta).

supports_collections ()

Zwróć True, jeśli urządzenie obsługuje kolekcje dla tej listy książek.

add_book (*book, replace_metadata*)

Dodaj książkę do listy książek. Intencją jest zachowanie wszelkich wewnętrznych metadanych urządzenia. Zwróć True, jeśli listy książek muszą być zsynchronizowane

remove_book (*book*)

Usuń książkę z listy książek. Popraw jednocześnie wszystkie metadane urządzenia

get_collections (*collection_attributes*)

Zwróć słownik kolekcji utworzonych na podstawie `collection_attributes`. Każdy wpis w słowniku ma postać nazwa kolekcji:[lista książek]

Lista książek jest posortowana według tytułu książki, z wyjątkiem zbiorów utworzonych z serii, w których używany jest indeks_serii.

Parametry

collection_attributes – Lista atrybutów obiektu książki

Urządzenia oparte na pamięci masowej USB

Bazową klasą dla takich urządzeń jest `calibre.devices.usbms.driver.USBMS` (strona 281). Ta klasa z kolei dziedziczy część swojej funkcjonalności ze swoich baz, udokumentowanych poniżej. Typowy podstawowy sterownik USBMS wygląda tak:

```
from calibre.devices.usbms.driver import USBMS

class PDNOVEL(USBMS):
    name = 'Pandigital Novel device interface'
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

gui_name = 'PD Novel'
description = _('Communicate with the Pandigital Novel')
author = 'Kovid Goyal'
supported_platforms = ['windows', 'linux', 'osx']
FORMATS = ['epub', 'pdf']

VENDOR_ID    = [0x18d1]
PRODUCT_ID   = [0xb004]
BCD          = [0x224]

THUMBNAIL_HEIGHT = 144

EBOOK_DIR_MAIN = 'eBooks'
SUPPORTS_SUB_DIRS = False

def upload_cover(self, path, filename, metadata):
    coverdata = getattr(metadata, 'thumbnail', None)
    if coverdata and coverdata[2]:
        with open('%s.jpg' % os.path.join(path, filename), 'wb') as coverfile:
            coverfile.write(coverdata[2])

```

class calibre.devices.usbms.device.**Device** (*plugin_path*)

Klasy bazowe: DeviceConfig, DevicePlugin (strona 269)

Ta klasa udostępnia logikę wspólną dla wszystkich sterowników urządzeń, które eksportują się jako urządzenia pamięci masowej USB. Zapewnia implementacje do montażu/wysuwania urządzeń USBMS na wszystkich platformach.

VENDOR_ID = 0

VENDOR_ID może być liczbą całkowitą, listą liczb całkowitych lub słownikiem. Jeśli jest to słownik, to musi to być słownik słowników postaci:

```

{
    integer_vendor_id : { product_id : [list of BCDs], ... },
    ...
}

```

PRODUCT_ID = 0

Liczba całkowita lub lista liczb całkowitych

BCD = None

BCD może mieć wartość Brak, aby nie rozróżniać urządzeń opartych na BCD, lub może to być lista numerów BCD wszystkich urządzeń obsługiwanych przez ten sterownik.

WINDOWS_MAIN_MEM = None

Ciąg identyfikujący pamięć główną urządzenia w ciągach identyfikatorów PnP systemu Windows. Może to być Brak, ciąg, lista ciągów lub skompilowane wyrażenie regularne.

WINDOWS_CARD_A_MEM = None

Ciąg identyfikujący pierwszą kartę urządzenia w ciągach identyfikatorów PnP systemu Windows. Może to być Brak, ciąg, lista ciągów lub skompilowane wyrażenie regularne.

WINDOWS_CARD_B_MEM = None

Ciąg identyfikujący drugą kartę urządzenia w ciągach identyfikatora PnP systemu Windows. Może to być Brak, ciąg, lista ciągów lub skompilowane wyrażenie regularne.

OSX_MAIN_MEM_VOL_PAT = None

Używany przez wykrywanie nowego sterownika do odróżnienia pamięci głównej od kart pamięci. Powinno być wyrażeniem regularnym, które pasuje do punktu montowania pamięci głównej przypisanego przez macOS

BACKLOADING_ERROR_MESSAGE = None

MAX_PATH_LEN = 250

Maksymalna długość ścieżek utworzonych na urządzeniu

NEWS_IN_FOLDER = True

Umieść wiadomości w osobnym folderze

reset (*key='-1', log_packets=False, report_progress=None, detected_device=None*)

Parametry

- **key** – Klucz do odblokowania urządzenia
- **log_packets** – Jeśli true, strumień pakietów do/z urządzenia jest rejestrowany
- **report_progress** – Funkcja wywoływana z postępem % (liczba od 0 do 100) dla różnych zadań. Jeśli zostanie wywołane z -1, oznacza to, że zadanie nie ma żadnych informacji o postępie
- **detected_device** – Informacje o urządzeniu ze skanera urządzenia

set_progress_reporter (*report_progress*)

Ustaw funkcję raportowania informacji o postępie.

Parametry

report_progress – Funkcja wywoływana z postępem % (liczba od 0 do 100) dla różnych zadań. Jeśli zostanie wywołane z -1, oznacza to, że zadanie nie ma żadnych informacji o postępie

card_prefix (*end_session=True*)

Zwróć dwuelementową listę prefiksu do ścieżek na kartach. Jeśli nie ma karty Brak jest ustawiony dla prefiksu karty. np. («/place», «/place2») (None, «place2») («place», None) (None, None)

total_space (*end_session=True*)

Uzyskaj całkowitą dostępną przestrzeń na punktach montowania:

1. Pamięć główna
2. Karta pamięci A
3. Karta pamięci B

Zwraca

Lista 3-elementowa z całkowitą przestrzenią w bajtach (1, 2, 3). Jeśli konkretne urządzenie nie ma żadnej z tych lokalizacji, powinno zwrócić 0.

free_space (*end_session=True*)

Uzyskaj wolne miejsce na punktach montażowych:

1. Pamięć główna
2. Karta A
3. Karta B

Zwraca

Lista 3 elementów z wolnym miejscem w bajtach (1, 2, 3). Jeśli dane urządzenie nie ma żadnej z tych lokalizacji, powinno zwrócić -1.

windows_sort_drives (*drives*)

Wywoływany w celu rozróżnienia pamięci głównej i karty pamięci dla urządzeń, które nie rozróżniają ich na podstawie `WINDOWS_CARD_NAME`. Na przykład: EB600

can_handle_windows (*usbdevice, debug=False*)

Opcjonalna metoda wykonywania dalszych kontroli na urządzeniu, aby sprawdzić, czy ten sterownik jest w stanie je obsłużyć. Jeśli tak nie jest, powinno zwrócić `False`. Ta metoda jest wywoływana dopiero po dopasowaniu dostawcy, identyfikatorów produktów i bcd, dzięki czemu można przeprowadzić stosunkowo czasochłonne kontrole. Domyślna implementacja zwraca `True`. Ta metoda jest wywoływana tylko w systemie Windows. Zobacz także `can_handle()`.

Zauważ, że dla urządzeń opartych na USBMS ta metoda domyślnie deleguje do `can_handle()`. Więc musisz tylko nadpisać `can_handle()` w swojej podklasie USBMS.

Parametry

usbdevice – Urządzenie usb zwrócone przez `calibre.devices.winusb.scan_usb_devices()`

open (*connected_device, library_uuid*)

Wykonaj inicjalizację na dowolnym urządzeniu. Wywoływane po wykryciu urządzenia, ale przed innymi funkcjami, które komunikują się z urządzeniem. Na przykład: W przypadku urządzeń, które prezentują się jako urządzenia pamięci masowej USB, ta metoda byłaby odpowiedzialna za zamontowanie urządzenia lub, jeśli urządzenie zostało zmontowane automatycznie, za sprawdzenie, gdzie zostało zamontowane. Metoda `calibre.devices.usbms.device.Device.open()` (strona 280) posiada implementację tej funkcji, która powinna służyć jako dobry przykład dla urządzeń pamięci masowej USB.

Ta metoda może zgłosić wyjątek `OpenFeedback`, aby wyświetlić komunikat dla użytkownika.

Parametry

- **connected_device** – Urządzenie, które próbujemy otworzyć. Jest to krotka z (identyfikator dostawcy, identyfikator produktu, bcd, nazwa producenta, nazwa produktu, numer seryjny urządzenia). Jednak niektóre urządzenia nie mają numeru seryjnego, a w systemie Windows obecne są tylko pierwsze trzy pola, pozostałe to Brak.
- **library_uuid** – UUID bieżącej biblioteki kalibrów. Może mieć wartość Brak, jeśli nie ma biblioteki (na przykład, gdy jest używana z wiersza poleceń).

eject ()

Odmontuj/wysuń urządzenie z systemu operacyjnego. Nie sprawdza to, czy istnieją oczekujące zadania GUI, które wymagają komunikacji z urządzeniem.

UWAGA: Ta metoda nie może być wywoływana w tym samym wątku, co pozostałe metody urządzeń.

post_yank_cleanup ()

Wywoływane, jeśli użytkownik wyciągnie urządzenie bez jego uprzedniego usunięcia.

sanitize_callback (*path*)

Wywołanie zwrotne umożliwiające poszczególnym sterownikom urządzeń zastąpienie oczyszczania ścieżki używanego przez `create_upload_path()`.

filename_callback (*default, mi*)

Wywołanie zwrotne umożliwiające sterownikom zmianę domyślnej nazwy pliku ustawionej przez `create_upload_path()`.

sanitize_path_components (*components*)

Wykonaj dowolne oczyszczanie specyficzne dla urządzenia na komponentach ścieżki dla plików, które mają być przesłane do urządzenia

get_annotations (*path_map*)

Zamień *path_map* na *annotation_map* plików znalezionych na urządzeniu

add_annotation_to_library (*db, db_id, annotation*)

Dodaj adnotację do biblioteki calibre

class calibre.devices.usbms.cli.CLI

class calibre.devices.usbms.driver.USBMS (*plugin_path*)

Klasy bazowe: *CLI* (strona 281), *Device* (strona 278)

Klasa bazowa dla wszystkich urządzeń USBMS. Implementuje logikę wysyłania/pobierania/aktualizowania metadanych/buforowania metadanych/itd.

description = 'Umożliwia komunikację z czytnikiem.'

Krótki string, opisujący, co robi ta wtyczka

author = 'John Schember'

Autor wtyczki

supported_platforms = ['windows', 'osx', 'linux']

Lista platform, na których pracuje ta wtyczka. Na przykład:

booklist_class

alias of BookList

book_class

alias of Book

FORMATS = []

Uporządkowana lista obsługiwanych formatów

CAN_SET_METADATA = []

Czy metadane w książkach można ustawić za pomocą GUI.

get_device_information (*end_session=True*)

Zapytaj urządzenie o informacje o urządzeniu. Zobacz L{DeviceInfoQuery}.

Zwraca

(nazwa urządzenia, wersja urządzenia, wersja oprogramowania na urządzeniu, typ MIME)

Krotka może opcjonalnie mieć piąty element, który jest słownikiem informacji o napędzie.

Zobacz *usbms.driver* na przykład.

set_driveinfo_name (*location_code, name*)

Ustaw nazwę urządzenia w pliku *driveinfo* na „nazwa”. To ustawienie będzie obowiązywać do momentu ponownego utworzenia pliku lub ponownej zmiany nazwy.

Urządzenia niedyskowe powinny implementować tę metodę w oparciu o kody lokalizacji zwracane przez metodę *get_device_information()*.

books (*oncard=None, end_session=True*)

Zwróć listę e-booków na urządzeniu.

Parametry

oncard – Jeśli «carda» lub «cardb» zwraca listę e-książek na określonej karcie pamięci, w przeciwnym razie zwraca listę e-książek w pamięci głównej urządzenia. Jeśli określono kartę i nie ma na niej żadnych książek, zwróć pustą listę.

Zwraca

Lista książek.

upload_books (*files, names, on_card=None, end_session=True, metadata=None*)

Prześlij listę książek na urządzenie. Jeśli plik już istnieje na urządzeniu, należy go zastąpić. Ta metoda powinna wywołać `FreeSpaceError`, jeśli na urządzeniu nie ma wystarczającej ilości wolnego miejsca. Tekst `FreeSpaceError` musi zawierać słowo „karta”, jeśli „na_karcie” nie ma wartości `Brak`, w przeciwnym razie musi zawierać słowo „pamięć”.

Parametry

- **files** – Lista ścieżek
- **names** – Lista nazw plików, które książki powinny mieć po przesłaniu na urządzenie. `dł(nazwy) == dł(pliki)`
- **metadata** – Jeśli nie `Brak`, jest to lista obiektów `Metadata`. Pomysł polega na tym, aby wykorzystać metadane do określenia, gdzie na urządzeniu umieścić książkę. `len(metadata) == len(pliki)`. Oprócz zwykłej okładki (ścieżka do okładki) może istnieć również atrybut miniatury, którego należy używać w pierwszej kolejności. Atrybut miniatury ma postać (`width, height, cover_data` jako `jpeg`).

Zwraca

Lista krotek 3-elementowych. Lista ma być przekazana do `add_books_to_metadata()` (strona 282).

upload_cover (*path, filename, metadata, filepath*)

Prześlij okładkę książki do urządzenia. Domyślna implementacja nic nie robi.

Parametry

- **path** – Pełna ścieżka do folderu, w którym znajduje się powiązana książka.
- **filename** – Nazwa pliku książki bez rozszerzenia.
- **metadata** – metadane należące do książki. Użyj `metadata.thumbnail` jako okładki
- **filepath** – Pełna ścieżka do pliku e-booka

add_books_to_metadata (*locations, metadata, booklists*)

Dodaj lokalizacje do list książek. Ta funkcja nie może komunikować się z urządzeniem.

Parametry

- **locations** – Wynik rozmowy z `L{upload_books}`
- **metadata** – Lista obiektów `Metadata`, taka sama jak dla `upload_books()` (strona 282).
- **booklists** – Krotka zawierająca wynik wywołań do `(books(oncard=None)(), books(oncard='carda')(), :meth'books(oncard=>cardb))`.

delete_books (*paths, end_session=True*)

Usuń książki ze ścieżek na urządzeniu.

remove_books_from_metadata (*paths, booklists*)

Usuń książki z listy metadanych. Ta funkcja nie może komunikować się z urządzeniem.

Parametry

- **paths** – ścieżki do książek na urządzeniu.
- **booklists** – Krotka zawierająca wynik wywołań do (`books(ocard=None)()`, `books(ocard='carda')()`, `:meth`books(ocard=>cardb)``)).

sync_booklists (*booklists*, *end_session=True*)

Zaktualizuj metadane na urządzeniu.

Parametry

booklists – Krotka zawierająca wynik wywołań do (`books(ocard=None)()`, `books(ocard='carda')()`, `:meth`books(ocard=>cardb)``)).

classmethod normalize_path (*path*)

Ścieżka zwrotna z natywnymi separatorami ścieżek platformy

12.1.8 Akcje interfejsu użytkownika

Jeśli dodajesz własną wtyczkę w pliku ZIP, powinieneś podklasy zarówno `InterfaceActionBase`, jak i `InterfaceAction`. Metoda `load_actual_plugin()` twojej podklasy `InterfaceActionBase` musi zwracać instancję obiektu twojej podklasy `InterfaceBase`.

class `calibre.gui2.actions.InterfaceAction` (*parent*, *site_customization*)

Klasy bazowe: `QObject`

Wtyczka reprezentująca „działanie”, które można wykonać w graficznym interfejsie użytkownika. Wszystkie elementy na pasku narzędzi i menu kontekstowe są zaimplementowane przez te wtyczki.

Zauważ, że ta klasa jest klasą bazową dla tych wtyczek, jednak aby zintegrować wtyczkę z systemem wtyczek calibre, musisz utworzyć klasę opakowującą, która odwołuje się do rzeczywistej wtyczki. Zobacz moduł `calibre.customize.builtins` po przykłady.

Jeśli dwa obiekty `InterfaceAction` (strona 283) mają taką samą nazwę, pierwszeństwo ma ten o wyższym priorytecie.

Podklasy powinny implementować metody `genesis()` (strona 285), `library_changed()` (strona 286), `location_selected()` (strona 285), `shutting_down()` (strona 286), `initialization_complete()` (strona 286) i `tag_browser_context_action()` (strona 286).

Po zainicjowaniu wtyczka ta ma dostęp do głównego interfejsu użytkownika calibre poprzez członka `gui`. Możesz uzyskać dostęp do innych wtyczek według nazwy, na przykład:

```
self.gui.iactions['Save To Disk']
```

Aby uzyskać dostęp do właściwej wtyczki, użyj atrybutu `interface_action_base_plugin`, ten atrybut staje się dostępny dopiero po zainicjowaniu wtyczki. Przydatne, jeśli chcesz używać metod z klasy wtyczek, takich jak `do_user_config()`.

`QAction` określone przez `action_spec` (strona 284) jest automatycznie tworzone i udostępniane jako `self.qaction`.

name = 'Implement me'

Nazwa wtyczki. Jeśli istnieją dwie wtyczki o tej samej nazwie, pierwszeństwo ma ta o wyższym priorytecie.

priority = 1

Priorytet wtyczki. Jeśli istnieją dwie wtyczki o tej samej nazwie, pierwszeństwo ma ta o wyższym priorytecie.

popup_type = 1

Typ wyskakującego menu, gdy ta wtyczka jest dodawana do paska narzędzi

auto_repeat = False

Czy ta akcja powinna być automatycznie powtarzana po przytrzymaniu klawisza skrótu.

action_spec = ('text', 'icon', None, None)

Of the form: (text, icon_path, tooltip, keyboard shortcut). icon, tooltip and keyboard shortcut can be None. keyboard shortcut must be either a string, None or tuple of shortcuts. If None, a keyboard shortcut corresponding to the action is not registered. If you pass an empty tuple, then the shortcut is registered with no default key binding.

action_shortcut_name = None

If not None, used for the name displayed to the user when customizing the keyboard shortcuts for the above action spec instead of action_spec[0]

action_add_menu = False

Jeśli True, menu jest automatycznie tworzone i dodawane do self.qaction

action_menu_clone_qaction = False

Jeśli True, klon self.qaction jest dodawany do menu self.qaction. Jeśli chcesz, aby tekst tej akcji różnił się od self.qaction, ustaw tę zmienną na nowy tekst

dont_add_to = frozenset({})

Zestaw lokalizacji, do których nie należy dodawać tej akcji. Zobacz `all_locations`, aby zobaczyć listę możliwych lokalizacji

dont_remove_from = frozenset({})

Zestaw lokalizacji, z których nie można usunąć tej akcji. Zobacz `all_locations`, aby zobaczyć listę możliwych lokalizacji

action_type = 'global'

Rodzaj działania „bieżący” oznacza działanie na bieżący widok „globalny” oznacza działanie, które nie działa na bieżący widok, ale raczej na kaliber jako całość

accepts_drops = False

Jeśli True, wtedy to `InterfaceAction` będzie miało możliwość interakcji ze zdarzeniami przeciągania i upuszczania. Zobacz metody, `accept_enter_event()` (strona 284), `meth`accept_drag_move_event``, `drop_event()` (strona 284), aby uzyskać szczegółowe informacje.

accept_enter_event (event, mime_data)

Ta metoda powinna zwrócić True, jeśli ta akcja interfejsu jest w stanie obsłużyć zdarzenie przeciągania. Nie dzwoni do akceptacji/ignorowania wydarzenia, o które zadba interfejs calibre.

accept_drag_move_event (event, mime_data)

Ta metoda powinna zwrócić True, jeśli ta akcja interfejsu jest w stanie obsłużyć zdarzenie przeciągania. Nie dzwoni do akceptacji/ignorowania wydarzenia, o które zadba interfejs calibre.

drop_event (event, mime_data)

Ta metoda powinna wykonać jakąś użyteczną akcję i zwrócić True, jeśli ta akcja interfejsu jest w stanie obsłużyć zdarzenie drop. Nie dzwoni do akceptacji/ignorowania wydarzenia, o które zadba interfejs calibre. W tej funkcji nie należy wykonywać operacji blokujących/długich. Zamiast tego wyemituj sygnał lub użyj `QTimer.singleShot` i wróć szybko. Zobacz wbudowane akcje dla przykładów.

create_menu_action (menu, unique_name, text, icon=None, shortcut=None, description=None, triggered=None, shortcut_name=None, persist_shortcut=False)

Wygodna metoda łatwego dodawania akcji do QMenu. Zwraca utworzone QAction. Ta akcja ma jeden dodatkowy atrybut `calibre_shortcut_unique_name`, który, jeśli nie, brak, odnosi się do unikalnej nazwy, pod którą ta akcja jest zarejestrowana w menedżerze klawiatury.

Parametry

- **menu** – QMenu, do którego zostanie dodana nowo utworzona akcja
- **unique_name** – Unikalna nazwa tej akcji, musi być globalnie unikalna, więc postaraj się, aby była jak najbardziej opisowa. W razie wątpliwości dodaj do niego UUID.
- **text** – Tekst akcji.
- **icon** – Either a QIcon or a file name. The file name is passed to the QIcon.ic() builtin, so you do not need to pass the full path to the images folder.
- **shortcut** – Ciąg, lista ciągów, Brak lub Fałsz. Jeśli False, dla tej akcji nie jest zarejestrowany żaden skrót klawiaturowy. Jeśli Brak, rejestrowany jest skrót klawiaturowy bez domyślnego przypisania klawiszy. Ciąg i lista ciągów rejestrują skrót z domyślnym przypisaniem klawiszy, jak określono.
- **description** – Opis tej akcji. Służy do ustawiania podpowiedzi.
- **triggered** – Wywołany, który jest połączony z wyzwalanym sygnałem tworzonej akcji.
- **shortcut_name** – Tekst wyświetlany użytkownikowi podczas dostosowywania skrótów klawiaturowych dla tej akcji. Domyślnie jest ustawiona na wartość `text`.
- **persist_shortcut** – Skróty do czynności, które nie zawsze się pojawiają lub są zależne od biblioteki, mogą zniknąć, gdy inne skróty klawiaturowe są edytowane, chyba że ``persist_shortcut`` ma wartość True.

`load_resources (names)`

Jeśli ta wtyczka znajduje się w pliku ZIP (wtyczka dodana przez użytkownika), ta metoda pozwoli Ci załadować zasoby z pliku ZIP.

Na przykład, aby załadować obraz:

```

pixmap = QPixmap()
pixmap.loadFromData(tuple(self.load_resources(['images/icon.png'])).
    ↪ values())[0])
icon = QIcon(pixmap)

```

Parametry

names – Lista ścieżek do zasobów w pliku ZIP przy użyciu / jako separatora

Zwraca

Słownik w postaci `{name : file_contents}`. Wszelkie nazwy, które nie zostały znalezione w pliku ZIP, nie będą obecne w słowniku.

`genesis ()`

Skonfiguruj tę wtyczkę. Wywoływane tylko raz podczas inicjalizacji. `self.gui` jest dostępny. Akcja określona przez `action_spec` (strona 284) jest dostępna jako `self.qaction`.

`location_selected (loc)`

Wywoływane, gdy zmienia się lista książek wyświetlana w calibre. Obecnie wartości dla `loc` to: `library`, `main`, `card` i `cardb`.

Ta metoda powinna włączać/wyłączać tę akcję i jej podakcje odpowiednio do lokalizacji.

library_about_to_change (*olddb, db*)

Wywoływane, gdy bieżąca biblioteka zostanie zmieniona.

Parametry

- **olddb** – LibraryDatabase odpowiadająca poprzedniej bibliotece.
- **db** – LibraryDatabase odpowiadająca nowej bibliotece.

library_changed (*db*)

Wywoływane, gdy bieżąca biblioteka zostanie zmieniona.

Parametry

db – LibraryDatabase odpowiadająca bieżącej bibliotece.

gui_layout_complete ()

Wywoływane raz na akcję, gdy układ głównego GUI zostanie ukończony. Jeśli Twoja akcja wymaga wprowadzenia zmian w układzie, należy to zrobić tutaj, a nie w *initialization_complete()* (strona 286).

initialization_complete ()

Wywoływane raz na akcję po zakończeniu inicjalizacji głównego GUI.

tag_browser_context_action (*index*)

Wywoływane podczas wyświetlania menu kontekstowego w przeglądarce znaczników. *index* to QModelIndex, który wskazuje na element przeglądarki Tag, który został kliknięty prawym przyciskiem myszy. Przetestuj go pod kątem ważności za pomocą *index.valid()* i uzyskaj bazowy obiekt TagTreeItem za pomocą *index.data(Qt.ItemDataRole.UserRole)*. Wszelkie obiekty akcji uzyskane tą metodą zostaną dodane do menu kontekstowego.

shutting_down ()

Wywoływane raz na wtyczkę, gdy główny GUI jest w trakcie zamykania. Zwolnij wszystkie używane zasoby, ale staraj się nie blokować wyłączania przez dłuższy czas.

class calibre.customize.**InterfaceActionBase** (*args, **kwargs)

Klasy bazowe: *Plugin* (strona 256)

supported_platforms = ['windows', 'osx', 'linux']

Lista platform, na których pracuje ta wtyczka. Na przykład:

author = 'Kovid Goyal'

Autor wtyczki

type = 'Akcja interfejsu użytkownika'

Typ wtyczki. Używane w celu kategoryzacji wtyczek w GUI.

can_be_disabled = False

Jeśli False, użytkownik nie będzie mógł wyłączyć wtyczki. Używać z rozwagą.

load_actual_plugin (*gui*)

Ta metoda musi zwracać rzeczywisty obiekt wtyczki akcji interfejsu.

12.1.9 Preferencje wtyczki

class calibre.customize.PreferencesPlugin (plugin_path)

Klasy bazowe: *Plugin* (strona 256)

Wtyczka reprezentująca widżet wyświetlany w oknie dialogowym Preferencje.

Ta wtyczka ma tylko jedną ważną metodę *create_widget()* (strona 287). Różne pola wtyczki kontrolują sposób jej kategoryzacji w interfejsie użytkownika.

supported_platforms = ['windows', 'osx', 'linux']

Lista platform, na których pracuje ta wtyczka. Na przykład:

author = 'Kovid Goyal'

Autor wtyczki

type = 'Ustawienia'

Typ wtyczki. Używane w celi kategoryzacji wtyczek w GUI.

can_be_disabled = False

Jeśli False, użytkownik nie będzie mógł wyłączyć wtyczki. Używać z rozwagą.

config_widget = None

Importuj ścieżkę do modułu, który zawiera klasę o nazwie ConfigWidget, która implementuje ConfigWidgetInterface. Używany przez *create_widget()* (strona 287).

category_order = 100

Gdzie na liście kategorii powinien znajdować się *category* (strona 287) tej wtyczki.

name_order = 100

Gdzie na liście nazw w kategorii powinien znajdować się *gui_name* (strona 287) tej wtyczki

category = None

Kategoria, w której powinna należeć ta wtyczka

gui_category = None

Nazwa kategorii wyświetlana użytkownikowi dla tej wtyczki

gui_name = None

Nazwa wyświetlana użytkownikowi dla tej wtyczki

icon = None

Ikona tej wtyczki powinna być ścieżką bezwzględną

description = None

Opis używany w podpowiedziach i tym podobnych

create_widget (parent=None)

Utwórz i zwróć rzeczywisty widżet Qt używany do ustawiania tej grupy preferencji. Widżet musi implementować *calibre.gui2.preferences.ConfigWidgetInterface* (strona 287).

Domyślna implementacja używa *config_widget* (strona 287) do utworzenia instancji widżetu.

class calibre.gui2.preferences.ConfigWidgetInterface

Ta klasa definiuje interfejs, który muszą zaimplementować wszystkie widżety wyświetlane w oknie dialogowym Preferencje. Zobacz *ConfigWidgetBase* (strona 288), aby zapoznać się z klasą bazową, która implementuje ten interfejs i definiuje również różne wygodne metody.

changed_signal = None

Ten sygnał musi być emitowany za każdym razem, gdy użytkownik zmieni wartość w tym widżecie

supports_restoring_to_defaults = True

Ustaw na True, jeśli zaimplementowano metodę `restore_to_defaults()`.

restore_defaults_desc = 'Przywróć domyślne wartości. Musisz kliknąć Zastosuj, aby zapisać domyślne ustawienia.'

Etykieta przycisku „Przywróć domyślne”

restart_critical = False

Jeśli True, okno dialogowe Preferencje nie pozwoli użytkownikowi na ustawienie dalszych preferencji. Działa tylko wtedy, gdy `commit()` (strona 288) zwraca True.

genesis (gui)

Wywoływany raz przed wyświetleniem widżetu, powinien wykonać wszelkie niezbędne ustawienia.

Parametry

gui – Graficzny interfejs użytkownika głównego calibre

initialize()

Powinien ustawić wszystkie wartości konfiguracyjne na ich początkowe wartości (wartości przechowywane w plikach konfiguracyjnych). Oświadczenie „zwrotu” jest opcjonalne. Zwróć False, jeśli okno dialogowe nie ma być pokazywane.

restore_defaults()

Powinna ustawić wszystkie wartości konfiguracyjne na wartości domyślne.

commit()

Zapisz wszystkie zmienione ustawienia. Zwróć True, jeśli zmiany wymagają ponownego uruchomienia, w przeciwnym razie False. Podnieś wyjątek `AbortCommit`, aby wskazać, że wystąpił błąd. Jesteś odpowiedzialny za przekazanie opinii użytkowników o tym, na czym polega błąd i jak go poprawić.

refresh_gui (gui)

Wywoływana raz po zatwierdzeniu tego widżetu. Odpowiedzialny za spowodowanie ponownego odczytania przez interfejs użytkownika wszelkich zmienionych ustawień. Zauważ, że domyślnie GUI i tak ponownie inicjuje różne elementy, więc większość widżetów nie będzie musiała używać tej metody.

initial_tab_changed()

Called if the initially displayed tab is changed before the widget is shown, but after it is initialized.

class calibre.gui2.preferences.ConfigWidgetBase (parent=None)

Klasa bazowa zawierająca kod umożliwiający łatwe dodawanie standardowych widżetów konfiguracyjnych, takich jak pola wyboru, pola kombi, pola tekstowe i tak dalej. Zobacz metodę `register()` (strona 289).

Ta klasa automatycznie obsługuje powiadomienia o zmianach, przywracanie ustawień domyślnych, translację między obiektami gui i obiektami konfiguracyjnymi itp. dla zarejestrowanych ustawień.

Jeśli twój widżet konfiguracji dziedziczy z tej klasy, ale zawiera ustawienia, które nie są zarejestrowane, powinieneś nadpisać metody `ConfigWidgetInterface` (strona 287) i wywołać metody klasy bazowej wewnątrz nadpisań.

changed_signal

Ten sygnał musi być emitowany za każdym razem, gdy użytkownik zmieni wartość w tym widżecie

supports_restoring_to_defaults = True

Ustaw na True, jeśli zaimplementowano metodę `restore_to_defaults()`.

restart_critical = False

Jeśli True, okno dialogowe Preferencje nie pozwoli użytkownikowi na ustawienie dalszych preferencji. Działa tylko wtedy, gdy `commit()` (strona 289) zwraca True.

register (*name, config_obj, gui_name=None, choices=None, restart_required=False, empty_string_is_None=True, setting=<class 'calibre.gui2.preferences.Setting'>*)

Zarejestruj ustawienie.

Parametry

- **name** – Nazwa ustawienia
- **config_obj** – Obiekt konfiguracyjny, który odczytuje/zapisuje ustawienie
- **gui_name** – Nazwa obiektu GUI, który przedstawia interfejs do zmiany ustawienia. Domyślnie przyjmuje się, że jest to 'opt_' + name.
- **choices** – Jeśli to ustawienie jest ustawieniem opartym na wielokrotnym wyborze (pole rozwijane), lista opcji do wyboru. Lista jest listą dwóch elementów krotek postaci: [(gui name, value), ...]
- **setting** – Klasa odpowiedzialna za zarządzanie tym ustawieniem. Klasa domyślna obsługuje prawie wszystkie przypadki, więc ten parametr jest rzadko używany.

initialize()

Powinien ustawić wszystkie wartości konfiguracyjne na ich początkowe wartości (wartości przechowywane w plikach konfiguracyjnych). Oświadczenie „zwrotu” jest opcjonalne. Zwróć False, jeśli okno dialogowe nie ma być pokazywane.

commit (*args)

Zapisz wszystkie zmienione ustawienia. Zwróć True, jeśli zmiany wymagają ponownego uruchomienia, w przeciwnym razie False. Podnieś wyjątek `AbortCommit`, aby wskazać, że wystąpił błąd. Jesteś odpowiedzialny za przekazanie opinii użytkowników o tym, na czym polega błąd i jak go poprawić.

restore_defaults (*args)

Powinna ustawić wszystkie wartości konfiguracyjne na wartości domyślne.

12.2 Zmienne środowiskowe

- `CALIBRE_CONFIG_DIRECTORY` — ustawia folder, w którym przechowywane/odczytywane są pliki konfiguracyjne.
- `CALIBRE_TEMP_DIR` - ustawia folder tymczasowy używany przez calibre
- `CALIBRE_CACHE_DIRECTORY` - ustawia calibre folderu używany do buforowania trwałych danych między sesjami
- `CALIBRE_OVERRIDE_DATABASE_PATH` - pozwala na określenie pełnej ścieżki do metadata.db. Dzięki temu można mieć metadata.db w innym miejscu niż folder biblioteki. Jest to przydatne jeśli folder biblioteki znajduje się na dysku sieciowym, który nie obsługuje blokowania plików.
- `CALIBRE_DEVELOP_FROM` — używany do uruchamiania ze środowiska programistycznego calibre. Zobacz *Konfigurowanie środowiska programistycznego dla calibre* (strona 345).
- `CALIBRE_OVERRIDE_LANG` - służy do wymuszenia języka używanego przez interfejs (kod języka ISO 639)
- `CALIBRE_TEST_TRANSLATION` - służy do testowania pliku .po tłumaczenia (powinna być ścieżką do pliku .po)

- CALIBRE_NO_NATIVE_FILEDIALOGS — powoduje, że Calibre nie używa natywnych okien dialogowych plików do wybierania plików/folderów.
- CALIBRE_NO_NATIVE_MENUBAR - causes calibre to not create a native (global) menu on Ubuntu Unity and similar Linux desktop environments. The menu is instead placed inside the window, as is traditional.
- CALIBRE_USE_SYSTEM_THEME - domyślnie, w systemie Linux, calibre używa własnego wbudowanego stylu Qt. Ma to na celu uniknięcie awarii i zawieszeń spowodowanych niezgodnością między wersją calibre Qt, z którą jest zbudowany, a systemem Qt. Minusem jest to, że calibre może nie pasować do wyglądu i stylu systemu. Jeśli ustawisz tę zmienną środowiskową w systemie Linux, spowoduje to, że Calibre użyje motywu systemowego – uważaj na awarie i zawieszanie się.
- CALIBRE_SHOW_DEPRECATION_WARNINGS — powoduje, że Calibre drukuje ostrzeżenia o wycofaniu na standardowe wyjście. Przydatne dla programistów calibre.
- CALIBRE_NO_DEFAULT_PROGRAMS - zapobiega automatycznemu rejestrowaniu przez Calibre typów plików, które może obsługiwać w systemie Windows.
- QT_QPA_PLATFORM - On Linux set this to wayland to force calibre to use Wayland and xcb to force use of X11.
- SYSFS_PATH - ścieżka do sysfs jeśli jest on zamontowany gdzieś indziej niż /sys
- http_proxy, https_proxy — używane w systemie Linux do określenia proxy HTTP(S)

Zobacz [Jak ustawić zmienne środowiskowe w systemie Windows](#)¹⁰⁸. Jeśli korzystasz z systemu macOS, możesz ustawić zmienne środowiskowe, tworząc plik ~/Library/Preferences/calibre/macos-env.txt i umieszczając w nim zmienne środowiskowe po jednej w wierszu, na przykład:

```
CALIBRE_DEVELOP_FROM=$HOME/calibre-src/src
CALIBRE_NO_NATIVE_FILEDIALOGS=1
CALIBRE_CONFIG_DIRECTORY=~/.config/calibre
```

12.3 Parametry

Parametry to zmienne, które mają wpływ na różne aspekty działania calibre. Są one dostępne w Ustawienia->Zaawansowane->Parametry. Domyślne wartości parametrów zostały przedstawione poniżej.

```
#!/usr/bin/env python
# vim:fileencoding=UTF-8:ts=4:sw=4:sta:et:sts=4:ai
# License: GPLv3 Copyright: 2010, Kovid Goyal <kovid at kovidgoyal.net>

# Contains various tweaks that affect calibre behavior. Only edit this file if
# you know what you are doing. If you delete this file, it will be recreated from
# defaults.

#: Auto increment series index
# The algorithm used to assign a book added to an existing series a series number.
# New series numbers assigned using this tweak are always integer values, except
# if a constant non-integer is specified.
# Possible values are:
#   next - First available integer larger than the largest existing number
#   first_free - First available integer larger than 0
#   next_free - First available integer larger than the smallest existing number
```

(ciąg dalszy na następnej stronie)

¹⁰⁸ <https://www.computerhope.com/issues/ch000549.htm>

(kontynuacja poprzedniej strony)

```

# last_free - First available integer smaller than the largest existing number.↵
↵Return largest existing + 1 if no free number is found
# const - Assign the number 1 always
# no_change - Do not change the series index
# a number - Assign that number always. The number is not in quotes. Note that 0.0↵
↵can be used here.
# Examples:
# series_index_auto_increment = 'next'
# series_index_auto_increment = 'next_free'
# series_index_auto_increment = 16.5
#
# Set the use_series_auto_increment_tweak_when_importing tweak to True to
# use the above values when importing/adding books. If this tweak is set to
# False (the default) then the series number will be set to 1 if it is not
# explicitly set during the import. If set to True, then the
# series index will be set according to the series_index_auto_increment setting.
# Note that the use_series_auto_increment_tweak_when_importing tweak is used
# only when a value is not provided during import. If the importing regular
# expression produces a value for series_index, or if you are reading metadata
# from books and the import plugin produces a value, then that value will
# be used irrespective of the setting of the tweak.
series_index_auto_increment = 'next'
use_series_auto_increment_tweak_when_importing = False

#: Add separator after completing an author name
# Set this if the completion separator should be appended to the end of the
# completed text to automatically begin a new completion operation for authors.
# It can be either True or False
authors_completer_append_separator = False

#: Author sort name algorithm
# The algorithm used to copy author to author_sort.
# Possible values are:
# invert: use "fn ln" -> "ln, fn"
# copy : copy author to author_sort without modification
# comma : use 'copy' if there is a ',' in the name, otherwise use 'invert'
# nocomma : "fn ln" -> "ln fn" (without the comma)
# When this tweak is changed, the author_sort values stored with each author
# must be recomputed by right-clicking on an author in the left-hand tags
# panel, selecting 'Manage authors', and pressing
# 'Recalculate all author sort values'.
#
# The author_name_suffixes are words that are ignored when they occur at the
# end of an author name. The case of the suffix is ignored and trailing
# periods are automatically handled.
#
# The same is true for author_name_prefixes.
#
# The author_name_copywords are a set of words which, if they occur in an
# author name, cause the automatically generated author sort string to be
# identical to the author's name. This means that the sort for a string like
# "Acme Inc." will be "Acme Inc." instead of "Inc., Acme".
#
# If author_use_surname_prefixes is enabled, any of the words in
# author_surname_prefixes will be treated as a prefix to the surname, if they
# occur before the surname. So for example, "John von Neumann" would be sorted
# as "von Neumann, John" and not "Neumann, John von".

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

author_sort_copy_method = 'comma'
author_name_suffixes = ('Jr', 'Sr', 'Inc', 'Ph.D', 'Phd',
                        'MD', 'M.D', 'I', 'II', 'III', 'IV',
                        'Junior', 'Senior')
author_name_prefixes = ('Mr', 'Mrs', 'Ms', 'Dr', 'Prof')
author_name_copywords = (
    'Agency', 'Corporation', 'Company', 'Co.', 'Council',
    'Committee', 'Inc.', 'Institute', 'National', 'Society', 'Club', 'Team',
    'Software', 'Games', 'Entertainment', 'Media', 'Studios',
)
author_use_surname_prefixes = False
author_surname_prefixes = ('da', 'de', 'di', 'la', 'le', 'van', 'von')

#: Splitting multiple author names
# By default, calibre splits a string containing multiple author names on
# ampersands and the words "and" and "with". You can customize the splitting
# by changing the regular expression below. Strings are split on whatever the
# specified regular expression matches, in addition to ampersands.
# Default: r'(?i),?\s+(and|with)\s+'
authors_split_regex = r'(?i),?\s+(and|with)\s+'

#: Use author sort in Tag browser
# Set which author field to display in the Tag browser (the list of authors,
# series, publishers etc on the left hand side). The choices are author and
# author_sort. This tweak affects only what is displayed under the authors
# category in the Tag browser and Content server. Please note that if you set this
# to author_sort, it is very possible to see duplicate names in the list because
# although it is guaranteed that author names are unique, there is no such
# guarantee for author_sort values. Showing duplicates won't break anything, but
# it could lead to some confusion. When using 'author_sort', the tooltip will
# show the author's name.
# Examples:
#   categories_use_field_for_author_name = 'author'
#   categories_use_field_for_author_name = 'author_sort'
categories_use_field_for_author_name = 'author'

#: Control partitioning of Tag browser
# When partitioning the Tag browser, the format of the subcategory label is
# controlled by a template: categories_collapsed_name_template if sorting by
# name, categories_collapsed_rating_template if sorting by average rating, and
# categories_collapsed_popularity_template if sorting by popularity. There are
# two variables available to the template: first and last. The variable 'first'
# is the initial item in the subcategory, and the variable 'last' is the final
# item in the subcategory. Both variables are 'objects'; they each have multiple
# values that are obtained by using a suffix. For example, first.name for an
# author category will be the name of the author. The sub-values available are:
#   name: the printable name of the item
#   count: the number of books that references this item
#   avg_rating: the average rating of all the books referencing this item
#   sort: the sort value. For authors, this is the author_sort for that author
#   category: the category (e.g., authors, series) that the item is in.
# Note that the "r" in front of the { is necessary if there are backslashes
# (\ characters) in the template. It doesn't hurt anything to leave it there
# even if there aren't any backslashes.
categories_collapsed_name_template = r'{first.sort:shorten(4,,0)} - {last.
↪sort:shorten(4,,0)}'
categories_collapsed_rating_template = r'{first.avg_rating:4.2f:ifempty(0)} - {last.

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

↪avg_rating:4.2f:ifempty(0)}'
categories_collapsed_popularity_template = r'{first.count:d} - {last.count:d}'

#: Specify columns to sort the booklist by on startup
# Provide a set of columns to be sorted on when calibre starts.
# The argument is None if saved sort history is to be used
# otherwise it is a list of column,order pairs. Column is the
# lookup/search name, found using the tooltip for the column
# Order is 0 for ascending, 1 for descending.
# For example, set it to [('authors',0),('title',0)] to sort by
# title within authors.
sort_columns_at_startup = None

#: Control how dates are displayed
# Format to be used for publication date and the timestamp (date).
# A string controlling how the publication date is displayed in the GUI
# d      the day as number without a leading zero (1 to 31)
# dd     the day as number with a leading zero (01 to 31)
# ddd    the abbreviated localized day name (e.g. 'Mon' to 'Sun').
# dddd   the long localized day name (e.g. 'Monday' to 'Sunday').
# M      the month as number without a leading zero (1-12)
# MM     the month as number with a leading zero (01-12)
# MMM    the abbreviated localized month name (e.g. 'Jan' to 'Dec').
# MMMM   the long localized month name (e.g. 'January' to 'December').
# yy     the year as two digit number (00-99)
# yyyy   the year as four digit number
# h      the hours without a leading 0 (0 to 11 or 0 to 23, depending on am/pm) '
# hh     the hours with a leading 0 (00 to 11 or 00 to 23, depending on am/pm) '
# m      the minutes without a leading 0 (0 to 59) '
# mm     the minutes with a leading 0 (00 to 59) '
# s      the seconds without a leading 0 (0 to 59) '
# ss     the seconds with a leading 0 (00 to 59) '
# ap     use a 12-hour clock instead of a 24-hour clock, with "ap" replaced by the_
↪localized string for am or pm
# AP     use a 12-hour clock instead of a 24-hour clock, with "AP" replaced by the_
↪localized string for AM or PM
# iso    the date with time and timezone. Must be the only format present
# For example, given the date of 9 Jan 2010, the following formats show
# MMM yyyy ==> Jan 2010      yyyy ==> 2010      dd MMM yyyy ==> 09 Jan 2010
# MM/yyyy ==> 01/2010      d/M/yy ==> 9/1/10    yy ==> 10
#
# publication default if not set: MMM yyyy
# timestamp default if not set: dd MMM yyyy
# last_modified_display_format if not set: dd MMM yyyy
gui_pubdate_display_format = 'MMM yyyy'
gui_timestamp_display_format = 'dd MMM yyyy'
gui_last_modified_display_format = 'dd MMM yyyy'

#: Control sorting of titles and series in the library display
# Control title and series sorting in the library view. If set to
# 'library_order', the title sort field will be used instead of the title.
# Unless you have manually edited the title sort field, leading articles such as
# The and A will be ignored. If set to 'strictly_alphabetic', the titles will be
# sorted as-is (sort by title instead of title sort). For example, with
# library_order, The Client will sort under 'C'. With strictly_alphabetic, the
# book will sort under 'T'.
# This flag affects calibre's library display. It has no effect on devices. In

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

# addition, titles for books added before changing the flag will retain their
# order until the title is edited. Editing a title and hitting Enter
# without changing anything is sufficient to change the sort. Or you can use
# the 'Update title sort' action in the Bulk metadata edit dialog to update
# it for many books at once.
title_series_sorting = 'library_order'

#: Control formatting of title and series when used in templates
# Control how title and series names are formatted when saving to disk/sending
# to device. The behavior depends on the field being processed. If processing
# title, then if this tweak is set to 'library_order', the title will be
# replaced with title_sort. If it is set to 'strictly_alphabetic', then the
# title will not be changed. If processing series, then if set to
# 'library_order', articles such as 'The' and 'An' will be moved to the end. If
# set to 'strictly_alphabetic', the series will be sent without change.
# For example, if the tweak is set to library_order, "The Lord of the Rings"
# will become "Lord of the Rings, The". If the tweak is set to
# strictly_alphabetic, it would remain "The Lord of the Rings". Note that the
# formatter function raw_field will return the base value for title and
# series regardless of the setting of this tweak.
save_template_title_series_sorting = 'library_order'

#: Set the list of words considered to be "articles" for sort strings
# Set the list of words that are to be considered 'articles' when computing the
# title sort strings. The articles differ by language. By default, calibre uses
# a combination of articles from English and whatever language the calibre user
# interface is set to. In addition, in some contexts where the book language is
# available, the language of the book is used. You can change the list of
# articles for a given language or add a new language by editing
# per_language_title_sort_articles. To tell calibre to use a language other
# than the user interface language, set, default_language_for_title_sort. For
# example, to use German, set it to 'deu'. A value of None means the user
# interface language is used. The setting title_sort_articles is ignored
# (present only for legacy reasons).
per_language_title_sort_articles = {
    # English
    'eng' : (r'A\s+', r'The\s+', r'An\s+'),
    # Esperanto
    'epo': (r'La\s+', r"L'", 'L'),
    # Spanish
    'spa' : (r'El\s+', r'La\s+', r'Lo\s+', r'Los\s+', r'Las\s+', r'Un\s+',
             r'Una\s+', r'Unos\s+', r'Unas\s+'),
    # French
    'fra' : (r'Le\s+', r'La\s+', r"L'", u'L', u'L', r'Les\s+', r'Un\s+', r'Une\
→s+',
             r'Des\s+', r'De\s+La\s+', r'De\s+', r"D'", r'D', r'D'),
    # Polish
    'pol': (),
    # Italian
    'ita': ('Lo\\s+', 'Il\\s+', "L'", 'L', 'La\\s+', 'Gli\\s+',
            'I\\s+', 'Le\\s+', 'Uno\\s+', 'Un\\s+', 'Una\\s+', "Un'",
            'Un', 'Dei\\s+', 'Degli\\s+', 'Delle\\s+', 'Del\\s+',
            'Della\\s+', 'Dello\\s+', "Dell'", 'Dell'),
    # Portuguese
    'por' : (r'A\s+', r'O\s+', r'Os\s+', r'As\s+', r'Um\s+', r'Uns\s+',
             r'Uma\s+', r'Umas\s+', ),
    # Romanian

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

'ron' : (r'Un\s+', r'O\s+', r'Nişte\s+', ),
# German
'deu' : (r'Der\s+', r'Die\s+', r'Das\s+', r'Den\s+', r'Ein\s+',
        r'Eine\s+', r'Einen\s+', r'Dem\s+', r'Des\s+', r'Einem\s+',
        r'Eines\s+'),
# Dutch
'nld' : (r'De\s+', r'Het\s+', r'Een\s+', r'"n\s+", r'"s\s+", r'Ene\s+',
        r'Ener\s+', r'Enes\s+', r'Den\s+', r'Der\s+', r'Des\s+',
        r'"t\s+"),
# Swedish
'swe' : (r'En\s+', r'Ett\s+', r'Det\s+', r'Den\s+', r'De\s+', ),
# Turkish
'tur' : (r'Bir\s+', ),
# Afrikaans
'afr' : (r'"n\s+", r'Die\s+', ),
# Greek
'ell' : (r'O\s+', r'I\s+', r'To\s+', r'Ta\s+', r'Tus\s+', r'Tis\s+',
        r'"Enas\s+", r'"Mia\s+", r'"Ena\s+", r'"Enan\s+", ),
# Hungarian
'hun' : (r'A\s+', r'Az\s+', r'Egy\s+',),
}
default_language_for_title_sort = None
title_sort_articles=r'^(A|The|An)\s+'

#: Specify a folder calibre should connect to at startup
# Specify a folder that calibre should connect to at startup using
# connect_to_folder. This must be a full path to the folder. If the folder does
# not exist when calibre starts, it is ignored.
# Example for Windows:
#     auto_connect_to_folder = 'C:/Users/someone/Desktop/testlib'
# Example for other operating systems:
#     auto_connect_to_folder = '/home/dropbox/My Dropbox/someone/library'
auto_connect_to_folder = ''

#: Specify renaming rules for SONY collections
# Specify renaming rules for SONY collections. This tweak is only applicable if
# metadata management is set to automatic. Collections on SONYs are named
# depending upon whether the field is standard or custom. A collection derived
# from a standard field is named for the value in that field.
#
# For example, if the standard 'series' column contains the value 'Darkover', then the
# collection name is 'Darkover'. A collection derived from a custom field will
# have the name of the field added to the value. For example, if a custom series
# column named 'My Series' contains the name 'Darkover', then the collection
# will by default be named 'Darkover (My Series)'. For purposes of this
# documentation, 'Darkover' is called the value and 'My Series' is called the
# category. If two books have fields that generate the same collection name,
# then both books will be in that collection.
#
# This set of tweaks lets you specify for a standard or custom field how
# the collections are to be named. You can use it to add a description to a
# standard field, for example 'Foo (Tag)' instead of the 'Foo'. You can also use
# it to force multiple fields to end up in the same collection.
#
# For example, you could force the values in 'series', '#my_series_1', and
# '#my_series_2' to appear in collections named 'some_value (Series)', thereby
# merging all of the fields into one set of collections.

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

#
# There are two related tweaks. The first determines the category name to use
# for a metadata field. The second is a template, used to determine how the
# value and category are combined to create the collection name.
# The syntax of the first tweak, sony_collection_renaming_rules, is:
# {'field_lookup_name':'category_name_to_use', 'lookup_name':'name', ...}
#
# The second tweak, sony_collection_name_template, is a template. It uses the
# same template language as plugboards and save templates. This tweak controls
# how the value and category are combined together to make the collection name.
# The only two fields available are {category} and {value}. The {value} field is
# never empty. The {category} field can be empty. The default is to put the
# value first, then the category enclosed in parentheses, it isn't empty:
# '{value} {category:| (|)}'
#
# Examples: The first three examples assume that the second tweak
# has not been changed.
#
# 1) I want three series columns to be merged into one set of collections. The
# column lookup names are 'series', '#series_1' and '#series_2'. I want nothing
# in the parenthesis. The value to use in the tweak value would be:
#   sony_collection_renaming_rules={'series':'', '#series_1':'', '#series_2':''}
#
# 2) I want the word '(Series)' to appear on collections made from series, and
# the word '(Tag)' to appear on collections made from tags. Use:
#   sony_collection_renaming_rules={'series':'Series', 'tags':'Tag'}
#
# 3) I want 'series' and '#myseries' to be merged, and for the collection name
# to have '(Series)' appended. The renaming rule is:
#   sony_collection_renaming_rules={'series':'Series', '#myseries':'Series'}
#
# 4) Same as example 2, but instead of having the category name in parentheses
# and appended to the value, I want it prepended and separated by a colon, such
# as in Series: Darkover. I must change the template used to format the category name
#
# The resulting two tweaks are:
#   sony_collection_renaming_rules={'series':'Series', 'tags':'Tag'}
#   sony_collection_name_template='{category:||: }{value}'
sony_collection_renaming_rules={}
sony_collection_name_template='{value}{category:| (|)}'

#: Specify how SONY collections are sorted
# Specify how SONY collections are sorted. This tweak is only applicable if
# metadata management is set to automatic. You can indicate which metadata is to
# be used to sort on a collection-by-collection basis. The format of the tweak
# is a list of metadata fields from which collections are made, followed by the
# name of the metadata field containing the sort value.
# Example: The following indicates that collections built from pubdate and tags
# are to be sorted by the value in the custom column '#mydate', that collections
# built from 'series' are to be sorted by 'series_index', and that all other
# collections are to be sorted by title. If a collection metadata field is not
# named, then if it is a series-based collection it is sorted by series order,
# otherwise it is sorted by title order.
# ([['pubdate', 'tags'], '#mydate'), (['series'], 'series_index'), (['*', 'title'])]
# Note that the bracketing and parentheses are required. The syntax is
# [ ( [list of fields], sort field ) , ( [list of fields] , sort field ) ]
# Default: empty (no rules), so no collection attributes are named.

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

sony_collection_sorting_rules = []

#: Control how tags are applied when copying books to another library
# Set this to True to ensure that tags in 'Tags to add when adding
# a book' are added when copying books to another library
add_new_book_tags_when_importing_books = False

#: Set the maximum number of sort 'levels'
# Set the maximum number of sort 'levels' that calibre will use to resort the
# library after certain operations such as searches or device insertion. Each
# sort level adds a performance penalty. If the database is large (thousands of
# books) the penalty might be noticeable. If you are not concerned about multi-
# level sorts, and if you are seeing a slowdown, reduce the value of this tweak.
maximum_resort_levels = 5

#: Choose whether dates are sorted using visible fields
# Date values contain both a date and a time. When sorted, all the fields are
# used, regardless of what is displayed. Set this tweak to True to use only
# the fields that are being displayed.
sort_dates_using_visible_fields = False

#: Fuzz value for trimming covers
# The value used for the fuzz distance when trimming a cover.
# Colors within this distance are considered equal.
# The distance is in absolute intensity units.
cover_trim_fuzz_value = 10

#: Control behavior of the book list
# You can control the behavior of double clicks and pressing Enter on the books
# list. Choices: open_viewer, do_nothing, show_book_details,
# show_locked_book_details, edit_cell, edit_metadata. Selecting anything other
# than open_viewer, show_book_details, or show_locked_book_details has the side
# effect of disabling editing a field using a single click.
# Default: open_viewer.
# Example: doubleclick_on_library_view = 'do_nothing'
# You can also control whether the book list scrolls per item or
# per pixel. Default is per item.
doubleclick_on_library_view = 'open_viewer'
enter_key_behavior = 'do_nothing'
horizontal_scrolling_per_column = False
vertical_scrolling_per_row = False

#: Language to use when sorting
# Setting this tweak will force sorting to use the
# collating order for the specified language. This might be useful if you run
# calibre in English but want sorting to work in the language where you live.
# Set the tweak to the desired ISO 639-1 language code, in lower case.
# You can find the list of supported locales at
# https://en.wikipedia.org/wiki/List\_of\_ISO\_639-1\_codes
# Default: locale_for_sorting = '' -- use the language calibre displays in
# Example: locale_for_sorting = 'fr' -- sort using French rules.
# Example: locale_for_sorting = 'nb' -- sort using Norwegian rules.
locale_for_sorting = ''

#: The number of seconds to wait before sending emails
# The number of seconds to wait before sending emails when using a
# public email server like GMX/Hotmail/Gmail. Default is: 5 minutes

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
# Setting it to lower may cause the server's SPAM controls to kick in,
# making email sending fail. Changes will take effect only after a restart of
# calibre. You can also change the list of hosts that calibre considers
# to be public relays here. Any relay host ending with one of the suffixes
# in the list below will be considered a public email server.
public_smtp_relay_delay = 301
public_smtp_relay_host_suffixes = ['gmail.com', 'live.com', 'gmx.com', 'outlook.com']

#: The maximum width and height for covers saved in the calibre library
# All covers in the calibre library will be resized, preserving aspect ratio,
# to fit within this size. This is to prevent slowdowns caused by extremely
# large covers
maximum_cover_size = (1650, 2200)

#: Where to send downloaded news
# When automatically sending downloaded news to a connected device, calibre
# will by default send it to the main memory. By changing this tweak, you can
# control where it is sent. Valid values are "main", "carda", "cardb". Note
# that if there isn't enough free space available on the location you choose,
# the files will be sent to the location with the most free space.
send_news_to_device_location = "main"

#: Unified toolbar on macOS
# If you enable this option and restart calibre, the toolbar will be 'unified'
# with the titlebar as is normal for macOS applications. However, doing this has
# various bugs, for instance the minimum width of the toolbar becomes twice
# what it should be and it causes other random bugs on some systems, so turn it
# on at your own risk!
unified_title_toolbar_on_osx = False

#: Save original file when converting/polishing from same format to same format
# When calibre does a conversion from the same format to the same format, for
# example, from EPUB to EPUB, the original file is saved, so that in case the
# conversion is poor, you can tweak the settings and run it again. By setting
# this to False you can prevent calibre from saving the original file.
# Similarly, by setting save_original_format_when_polishing to False you can
# prevent calibre from saving the original file when polishing.
save_original_format = True
save_original_format_when_polishing = True

#: Number of recently viewed books to show
# Right-clicking the "View" button shows a list of recently viewed books. Control
# how many should be shown, here.
gui_view_history_size = 15

#: Change the font size of the Book details panel in the interface
# Change the font size at which book details are rendered in the side panel and
# comments are rendered in the metadata edit dialog. Set it to a positive or
# negative number to increase or decrease the font size.
change_book_details_font_size_by = 0

#: What format to default to when using the "Unpack book" feature
# The "Unpack book" feature of calibre allows direct editing of a book format.
# If multiple formats are available, calibre will offer you a choice
# of formats, defaulting to your preferred output format if it is available.
# Set this tweak to a specific value of 'EPUB' or 'AZW3' to always default
# to that format rather than your output format preference.
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

# Set to a value of 'remember' to use whichever format you chose last time you
# used the "Unpack book" feature.
# Examples:
#   default_tweak_format = None           (Use output format)
#   default_tweak_format = 'EPUB'
#   default_tweak_format = 'remember'
default_tweak_format = None

#: Do not preselect a completion when editing authors/tags/series/etc.
# This means that you can make changes and press Enter and your changes will
# not be overwritten by a matching completion. However, if you wish to use the
# completions you will now have to press Tab to select one before pressing
# Enter. Which technique you prefer will depend on the state of metadata in
# your library and your personal editing style.
#
# If preselect_first_completion is False and you want Tab to accept what you
# typed instead of the first completion then set tab_accepts_uncompleted_text
# to True. If you do this then to select from the completions you must press
# the Down or Up arrow keys. The tweak tab_accepts_uncompleted_text is ignored
# if preselect_first_completion is True
preselect_first_completion = False
tab_accepts_uncompleted_text = False

#: Completion mode when editing authors/tags/series/etc.
# By default, when completing items, calibre will show you all the candidates
# that start with the text you have already typed. You can instead have it show
# all candidates that contain the text you have already typed. To do this, set
# completion_mode to 'contains'. For example, if you type asi it will match both
# Asimov and Quasimodo, whereas the default behavior would match only Asimov.
completion_mode = 'prefix'

#: Sort the list of libraries alphabetically
# The list of libraries in the Copy to library and Quick switch menus are
# normally sorted by most used. However, if there are more than a certain
# number of such libraries, the sorting becomes alphabetic. You can set that
# number here. The default is ten libraries.
many_libraries = 10

#: Choose available output formats for conversion
# Restrict the list of available output formats in the conversion dialogs.
# For example, if you only want to convert to EPUB and AZW3, change this to
# restrict_output_formats = ['EPUB', 'AZW3']. The default value of None causes
# all available output formats to be present.
restrict_output_formats = None

#: Set the thumbnail image quality used by the Content server
# The quality of a thumbnail is largely controlled by the compression quality
# used when creating it. Set this to a larger number to improve the quality.
# Note that the thumbnails get much larger with larger compression quality
# numbers.
# The value can be between 50 and 99
content_server_thumbnail_compression_quality = 75

#: Image file types to treat as e-books when dropping onto the "Book details" panel
# Normally, if you drop any image file in a format known to calibre onto the
# "Book details" panel, it will be used to set the cover. If you want to store
# some image types as e-books instead, you can set this tweak.

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

# Examples:
#   cover_drop_exclude = {'tiff', 'webp'}
cover_drop_exclude = ()

#: Exclude fields when copy/pasting metadata
# You can ask calibre to not paste some metadata fields when using the
# Edit metadata->Copy metadata/Paste metadata actions. For example,
# exclude_fields_on_paste = ['cover', 'timestamp', '#mycolumn']
# to prevent pasting of the cover, Date and custom column, mycolumn.
# You can also add a shortcut in Preferences->Shortcuts->Edit metadata
# to paste metadata ignoring this tweak.
exclude_fields_on_paste = []

#: Skip internet connected check
# Skip checking whether the internet is available before downloading news.
# Useful if for some reason your operating systems network checking
# facilities are not reliable (for example NetworkManager on Linux).
skip_network_check = False

#: Tab stop width in the template editor
# Sets the width of the tab stop in the template editor in "average characters".
# For example, a value of 1 results in a space with the width of one average_
↪character.
template_editor_tab_stop_width = 4

#: Value for undefined numbers when sorting
# Sets the value to use for undefined numbers when sorting.
# For example, the value -10 sorts undefined numbers as if they were set to -10.
# Use 'maximum' for the largest possible number. Use 'minimum' for the smallest
# possible number. Quotes are optional if entering a number.
# Examples:
#   value_for_undefined_numbers_when_sorting = -100
#   value_for_undefined_numbers_when_sorting = '2'
#   value_for_undefined_numbers_when_sorting = -0.01
#   value_for_undefined_numbers_when_sorting = 'minimum'
#   value_for_undefined_numbers_when_sorting = 'maximum'
value_for_undefined_numbers_when_sorting = 0

#: Allow template database functions in composite columns
# If True then the template database functions book_values() and book_count()
# can be used in composite custom columns. Note: setting this tweak to True and
# using these functions in composites can be very slow.
# Default: False
allow_template_database_functions_in_composites = False

#: Change the programs that are run when opening files/URLs
# By default, calibre passes URLs to the operating system to open using
# whatever default programs are configured there. Here you can override
# that by specifying the program to use, per URL type. For local files,
# the type is "file" and for web links it is "http*". For example:
# openers_by_scheme = { "http*": "firefox %u" } will make calibre run Firefox
# for https://whatever URLs. %u is replaced by the URL to be opened. The scheme
# takes a glob pattern allowing a single entry to match multiple URL types.
openers_by_scheme = {}

#: Set the first day of the week for calendar popups

```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```
# It must be one of the values Default, Sunday, Monday, Tuesday, Wednesday,
# Thursday, Friday, or Saturday, all in English, spelled exactly as shown.
calendar_start_day_of_week = 'Default'
```

12.4 Własne ikony, szablony itp.

Informacja: calibre has direct support for icon themes, there are several icon themes available for calibre, that you can use by going to *Preferences → Interface → Look & Feel → Change icon theme*. It is preferable to use icon themes over overriding individual icons.

calibre allows you to override the static resources, like icons, JavaScript and templates for the metadata jacket, catalogs, etc. with customized versions that you like. All static resources are stored in the resources sub-folder of the calibre install location. On Windows, this is usually `C:\Program Files\Calibre2\app\resources`. On macOS, `/Applications/calibre.app/Contents/Resources/resources/`. On Linux, if you are using the binary installer from the calibre website it will be `/opt/calibre/resources`. These paths can change depending on where you choose to install calibre.

Nie powinieneś zmieniać plików w tym folderze zasobów, ponieważ twoje zmiany zostaną nadpisane przy następnej aktualizacji calibre. Zamiast tego przejdź do *Preferencje → Zaawansowane → Różne* i kliknij *Otwórz folder konfiguracji calibre*. W tym folderze konfiguracyjnym utwórz podfolder o nazwie zasoby i umieść w nim pliki, które chcesz zastąpić. Umieść pliki w odpowiednich podfolderach, na przykład umieść obrazy w `resources/images` itp. calibre automatycznie użyje pliku niestandardowego zamiast wbudowanego przy następnym uruchomieniu.

For example, if you wanted to change the icon for the *Remove books* action, you would first look in the built-in resources folder and see that the relevant file is `resources/images/remove_books.png`. Assuming you have an alternate icon in PNG format called `my_remove_books.png` you would save it in the configuration folder as `resources/images/remove_books.png`. All the icons used by the calibre user interface are in `resources/images` and its sub-folders. Placing an override file here will have even higher priority than a custom icon theme.

12.5 Tworzenie własnego motywu ikon dla calibre

If you have created a beautiful set of icons and wish to share them with other calibre users via calibre's builtin icon theme support, you can easily package up your icons into a theme. To do so, go to *Preferences → Miscellaneous → Create icon theme*, select the folder where you have put your icons. Then fill up the theme metadata and click OK. This will result in a ZIP file containing the theme icons. You can upload that to the calibre forum at [Mobileread](https://www.mobileread.com/forums/forumdisplay.php?f=166)¹⁰⁹ and then I will make your theme available via calibre's builtin icon theme system. By default, the icon theme you just created will also be installed as the current theme in calibre. If you are testing your theme, remember to remove the images from the `resources/images` folder so that the icons from the theme are used.

As of calibre 6, you can have custom icons for light and dark mode. Simply create two versions of the icon and name the files with the suffix `-for-dark-theme` and `-for-light-theme`. For example, `modified-for-dark-theme.png` and `modified-for-light-theme.png`. Then calibre will automatically use the appropriate icon based on the current theme.

¹⁰⁹ <https://www.mobileread.com/forums/forumdisplay.php?f=166>

12.6 Dostosowywanie calibre za pomocą wtyczek

calibre has a very modular design. Almost all functionality in calibre comes in the form of plugins. Plugins are used for conversion, for downloading news (though these are called recipes), for various components of the user interface, to connect to different devices, to process files when adding them to calibre and so on. You can get a complete list of all the built-in plugins in calibre by going to *Preferences* → *Advanced* → *Plugins*.

Można stworzyć własne wtyczki, by dostosować i rozszerzyć działanie calibre. Architektura wtyczek jest bardzo prosta, patrz podręcznik: *Pisanie własnych wtyczek, by rozszerzyć funkcjonalność calibre* (strona 224).

Once you have written a plugin, you can upload that to the calibre plugins forum at [Mobileread](https://www.mobileread.com/forums/)¹¹⁰ and it will be made available via calibre's builtin plugin updater.

¹¹⁰ <https://www.mobileread.com/forums/forumdisplay.php?f=237>

```
kovid giskard ~/work/libprs500/src/libprs500/manual $
```

Informacja: On macOS, the command line tools are inside the calibre bundle, for example, if you installed calibre in /Applications the command line tools are in /Applications/calibre.app/Contents/MacOS/. So, for example, to run ebook-convert you would use: /Applications/calibre.app/Contents/MacOS/ebook-convert.

13.1 Udokumentowane polecenia

13.1.1 calibre

```
calibre [opcje] [ścieżka_do_ebooka lub url calibre ...]
```

Uruchom główny graficzny interfejs użytkownika **calibre** i opcjonalnie dodaj e-book pod adresem path_to_ebook do bazy danych. Możesz także określić adresy URL **calibre**, aby wykonać różne inne działania, niż tylko dodawanie książek. Na przykład:

calibre://view-book/test_library/1842/epub

Otworzy książkę o identyfikatorze 1842 w formacie EPUB z biblioteki „test_library” w przeglądarce e-booków **calibre**. Nazwy bibliotek to nazwy folderów plików biblioteki ze spacjami zastąpionymi podkreśleniami. Pełny opis ww różne działania oparte na adresach URL znajdują się w Podręczniku użytkownika.

Ilekoć przekazujesz argumenty do **calibre**, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

[opcje]

--detach

Odłącz od terminala sterującego, jeśli istnieje (tylko Linux)

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--ignore-plugins

Pomiń niestandardowe wtyczki - przydatne, gdy zainstalujesz wtyczkę, która blokuje uruchomienie calibre

--no-update-check

Nie sprawdzaj czy uaktualnienie jest dostępne

--shutdown-running-calibre, -s

Powoduje zamknięcie wszystkich uruchomionych instancji calibre. Należy pamiętać, że wszystkie niezakończone zadania zostaną przerwane więc należy używać tego przełącznika ostrożnie.

--start-in-tray

Uruchom zminimalizowany w zasobniku systemowym.

--verbose, -v

Pominięto, nie używać. Istnieje tylko dla wstecznej zgodności

--version

wyświetla wersję programu i kończy działanie

--with-library

Użyj biblioteki w podanej lokalizacji.

13.1.2 calibre-customize

```
calibre-customize options
```

Dostosuj calibre poprzez załadowanie zewnętrznych wtyczek.

Ileokroć przekazujesz argumenty do **calibre-customize**, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „./some path/with spaces”

[opcje]

--add-plugin, -a

Dodaj wtyczkę, podając ścieżkę do pliku, który ją zawiera.

--build-plugin, -b

Dla twórców wtyczek: ścieżka do folderu, w którym tworzysz wtyczkę. To polecenie automatycznie spakuje wtyczkę i zaktualizuje ją w Calibre.

--customize-plugin

Customize plugin. Specify name of plugin and customization string separated by a comma. The customization string is the same as you would enter when customizing the plugin in the main calibre GUI.

--disable-plugin

Wyłącz wtyczkę

--enable-plugin

Włącz wtyczkę

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--list-plugins, -l

Wyświetl wszystkie zainstalowane wtyczki

--remove-plugin, -r

Usuń własną wtyczkę - wg nazwy. Nie ma wpływu na wbudowane wtyczki

--version

wyświetla wersję programu i kończy działanie

13.1.3 calibre-debug`calibre-debug [opcje]`

Różne interfejsy wiersza poleceń przydatne do debugowania calibre. Bez opcji, to polecenie uruchamia osadzony interpreter Pythona. Możesz także uruchomić main calibre GUI, przeglądarka e-booków calibre i edytor calibre w trybie debugowania.

Zawiera również interfejsy do różnych bitów calibre, których nie ma dedykowane narzędzia wiersza poleceń, takie jak ustawianie podrzędnych czcionek, narzędzie do porównywania e-booków i tak dalej na.

Możesz także użyć **calibre-debug** do uruchamiania samodzielnych skryptów. Aby to zrobić, użyj go w ten sposób:

```
calibre-debug -e myscript.py -- --option1 --option2 file1 file2 ...
```

Wszystko po `--` jest przekazywane do skryptu. Możesz także użyć **calibre-debug** jako shebang w skryptach, jak to:

```
#!/usr/bin/env -S calibre-debug -e -- -
```

Ileokroć przekazujesz argumenty do **calibre-debug**, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „./some path/with spaces”

[opcje]**--add-simple-plugin**

Dodaj prostą wtyczkę (tj. wtyczkę, składającą się wyłącznie z pliku .py), podając ścieżkę do pliku py, zawierającego kod programu.

--command, -c

Uruchom kod Pythona.

--debug-device-driver, -d

Debuguj wykrycie urządzenia

--default-programs

(Wy)rejestruj calibre z domyślnych programów Windows. `--default-programs` (strona 305) = (register|unregister)

--diff

Uruchom narzędzie porównywania calibre. Np: `calibre-debug --diff` (strona 305) plik1 plik2

--edit-book

Uruchom narzędzie calibre „Edytuj książkę” w trybie debugowania.

--exec-file, -e

Uruchom kod Pythona w pliku.

--explode-book, -x

Rozbij książkę do określonego folderu. Użycie: -x plik.epub katalog_wyjściowy Eksportuje książkę jako zbiór plików HTML i metadanych, które można edytować za pomocą standardowych narzędzi do edycji HTML. Działa z plikami EPUB, AZW3, HTMLZ i DOCX.

--export-all-calibre-data

Wyeksportuj wszystkie dane calibre (książki/ustawienia/wtyczki). Zwykle zostaniesz poproszony o podanie folderu eksportu i bibliotek do wyeksportowania. Można również określić je jako argumenty wiersza poleceń, aby pominąć pytania. Użyj bezwzględnych ścieżek dla folderu eksportu i bibliotek. Specjalne słowo kluczowe "all" może być użyte do wyeksportowania wszystkich bibliotek. Przykłady: calibre-debug --export-all-calibre-data (strona 306) # for interactive use calibre-debug --export-all-calibre-data (strona 306) /path/to/empty/export/folder /path/to/library/folder1 /path/to/library2 calibre-debug --export-all-calibre-data (strona 306) /export/folder all # export all known libraries

--fix-multiprocessing

Tylko do użytku wewnętrznego

--gui, -g

Uruchom GUI w trybie debugowania. Komunikaty są kierowane na standardowe wyjście (stdout i stderr).

--gui-debug

Uruchom z konsolą debugowania i zapisywaniem logu we wskazanej ścieżce. Wyłącznie dla programistów, użyj opcji -g aby uruchomić GUI w trybie debugowania

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--implode-book, -i

Łączenie wcześniej podzielonej książki Użycie: -i katalog_wyjściowy plik.epub Importuje książkę z plików znajdujących się w katalog_wyjściowy, który został utworzony przez wcześniejsze wywołanie --explode-book (strona 306). Pamiętaj aby wybrać taki sam typ pliku jak podczas dzielenia książki.

--import-calibre-data

Importuj uprzednio eksportowane dane

--inspect-mobi, -m

Sprawdź plik(i) w podanym katalogu

--paths

Wyświetl ścieżki niezbędne do ustawienia środowiska calibre

--run-plugin, -r

Uruchom wtyczkę, która udostępnia interfejs wiersza poleceń. Na przykład: calibre-debug -r "Nazwa wtyczki" --file1 --option1 Wszystko po -- zostanie przekazane do wtyczki jako argumenty.

--run-test, -t

Uruchom nazwane testy. Użyj specjalnej wartości „all”, aby uruchomić wszystkie testy. Jeśli nazwa testu zaczyna się od kropki, zakłada się, że jest to nazwa modułu. Jeśli nazwa testu zaczyna się od @, zakłada się, że jest to nazwa kategorii.

--run-without-debug

Nie uruchamiaj z ustawioną flagą DEBUG

--shutdown-running-calibre, -s

Powoduje zamknięcie wszystkich uruchomionych instancji calibre. Należy pamiętać, że wszystkie niezakończone zadania zostaną przerwane więc należy używać tego przełącznika ostrożnie.

--subset-font, -f

Zredukuj określoną czcionkę. Użycie -- po tej opcji należy podać parametry przekazywane do programu redukującego czcionki.

--test-build

Testuj moduły binarne w buildzie

--version

wyświetla wersję programu i kończy działanie

--viewer, -w

Uruchom podgląd książek w trybie debugowania

13.1.4 calibre-server

```
calibre-server [opcje] [ścieżka do folderu biblioteki...]
```

Uruchom serwer treści calibre. Serwer zawartości calibre ujawnia Twoje Biblioteki calibre przez Internet. Możesz określić ścieżkę do biblioteki foldery jako argumenty do **calibre-server**. Jeśli nie określisz żadnych ścieżek, wszystkie zostaną użyte biblioteki, o których wie główny program calibre.

Ileokroć przekazujesz argumenty do **calibre-server**, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „./some path/with spaces”

[opcje]

--access-log

Ścieżka do pliku dziennika dostępu. Ten dziennik zawiera informacje o klientach łączących się z serwerem i wysyłających żądania. Domyślnie rejestrowanie dostępu nie jest wykonywane.

--ajax-timeout

Czas (w sekundach) oczekiwania na odpowiedź z serwera podczas wykonywania zapytań.

--auth-mode

Wybierz rodzaj używanego uwierzytelnienia. Ustaw tryb uwierzytelniania HTTP używany przez serwer. Ustaw jako „podstawowy”, jeśli umieszczasz ten serwer za proxy SSL. W przeciwnym razie pozostaw to jako „auto”, co spowoduje użycie „podstawowego”, jeśli skonfigurowano SSL, w przeciwnym razie użyje „podsumowania”.

--auto-reload

Automatycznie przeładuj serwer, gdy zmieni się kod źródłowy. Przydatny do rozwoju. Należy również podać niewielką wartość limitu czasu wyłączenia.

--ban-after

Liczba nieudanych logowań powodująca blokadę. Liczba nieudanych logowań, po których adres IP jest blokowany

--ban-for

Blokuj adresy IP, z których dokonano wielokrotnych nieudanych logowań. Tymczasowo blokuje dostęp dla adresów IP, z których dokonano wielokrotnych nieudanych logowań przez określoną liczbę minut. Opcja przydatna, aby zablokować próby odgadnięcia haseł. Jeśli ustawiona jest na zero, blokada nie będzie działała.

--book-list-mode

Wybierz domyślny tryb wyświetlania książek. Ustaw domyślny tryb listy książek, który będzie używany dla nowych użytkowników. Poszczególni użytkownicy mogą zastąpić ustawienia domyślne we własnych ustawieniach. Domyślnie jest używana siatka okładki.

--compress-min-size

Minimalny rozmiar, dla którego odpowiedzi wykorzystują kompresję danych (w bajtach).

--custom-list-template

Ścieżka do pliku JSON, zawierającego szablon dla własnego wyglądu listy książek. Najłatwiejszym sposobem na utworzenie takiego pliku szablonu jest przejście do Ustawienia-> Współdzielenie poprzez sieć-> Szablon listy książek w calibre, utworzenie szablonu i wyeksportowanie go..

--daemonize

Uruchom proces w tle jako daemon (tylko Linux).

--displayed-fields

Ogranicz wyświetlane pól zdefiniowanych przez użytkownika. Oddzielone przecinkami lista zdefiniowanych przez użytkownika pól metadanych, które będą wyświetlane przez serwer zawartości w widokach /opds i /mobile. Jeśli określisz tę opcję, żadne pola, których nie ma na tej liście, nie będą wyświetlane. Na przykład: my_rating,my_tags

--enable-allow-socket-preallocation, --disable-allow-socket-preallocation

Wstępna alokacja gniazda, na przykład z aktywacją gniazda systemowego. Domyślnie ta opcja jest włączona.

--enable-auth, --disable-auth

Uwierzytelnianie na podstawie hasła w celu uzyskania dostępu do serwera. Domyślnie serwer daje każdemu nieograniczony dostęp. Za pomocą tej opcji można ograniczyć dostęp tylko do wstępnie zdefiniowanych użytkowników. Domyślnie ta opcja jest wyłączona.

--enable-fallback-to-detected-interface,

--disable-fallback-to-detected-interface

Powrót do automatycznie wykrywanego interfejsu. Jeśli z jakiegoś powodu serwer nie może połączyć się z interfejsem określonym w opcji Listen_on, spróbuje wykryć interfejs, który łączy się ze światem zewnętrznym i połączyć się z nim. Domyślnie ta opcja jest włączona.

--enable-local-write, --disable-local-write

Zezwalaj nieuwierzytelnionym połączeniom lokalnym na wprowadzanie zmian. Zwykle, jeśli nie włączysz uwierzytelniania, serwer działa w trybie tylko do odczytu, aby anonimowi użytkownicy nie mogli wprowadzać zmian w bibliotekach calibre. Ta opcja umożliwia każdemu łączącemu się z tego samego komputera, na którym działa serwer, wprowadzanie zmian. Jest to przydatne, jeśli chcesz uruchomić serwer bez uwierzytelniania, ale nadal używać programu calibredb do wprowadzania zmian w bibliotekach calibre. Pamiętaj, że włączenie tej opcji oznacza, że każdy program uruchomiony na komputerze może wprowadzać zmiany w bibliotekach calibre. Domyślnie ta opcja jest wyłączona.

--enable-log-not-found, --disable-log-not-found

Zapisuj do dziennika żądania HTTP 404 (Nie znaleziono). Domyślnie serwer zapisuje do dziennika wszystkie żądania HTTP do zasobów, które nie zostały znalezione. Może to powodować generowanie wielu śmieciowych wpisów do dziennika, zwłaszcza jeśli twój serwer stanie się celem botów. Użyj tej opcji, aby wyłączyć zapis tych żądań do dziennika. Domyślnie ta opcja jest włączona.

--enable-use-bonjour, --disable-use-bonjour

Reklamuj kanały OPDS przez Bonjour. Reklamuj kanały OPDS za pośrednictwem usługi Bonjour, aby aplikacje czytające oparte na OPDS mogły automatycznie wykrywać i łączyć się z serwerem. Domyślnie ta opcja jest włączona.

--enable-use-sendfile, --disable-use-sendfile

Przesyłanie plików bez kopiowania w celu zwiększenia wydajności. Spowoduje to użycie transferów w jądrze bez kopiowania podczas wysyłania plików przez sieć, zwiększając wydajność. Jednak może to spowodować uszkodzone transfery plików w niektórych uszkodzonych systemach plików. Jeśli zauważysz uszkodzone transfery plików, wyłącz je. Domyślnie ta opcja jest włączona.

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--ignored-fields

Ignorowane pola metadanych. Oddzielona przecinkami lista zdefiniowanych przez użytkownika pól metadanych, które nie będą wyświetlane przez serwer zawartości w widokach /opds i /mobile. Na przykład: my_rating,my_tags

--listen-on

Interfejs, na którym będziemy nasłuchiwać połączeń. Domyślnym ustawieniem jest nasłuchiwanie na wszystkich dostępnych interfejsach IPv6 i IPv4. Można to zmienić na przykład na "127.0.0.1", aby nasłuchiwać tylko połączeń IPv4 z komputera lokalnego, lub na "0.0.0.0", aby nasłuchiwać wszystkich przychodzących połączeń IPv4.

--log

Ścieżka do pliku dziennika dla dziennika serwera. Ten dziennik zawiera informacje o serwerze i błędy, a nie dzienniki dostępu. Domyślnie jest zapisywany na standardowe wyjście.

--manage-users

Zarządzaj bazą danych użytkowników uprawnionych do łączenia się z tym serwerem. Możesz go używać w trybie automatycznym, dodając `-`. Zobacz `calibre-server --manage-users` (strona 309) `-- help`, aby uzyskać szczegółowe informacje. Zobacz także opcję `--userdb` (strona 310).

--max-header-line-size

Max. rozmiar pojedynczego nagłówka HTTP (w KB).

--max-job-time

Maksymalny czas na procesy robocze. Maksymalny czas działania procesów roboczych (w minutach). Ustaw na zero bez ograniczeń.

--max-jobs

Maksymalna liczba procesów roboczych. Procesy robocze są uruchamiane w razie potrzeby i wykorzystywane do dużych zadań, takich jak przygotowywanie książki do przeglądania, dodawanie książek, konwersja itp. Zwykle maks. liczba takich procesów zależy od liczby rdzeni procesora. Możesz kontrolować to za pomocą tego ustawienia.

--max-log-size

Maks. rozmiar pliku dziennika (w MB). Maksymalny rozmiar plików dziennika generowanych przez serwer. Kiedy dziennik staje się większy niż ten rozmiar, jest automatycznie obracany. Ustaw na zero, aby wyłączyć rotację dziennika.

--max-opds-items

Maksymalna liczba książek w kanałach OPDS. Maksymalna liczba książek, które serwer zwróci w pojedynczym kanale pobierania OPDS.

--max-opds-ungrouped-items

Maksymalna liczba nie pogrupowanych elementów w kanałach OPDS. Grupuj elementy w kategorii takie jak autor/etykiety po pierwszej literze gdy zaistnieje więcej niż podana liczba elementów. Ustaw na zero aby wyłączyć.

--max-request-body-size

Max. dozwolony rozmiar plików przesłanych na serwer (w MB).

--num-per-page

Liczba książek wyświetlanych na pojedynczej stronie. Liczba książek wyświetlanych na pojedynczej stronie w przeglądarce.

--pidfile

Zapisz identyfikator procesu do określonego pliku

--port

Port, na którym będziemy nasłuchiwać połączeń.

--search-the-net-urls

Ścieżka do pliku JSON zawierającego adresy URL funkcji „Szukaj w internecie”. Najłatwiejszym sposobem utworzenia takiego pliku jest przejście do Preferencje-> Udostępnianie przez sieć-> Przeszukaj internet w calibre, utwórz adresy URL i wyeksportuj je.

--shutdown-timeout

Całkowity czas w sekundach oczekiwania na pełne wyłączenie.

--ssl-certfile

Ścieżka do pliku z certyfikatem SSL.

--ssl-keyfile

Ścieżka do pliku z kluczem prywatnym SSL.

--timeout

Czas (w sekundach), po którym bezczynne połączenie zostanie zamknięte.

--trusted-ips

Zezwalaj na nieuwierzytelnione połączenia z określonych adresów IP, aby wprowadzać zmiany. Zwykle, jeśli nie włączysz uwierzytelniania, serwer działa w trybie tylko do odczytu, aby anonimowi użytkownicy nie mogli wprowadzać zmian w bibliotekach calibre. Ta opcja umożliwia każdemu łączącemu się z określonych adresów IP wprowadzanie zmian. Musi to być lista adresów lub specyfikacji sieci oddzielonych przecinkami. Jest to przydatne, jeśli chcesz uruchomić serwer bez uwierzytelniania, ale nadal używać programu calibredb do wprowadzania zmian w bibliotekach calibre. Zwróć uwagę, że włączenie tej opcji oznacza, że każdy łączący się z określonych adresów IP może wprowadzać zmiany w bibliotekach calibre.

--url-prefix

Prefiks poprzedzający wszystkie adresy URL. Przydatne, jeśli chcesz uruchomić ten serwer za zwrotnym proxy. Na przykład użyj /calibre jako prefiksu adresu URL.

--userdb

Ścieżka do bazy danych użytkowników, która ma być używana do uwierzytelniania. Baza danych to plik SQLite. Aby go utworzyć, użyj `--manage-users` (strona 309). Więcej informacji na temat zarządzania użytkownikami można znaleźć pod adresem: <https://manual.calibre-ebook.com/pl/server.html#managing-user-accounts-from-the-command-line-only>

--version

wyświetla wersję programu i kończy działanie

--worker-count

Liczba wątków roboczych używanych do przetwarzania żądań.

13.1.5 calibre-smtp

```
calibre-smtp [opcje] [from to text]
```

Wyślij mail przy użyciu protokołu SMTP. **calibre-smtp** ma dwa tryby pracy. W trybie tworzenia wiadomości określasz od kogo, do kogo i tekst wiadomości, na podstawie których jest budowana, a następnie wysyłana wiadomość. W trybie filtrowania **calibre-smtp** odczytuje dane z STDIN i wysyła je. text jest treścią wiadomości. Jeśli treść nie jest podana, jest ona odczytywana z STDIN. to jest adresem e-mailowym odbiorcy wiadomości, zaś from - adresem nadawcy. Jeśli kompletna wiadomość jest odczytywana z STDIN adresy nadawcy i odbiorcy są używane jedynie do negocjacji w protokole SMTP, nagłówki wiadomości nie są modyfikowane.

Ileokroć przekazujesz argumenty do **calibre-smtp**, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „./some path/with spaces”

[opcje]

--fork, -f

Oddziel i dostarcz wiadomość w tle. Jeśli użyjesz tej opcji, powinieneś użyć również parametru **--outbox**, by móc zareagować na błędy przy dostarczaniu wiadomości.

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--localhost, -l

Lokalna nazwa komputera. Używana przy połączeniu z serwerem SMTP.

--outbox, -o

Ścieżka do folderu, w którym będą przechowywane wiadomości, których nie udało się wysłać.

--timeout, -t

Limit czasu połączenia

--verbose, -v

Podaj więcej danych

--version

wyświetla wersję programu i kończy działanie

COMPOSE MAIL

Opcje tworzenia wiadomości. Ignorowane, jeśli treść wiadomości nie została podana

--attachment, -a

Plik do załączenia do maila

--subject, -s

Temat maila

SMTP RELAY

Opcje, aby korzystać z serwera SMTP do wysyłania poczty. calibre będzie próbował wysłać e-mail bezpośrednio, chyba, że `--relay` jest określony.

--cafile

Ścieżka do pliku połączonych certyfikatów CA w formacie PEM, służącym do weryfikacji certyfikatu serwera podczas korzystania z TLS. Domyślnie używane są certyfikaty systemu CA.

--dont-verify-server-certificate

Nie weryfikuj certyfikatu serwera podczas łączenia za pomocą TLS. To było domyślne zachowanie w wersjach calibre starszych niż 3.27. Jeśli używasz przekaźnika z samopodpisanym lub w inny sposób nieważnym certyfikatem, możesz użyć tej opcji, aby przywrócić zachowanie wcześniejsze niż 3.27

--encryption-method, -e

Metoda szyfrowania używana przy połączeniu z serwerem pocztowym. Do wyboru są: TLS, SSL lub NONE. Domyślną jest TLS. UWAGA: wybranie NONE jest bardzo niebezpieczne

--password, -p

Hasło

--port

Port do połączenia z serwerem pocztowym. Domyślnie jest to 465 jeśli używana się SSL, a 25 w innym przypadku.

--relay, -r

Serwer SMTP, używany do wysyłania poczty.

--username, -u

Nazwa użytkownika

13.1.6 calibredb

```
calibredb polecenie [opcje] [argumenty]
```

`command:calibredb` jest interfejsem wiersza poleceń do bazy danych calibre. Posiada kilka podkomend, udokumentowanych poniżej.

komenda:`calibredb` może być użyta do manipulowania bazą danych calibre określoną przez ścieżkę lub calibre *Content server* działającą na komputerze na komputerze lokalnym lub przez Internet. Można uruchomić calibre *guilabel:Serwer treści* używając **calibre-server** lub w głównym programie calibre klikając *Connect/share* → *Start Content server*. Ponieważ **calibredb** może dokonywać zmian w bibliotekach calibre, należy najpierw skonfigurować uwierzytelnianie na serwerze. Istnieją są na to dwa sposoby:

- Jeśli planujesz łączyć się tylko z serwerem uruchomionym na tym samym komputerze,

można po prostu użyć opcji `--enable-local-write` serwera Content, aby zezwolić dowolnemu programowi, w tym `calibredb`, działającemu na komputerze na komputerze lokalnym na dokonywanie zmian w danych calibre. Podczas uruchamiania serwera z głównego programu calibre, opcja ta znajduje się w *guilabel:Preferencje->Udostępnianie przez sieć->Zaawansowane*.

- Jeśli chcesz włączyć dostęp przez Internet, powinieneś skonfigurować konta użytkowników na serwerze i użyć

konta użytkowników na serwerze i użyć opcji `--username` (strona 314) i `--password` (strona 314). aby nadać im dostęp. Można skonfigurować uwierzytelnianie użytkownika dla **calibre-server** używając opcji `--enable-auth`. i używając `--manage-users` do utworzenia kont użytkowników. Jeśli uruchamiasz serwer z głównego programu calibre, użyj *guilabel:Preferencje->Sharing over the net->Require username/password*.

Aby połączyć się z uruchomionym serwerem treści, przekaż adres URL serwera do opcji `--with-library` (strona 314), zobacz dokumentację tej opcji dla szczegóły i przykłady.

- *OPCJE GLOBALNE* (strona 314)
- *list* (strona 314)
- *add* (strona 315)
 - *DODAWANIE Z FOLDERÓW* (strona 316)
- *remove* (strona 317)
- *add_format* (strona 317)
- *remove_format* (strona 317)
- *show_metadata* (strona 317)
- *set_metadata* (strona 318)
- *export* (strona 318)
- *catalog* (strona 319)
 - *EPUB OPCJE* (strona 320)
- *saved_searches* (strona 321)
- *add_custom_column* (strona 322)
- *custom_columns* (strona 322)
- *remove_custom_column* (strona 322)
- *set_custom* (strona 323)
- *restore_database* (strona 323)
- *check_library* (strona 323)
- *list_categories* (strona 324)
- *backup_metadata* (strona 324)
- *clone* (strona 324)
- *embed_metadata* (strona 325)
- *search* (strona 325)
- *fts_index* (strona 325)
- *fts_search* (strona 326)

OPCJE GLOBALNE

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--library-path, --with-library

Ścieżka do biblioteki calibre. Domyślnie używana jest ścieżka zapisana w ustawieniach. Możesz także połączyć się z serwerem zawartości calibre, aby wykonywać działania na zdalnych bibliotekach. Aby to zrobić, użyj adresu URL w postaci: `http://nazwa_hosta:port/#id_biblioteki`, na przykład `http://localhost:8080/#mylibrary.id_biblioteki` to identyfikator biblioteki, z którą chcesz się połączyć na serwerze zawartości. Możesz użyć specjalnej wartości `Library_id` wynoszącej `-`, aby uzyskać listę identyfikatorów bibliotek dostępnych na serwerze. Aby uzyskać szczegółowe informacje na temat konfigurowania dostępu przez serwer treści, zobacz <https://manual.calibre-ebook.com/pl/generated/pl/calibredb.html>.

--password

Hasło do połączenia z serwerem treści calibre. Aby odczytać hasło ze standardowego wejścia, użyj specjalnej wartości: `<stdin>`. Aby odczytać hasło z pliku, użyj: `<f:/path/to/file>` (tj. `<f:` followed by the full path to the file and a trailing `>`). Wymagane są nawiasy kątowe podane powyżej, pamiętaj, aby je uciec lub użyj cudzysłowu dla swojej powłoki.

--timeout

Limit czasu w sekundach podczas łączenia się z biblioteką calibre przez sieć. Wartość domyślna to dwie minuty.

--username

Nazwa użytkownika służąca do połączenia się z serwerem zawartości calibre

--version

wyświetla wersję programu i kończy działanie

list

```
calibredb list [options]
```

Wyświetl książki dostępne w bazie danych calibre.

Ilekoć przekazujesz argumenty do `calibredb`, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: `„some path/with spaces”`

--ascending

Posortuj wyniki rosnąco

--fields, -f

Pola do wyświetlenia podczas wyświetlania książek w bazie danych. Powinna być oddzielona przecinkami lista pól Dostępne pola: `author_sort`, `authors`, `comments`, `cover`, `formats`, `identifiers`, `isbn`, `languages`, `last_modified`, `pubdate`, `publisher`, `rating`, `series`, `series_index`, `size`, `tags`, `template`, `timestamp`, `title`, `uuid` Domyślnie: `%` domyślnie. Do wybrania wszystkich pól można użyć specjalnego pola „wszystkie”. Oprócz wbudowanych pól powyżej pola niestandardowe są również dostępne jako `*nazwa_pola`, na przykład dla niestandardowego pola `#ocena`, użyj nazwy: `*ocena`

--for-machine

Generuj wynik w formacie JSON, bardziej odpowiednim do przetwarzania maszynowego. W tym przypadku szerokość wiersza i separator są ignorowane.

--limit

Maksymalna liczba wyświetlanych rezultatów. Domyślnie: `all`

--line-width, -w

Maksymalna szerokość pojedynczej linii w pliku wynikowym. Domyślnie dostosowana do wykrytego rozmiaru ekranu.

--prefix

Przedrostek dla wszystkich ścieżek do plików. Domyślnie jest to ścieżka absolutna do folderu biblioteki.

--search, -s

Filtruj wyniki wg zapytania wyszukiwania. Format zapytania wyszukiwania można znaleźć w dokumentacji wyszukiwania w Podręczniku użytkownika. Domyślnie brak filtrowania.

--separator

Znak używany do oddzielania pól. Domyślny to spacja.

--sort-by

Pole, według którego mają być sortowane wyniki. Możesz określić wiele pól, oddzielając je przecinkami. Dostępne pola: author_sort, authors, comments, cover, formats, identifiers, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, size, tags, template, timestamp, title, uuid Domyślnie: id

--template

Szablon do uruchomienia, jeśli na liście pól znajduje się „template”. Domyślnie: Brak

--template_file, -t

Ścieżka do pliku zawierającego szablon do uruchomienia, jeśli na liście pól znajduje się „template”. Domyślnie: Brak

--template_heading

Nagłówek kolumny szablonu. Domyślnie: %domyślnie. Ta opcja jest ignorowana, jeśli ustawiona jest opcja `--for-machine` (strona 314)

add

```
calibredb add [options] file1 file2 file3 ...
```

Dodaj określone pliki jako książki do bazy danych. Możesz także określić foldery, zobacz poniższe opcje związane z folderami.

Ileokroć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--authors, -a

Określ autorów dodanych książek

--automerger, -m

Jeśli zostaną znalezione książki o podobnych tytułach i autorach, automatycznie połącz przychodzące formaty (pliki) z istniejącymi rekordami książek. Wartość „ignoruj” oznacza, że zduplikowane formaty są odrzucane. Wartość „nadpisuj” oznacza, że zduplikowane formaty w bibliotece są nadpisywane nowo dodanymi plikami. Wartość „nowy_rekord” oznacza, że zduplikowane formaty są umieszczane w nowym rekordzie książki.

--cover, -c

Ścieżka do okładki używanej dla dodanej książki

--duplicates, -d

Dodawaj książki do bazy danych, nawet jeśli już istnieją. Porównanie odbywa się na podstawie tytułów książek i autorów. Zauważ, że opcja `--automerger` (strona 315) ma pierwszeństwo.

--empty, -e

Dodaj pustą książkę (książkę bez żadnych formatów)

--identifier, -I

Ustaw identyfikatory tej książki, np. -I asin:XXX -I isbn:YYY

--isbn, -i

Podaj ISBN dodanych książek

--languages, -l

Oddzielona przecinkami lista języków (najlepiej użyć kodów językowych ISO639, choć niektóre nazwy języków mogą również zostać rozpoznane)

--series, -s

Podaj serie dla dodanych książek

--series-index, -S

Określ numer w cyklu dla dodanych książek

--tags, -T

Podaj etykiety dla dodanych książek

--title, -t

Określ tytuły dodanych książek

DODAWANIE Z FOLDERÓW

Opcje kontroli dodawania książek z folderów. Domyślnie dodawane są tylko pliki, które mają rozszerzenia znanych typów plików e-booków.

--add

Wzorzec nazwy pliku (glob), pliki pasujące do tego wzorca zostaną dodane podczas skanowania folderów w poszukiwaniu plików, nawet jeśli nie są one znanego typu pliku e-book. Można określić wielokrotnie dla wielu wzorów.

--ignore

Wzorzec nazwy pliku (glob), pliki pasujące do tego wzorca będą ignorowane podczas skanowania folderów w poszukiwaniu plików. Można określić wielokrotnie dla wielu wzorów. Na przykład: *.pdf zignoruje wszystkie pliki PDF

--one-book-per-directory, -1

Załóżmy, że każdy folder ma tylko jedną książkę logiczną i że wszystkie znajdujące się w nim pliki mają różne formaty e-booków tej książki

--recurse, -r

Przetwarzaj foldery rekurencyjnie

remove

```
calibredb remove ids
```

Usuń książki zidentyfikowane przez identyfikatory z bazy danych. ids powinny być oddzieloną przecinkami listą numerów identyfikacyjnych (numery identyfikacyjne można uzyskać za pomocą polecenia wyszukiwania). Na przykład 23,34,57-85 (przy określaniu zakresu ostatnia liczba z zakresu nie jest uwzględniana).

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--permanent

Nie używaj Kosz

add_format

```
calibredb add_format [opcje] id plik_książki
```

Dodaj plik_książki do dostępnych formatów książki o identyfikatorze id. Identyfikator można uzyskać za pomocą polecenia wyszukiwania. Jeśli format istnieje, zostanie zastąpiony - o ile nie użyto opcji nie zastępować.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--as-extra-data-file

Dodaj plik jako dodatkowy plik danych do książki, a nie jako format ebooka.

--dont-replace

Nie zamieniaj formatu jeśli istnieje

remove_format

```
calibredb remove_format [options] id fmt
```

Usuń format fmt z książki logicznej zidentyfikowanej przez id. Możesz uzyskać id za pomocą polecenia wyszukiwania. fmt powinno być rozszerzeniem pliku, takim jak LRF, TXT lub EPUB. Jeśli książka logiczna nie ma dostępnego fmt, nie rób nic.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

show_metadata

```
calibredb show_metadata [options] id
```

Pokaż metadane przechowywane w bazie danych calibre dla książki identyfikowanej przez identyfikator. id to numer identyfikacyjny z polecenia wyszukiwania.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--as-opf

Wydrukuj metadane w formacie OPF (XML)

set_metadata

```
calibredb set_metadata [options] book_id [/path/to/metadata.opf]
```

Ustaw metadane przechowywane w bazie danych calibre dla książki identyfikowanej przez `book_id` z pliku OPF `metadata.opf`. `book_id` to numer identyfikacyjny książki z polecenie wyszukiwania. Możesz szybko zapoznać się z formatem OPF, używając pliku `--as-opf` przełącz na polecenie `show_metadata`. Możesz także ustawić metadane pliku poszczególnie pola za pomocą opcji `--field`. Jeśli użyjesz opcji `--field`, tam nie ma potrzeby określania pliku OPF.

Ileokroć przekazujesz argumenty do `calibredb`, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--field, -f

Pole do zmiany w formacie `nazwa_pola:wartość`. Na przykład `--field` (strona 318) `tags:etykieta1,etykieta2`. Użyj `--list-fields` (strona 318) jeśli chcesz otrzymać listę wszystkich pól. Możesz użyć tej opcji wielokrotnie, aby zmienić jednocześnie wiele pól. Uwaga: dla języków należy użyć kodów języków zgodnych z ISO639 (tzn. en dla angielskiego, fr dla francuskiego, pl dla polskiego itd.) Dla identyfikatorów należy użyć składni `--field` (strona 318) `identifiers:isbn:XXXX,doi:YYYYY`. W przypadku pól boolean (tak/nie) należy użyć `true/false` albo `tak/nie`.

--list-fields, -l

Lista pól, które mogą być użyte z opcją `--field` (strona 318)

export

```
calibredb export [opcje] ids
```

Eksportuje książki określone przez `ids` (lista rozdzielana przecinkami) do systemu plików. Operacja eksportu zapisuje wszystkie formaty książki, jej okładkę i metadane (w pliku pliku OPF). Wszelkie dodatkowe pliki danych powiązane z książką są również zapisywane. Numery id można uzyskać za pomocą polecenia wyszukiwania.

Ileokroć przekazujesz argumenty do `calibredb`, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--all

Eksportuj wszystkie książki w bazie, ignoruj listę identyfikatorów.

--dont-asciiize

Przekonwertuj wszystkie inne niż angielskie znaki Calibre na angielskie odpowiedniki nazw plików. Jest to przydatne, jeśli zapisujesz do starszego systemu plików bez pełnej obsługi nazw plików Unicode. Określając ten przełącznik wyłączysz to zachowanie.

--dont-save-cover

Calibre zapisze okładkę w oddzielnym pliku obok właściwego pliku z książką. Określając ten przełącznik wyłączysz to zachowanie.

--dont-save-extra-files

Zapisywanie wszelkich plików danych powiązanych z książką podczas zapisywania książki. Określając ten przełącznik wyłączysz to zachowanie.

--dont-update-metadata

Zwykle calibre uaktualni metadane w zapisanych plikach tymi, które są w bibliotece calibre. Sprawia, że zapisywanie na dysku jest wolniejsze. Określając ten przełącznik wyłączysz to zachowanie.

--dont-write-opf

Calibre zapisze metadane do osobnego pliku OPF obok właściwego pliku z książką. Określając ten przełącznik wyłączysz to zachowanie.

--formats

Oddzielana przecinkami lista formatów do zapisania dla każdej książki. Domyślnie zapisywane są wszystkie dostępne formaty.

--progress

Postęp

--replace-whitespace

Zastąp spacje podkreśleniami.

--single-dir

Eksportuj wszystkie książki do pojedynczego katalogu

--template

Szablon do kontrolowania nazwy pliku i struktury folderów zapisanych plików. Domyślnie jest to „{author_sort}/{title}/{title} - {authors}”, które zapisuje książki w podfolderze dla poszczególnych autorów z nazwami plików zawierającymi tytuł i autora. Dostępne kontrolki to: {author_sort, authors, id, isbn, languages, last_modified, pubdate, publisher, rating, series, series_index, tags, timestamp, title}

--timefmt

Format wyświetlania dat. %d - dzień, %b - miesiąc, %m - numer miesiąca, %Y - rok. Domyślnie: %b, %Y

--to-dir

Eksportuj książki do wybranego katalogu. Domyślny to .

--to-lowercase

Przekształć ścieżki dostępu na małe litery.

catalog

```
calibredb catalog /path/to/destination.(csv|epub|mobi|xml...) [options]
```

Export a **catalog** in format specified by path/to/destination extension. Options control how entries are displayed in the generated **catalog** output. Note that different **catalog** formats support different sets of options. To see the different options, specify the name of the output file and then the `-help` option.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „some path/with spaces”

--ids, -i

Lista ID oddzielona przecinkami do katalogu. Jeśli jest zadeklarowane, `--search` (strona 319) jest ignorowane. Domyślnie: wszystkie (all)

--search, -s

Filtruj wyniki wg zapytania wyszukiwania. Format zapytania wyszukiwania można znaleźć w dokumentacji wyszukiwania w Podręczniku użytkownika. Domyślnie: brak filtrowania

--verbose, -v

Pokazuj szczegółową informację wyjściową. Przydatne przy debugowaniu

EPUB OPCJE

--catalog-title

Tytuł wygenerowanego katalogu używany jako tytuł w metadanych. Domyślnie: 'My Books' Dotyczy formatów wyjściowych: AZW3, EPUB, MOBI

--cross-reference-authors

Tworzy odsyłacze w sekcji Autorzy dla książek z wieloma autorami np. prace zbiorowe. Domyślnie: 'False' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--debug-pipeline

Zapisz dane wyjściowe z różnych etapów potoku konwersji w określonym folderze. Przydatne, jeśli nie masz pewności, na którym etapie procesu konwersji występuje błąd. Domyślnie: „None” Dotyczy formatów wyjściowych: AZW3, EPUB, MOBI

--exclude-genre

Wyrażenie regularne opisujące etykiety do wykluczenia jako gatunki. Domyślnie: '[.+]!^+\$' wyklucza etykiety w nawiasach kwadratowych, tzn. '[Project Gutenberg]', i '+', domyślny tag dla przeczytanych książek. Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--exclusion-rules

Określa reguły używane do wykluczania książek z generowanego katalogu. Schemat reguły wykluczającej to: ('<rule name>', 'Etykieta', '<comma-separated list of tags>') lub ('<rule name>', '<custom column>', '<pattern>'). Na przykład: (('Archiwalne książki', '#status', 'Archiwalne'),) wykluczy książki, które w polu użytkownika 'status' ma wartość 'Archiwalne'. Jeśli zdefiniowano wiele reguł wszystkie są brane pod uwagę. Domyślnie: (('Catalogs', 'Tags', 'Catalog'),) Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--generate-authors

Włącz sekcję "Autorzy" do katalogu. Domyślnie: 'False' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--generate-descriptions

Włącz sekcję "Opisy" do katalogu. Domyślnie: 'False' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--generate-genres

Włącz sekcję "Gatunki" do katalogu. Domyślnie: 'False' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--generate-recently-added

Włącz sekcję "Ostatnio dodane" do katalogu. Domyślnie: 'False' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--generate-series

Włącz sekcję "Serie" do katalogu. Domyślnie: 'False' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--generate-titles

Włącz sekcję "Tytuły" do katalogu. Domyślnie: 'False' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--genre-source-field

Pole źródłowe dla sekcji "Gatunki". Domyślnie: 'Etykiety' Dotyczy formatów wyjściowych: AZW3, EPUB, MOBI

--header-note-source-field

Pole użytkownika zawierające tekst włączany do opisu. Domyślnie: '' Dotyczy formatów wyjściowych: AZW3, EPUB, MOBI

--merge-comments-rule

#<custom field>:[before|after]:[True|False] określa: <custom field> Pole niestandardowe zawierające notatki do scalenia z komentarzami • [before|after] Umieszczanie uwag w stosunku do komentarzy • [True|False] Pomiędzy notatkami i komentarzami wstawiana jest pozioma linia Domyślnie: ' :: ' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--output-profile

Określa profil wyjściowy. W niektórych przypadkach profil wyjściowy jest konieczny do optymalizacji dla konkretnego urządzenia. Na przykład w przypadku 'kindle' lub 'kindle_dx' tworzy strukturę spisu treści z sekcjami i artykułami. Domyślnie: 'None' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--prefix-rules

Określa reguły używane do przypisywania przedrostków oznaczających przeczytane książki, listę życzeń i inne zdefiniowane przez użytkownika. Schemat reguły jest następujący: ('<nazwa reguły>', '<pole źródłowe>', '<wzorzec>', '<przedrostek>'). Jeśli zdefiniowano wiele reguł zostanie zastosowana pierwsza pasująca. Domyślnie: "((('Read books', 'tags', '+', '✓'), ('Wishlist item', 'tags', 'Wishlist', '×')))" Dotyczy formatów: AZW3, EPUB, MOBI

--preset

Użyj zestawu stworzonego za pomocą GUI Catalog buildera. Zestaw zawiera wszystkie ustawienia potrzebne do utworzenia katalogu. Domyślnie: 'None' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--thumb-width

Podpowieź rozmiaru (w calach) dla okładek książek w katalogu. Zakres: 1.0 – 2.0 Domyślnie: '1.0' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

--use-existing-cover

Zamień istniejącą okładkę przy generowaniu katalogu. Domyślnie: 'False' Dotyczy formatów wyjściowych AZW3, EPUB, MOBI

saved_searches

```
calibredb saved_searches [options] (list|add|remove)
```

Zarządzaj zapisanymi wyszukiwaniami przechowywanymi w tej bazie danych. Jeśli spróbujesz dodać zapytanie o nazwie, która już istnieje, będzie to wymieniony.

Składnia dodawania:

```
calibredb saved_searches add search_name search_expression
```

Składnia do usuwania:

```
%p rog saved_searches remove search_name
```

Ilekość przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

add_custom_column

```
calibredb add_custom_column [options] label name datatype
```

Stwórz własną kolumnę. label to nazwa kolumny w języku komputerowym. Nie powinna zawierać spacji ani dwukropka. name to nazwa ludzka kolumny. datatype to jeden z: bool, comments, composite, datetime, enumeration, float, int, rating, series, text

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--display

Słownik opcji, określających jak będą interpretowane dane w tej kolumnie. Są one w formacie JSON. Do wyświetlenia kolumn można użyć `--display` (strona 322) `"{\ "enum_values\ ":[\ "val1\ ", \ "val2\ "]}"`. Zmienna display może przyjąć wiele opcji. Dla kolejnych typów kolumn: złożone: `composite_template`, `composite_sort`, `make_category`, `contains_html`, `use_decorations` daty i czasu: `date_format` wyliczenia: `enum_values`, `enum_colors`, `use_decorations` liczbowe: `number_format` tekstowe: `is_names`, `use_decorations`. Najlepszym sposobem na znalezienie właściwej kombinacji jest stworzenie własnej kolumny odpowiedniego typu w interfejsie użytkownika, a potem zajrzenie do pliku OPF książki (przy założeniu, że plik OPF został uaktualniony po stworzeniu kolumny). Tam znajduje się JSON dla nowej kolumny.

--is-multiple

Ta kolumna przechowuje dane etykietopodobne (np. wielokrotne wartości oddzielone przecinkami). Ma zastosowanie tylko jeśli typ danych to tekst.

custom_columns

```
calibredb custom_columns [opcje]
```

Wyświetla listę dostępnych, własnych kolumn. Pokazuje ich nazwę oraz id.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--details, -d

Pokaż szczegóły dla każdej kolumny.

remove_custom_column

```
calibredb remove_custom_column [options] label
```

Usuń niestandardową kolumnę oznaczoną etykietą. Możesz zobaczyć dostępne kolumny z poleceniem `custom_columns`.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--force, -f

Nie pytaj o potwierdzenie

set_custom

```
calibredb set_custom [options] column id value
```

Ustaw wartość kolumny niestandardowej dla książki identyfikowanej przez identyfikator. Możesz uzyskać listę identyfikatorów za pomocą polecenia wyszukiwania. Możesz uzyskać listę niestandardowych nazw kolumn za pomocą `custom_columns` Komenda.

Ilekoć przekazujesz argumenty do `calibredb`, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--append, -a

Jeśli kolumna przechowuje wielokrotne wartości, dołącz określone wartości do istniejących, zamiast je zastępować.

restore_database

```
calibredb restore_database [options]
```

Przywróć tę bazę danych z metadanych przechowywanych w plikach OPF w każdym folder biblioteki calibre. Jest to przydatne, jeśli plik `metadata.db` został uszkodzony.

OSTRZEŻENIE: To polecenie całkowicie regeneruje twoją bazę danych. Przegrasz wszystkie zapisane wyszukiwania, kategorie użytkowników, wtyczki, zapisane konwersje według książki ustawienia i niestandardowe przepisy. Przywrócone metadane będą tak dokładne, jak co znajduje się w plikach OPF.

Ilekoć przekazujesz argumenty do `calibredb`, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--really-do-it, -r

Naprawdę dokonaj przywrócenia. Ta komenda nie uruchomi się dopóki ta opcja nie zostanie określona.

check_library

```
calibredb check_library [options]
```

Wykonaj sprawdzenie w systemie plików biblioteki. Zgłoszenia są `invalid_titles`, `extra_titles`, `invalid_authors`, `extra_authors`, `missing_formats`, `extra_formats`, `extra_files`, `missing_covers`, `extra_covers`, `failed_folders`

Ilekoć przekazujesz argumenty do `calibredb`, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--csv, -c

Wynik w CSV

--ignore_extensions, -e

Lista rozszerzeń rozdzielona przecinkiem do zignorowania. Domyślnie: wszystkie

--ignore_names, -n

Lista nazw rozdzielona przecinkiem do zignorowania. Domyślnie: wszystkie

--report, -r

Lista raportów rozdzielona przecinkiem. Domyślnie: wszystkie

--vacuum-fts-db

Wyczyść bazę danych wyszukiwania pełnotekstowego. W zależności od rozmiaru bazy danych może to być bardzo powolne i wymagać dużej ilości pamięci.

list_categories

```
calibredb list_categories [options]
```

Utwórz raport informacji o kategoriach w bazie danych. The informacja jest odpowiednikiem tego, co jest wyświetlane w przeglądarce znaczników.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--categories, -r

Lista kategorii wyszukiwań nazw oddzielona przecinkami. Domyślnie: wszystkie

--csv, -c

Wynik w CSV

--dialect

Rodzaj pliku CSV do utworzenia. Wybór: excel, excel-tab, unix

--item_count, -i

Wyprowadź tylko taką ilość elementów w kategorii zamiast wyliczać na każdy element wewnątrz kategorii

--width, -w

Maksymalna szerokość pojedynczej linii w pliku wynikowym. Domyślnie dostosowana do wykrytego rozmiaru ekranu.

backup_metadata

```
calibredb backup_metadata [options]
```

Utwórz kopię zapasową metadanych przechowywanych w bazie danych w osobnych plikach OPF w każdym folder książek. Zwykle dzieje się to automatycznie, ale możesz to uruchomić polecenie, aby wymusić ponowne generowanie plików OPF, z opcją **-all**.

Pamiętaj, że zwykle nie ma takiej potrzeby, ponieważ kopia zapasowa plików OPF jest tworzona automatycznie przy każdej zmianie metadanych.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

--all

Domyślnie to polecenie operuje na książkach, które mają nieaktualne pliki OPF. Ta opcja powoduje, że działaniem zostaną objęte wszystkie książki.

clone

```
calibredb clone path/to/new/library
```

Utwórz klon bieżącej biblioteki. Tworzy to nową, pustą bibliotekę, która zawiera wszystkie takie same kolumny niestandardowe, biblioteki wirtualne i inne ustawienia, jak bieżąca biblioteka.

Sklonowana biblioteka nie będzie zawierać żadnych książek. Jeśli chcesz utworzyć pełny duplikat, w tym wszystkie książki, a następnie po prostu użyj narzędzi systemu plików, aby skopiować folder biblioteki.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

embed_metadata

```
calibredb embed_metadata [opcje] id_książki
```

Uaktualnij metadane w plikach książki z biblioteki calibre na podstawie bazy danych biblioteki. Domyślnie metadane są uaktualniane jedynie podczas eksportowania plików z calibre, to polecenie pozwala na uaktualnienie ich bez eksportu. Należy pamiętać, że różne formaty obsługują różne metadane. Można użyć specjalnej wartości «all» jako id_książki, by uaktualnić metadane we wszystkich książkach w bibliotece. Można również podać wiele identyfikatorów książek, oddzielając je spacjami oraz zakresy identyfikatorów, w których są one oddzielone myślnikami. Na przykład: **calibredb embed_metadata 1 2 10-15 23**

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „/some path/with spaces”

--only-formats, -f

Uaktualnij metadane wyłącznie w plikach o podanym formacie. Podaj wielokrotnie dla wielu formatów. Domyślnie uaktualniane są wszystkie formaty.

search

```
calibredb search [options] search expression
```

Przeszukaj bibliotekę dla określonego wyszukiwanego terminu, zwracając przecinek rozdzielony lista identyfikatorów książek pasujących do wyrażenia wyszukiwania. Format wyjściowy jest przydatny do zasilania innych poleceń, które akceptują listę identyfikatorów jako dane wejściowe.

Wyrażenie wyszukiwania może być dowolne z potężnego zapytania wyszukiwania calibre język, na przykład: **calibredb search author:asimov «title:"i robot"»**

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „/some path/with spaces”

--limit, -l

Maksymalna liczba wyświetlanych rezultatów. Domyślnie wszystkie.

fts_index

```
calibredb fts_index [opcje] enable/disable/status/reindex
```

Kontroluje proces indeksowania wyszukiwania pełnotekstowego.

enable Włącza indeksowanie FTS dla tej biblioteki **disable** Wyłącza indeksowanie FTS dla tej biblioteki **status** Pokazuje bieżący status indeksowania **reindex** Może być użyty do ponownego indeksowania poszczególnych książek lub całej biblioteki. Aby ponownie zindeksować poszczególne książki należy podać identyfikatory książek jako dodatkowe argumenty po poleceniu **reindex**. Jeśli nie podano identyfikatorów książek indeksowana jest cała biblioteka.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „/some path/with spaces”

--indexing-speed

Szybkość indeksowania. Użyj szybko, aby szybko indeksować przy użyciu wszystkich zasobów komputera, i wolno, aby indeksować mniej zasobów. Zauważ, że prędkość jest resetowana do niskiej po każdym wywołaniu.

--wait-for-completion

Poczekaj, aż wszystkie książki zostaną zindeksowane, okresowo pokazując postęp indeksowania

fts_search

```
calibredb fts_search [options] search expression
```

Wykonaj wyszukiwanie pełnotekstowe w całej bibliotece lub jej podzbiorze.

Ilekoć przekazujesz argumenty do calibredb, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „/some path/with spaces”

--do-not-match-on-related-words

Dopasuj tylko dokładne słowa, które nie są powiązane. Więc korekta nie będzie pasować do korekty.

--include-snippets

Dołącz fragmenty tekstu otaczającego każde dopasowanie. Zauważ, że to znacznie spowalnia wyszukiwanie.

--indexing-threshold

Jaka część biblioteki musi zostać zindeksowana, zanim wyszukiwanie będzie dozwolone, w procentach. Domyślnie 90

--match-end-marker

Znacznik używany do wskazania końca dopasowanego słowa we fragmencie

--match-start-marker

Znacznik używany do wskazania początku dopasowanego słowa we fragmencie

--output-format

Format, w którym mają być wyświetlane wyniki wyszukiwania. „text” dla zwykłego tekstu lub „json” dla danych wyjściowych JSON.

--restrict-to

Ogranicz wyszukiwane książki, używając wyrażenia wyszukiwania lub identyfikatorów. Na przykład: ids:1,2,3, aby ograniczyć według identyfikatorów lub search:tag:foo, aby ograniczyć do książek z tagiem foo.

13.1.7 ebook-convert

```
ebook-convert input_file output_file [options]
```

Konwertuj e-booka z jednego formatu na inny.

input_file to dane wejściowe, a output_file to dane wyjściowe. Oba muszą być podane jako pierwsze dwa argumenty polecenia.

Wyjściowy format e-booka jest odgadywany na podstawie rozszerzenia pliku output_file. plik_wyjściowy może mieć również specjalny format .EXT, gdzie EXT jest rozszerzeniem pliku wyjściowego. W takim przypadku nazwa pliku wyjściowego pochodzi od nazwy pliku wejściowego. Zauważ, że nazwy plików nie mogą zaczynać się od myślnika. Wreszcie, jeśli plik_wyjściowy nie ma rozszerzenia, jest traktowany jako folder i do tego folderu jest zapisywany „otwarty e-book” (OEB) składający się z plików HTML. Te pliki to pliki, które normalnie zostałyby przekazane do wtyczki wyjściowej.

Po określeniu pliku wejściowego i wyjściowego możesz dostosować konwersję, określając różne opcje. Dostępne opcje zależą od typów plików wejściowych i wyjściowych. Aby uzyskać pomoc, podaj plik wejściowy i wyjściowy, a następnie użyj opcji -h.

Pełna dokumentacja systemu konwersji zobacz [Konwersja książek](#) (strona 59)

Ilekoć przekazujesz argumenty do **ebook-convert**, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „/some path/with spaces”

Opcje i domyślne wartości są różne w zależności od formatu wejściowego i wyjściowego, zawsze więc należy je sprawdzić:

```
ebook-convert myfile.input_format myfile.output_format -h
```

Poniżej znajdują się opcje wspólne dla wszystkich formatów, a następnie te, które zależą od formatu wejścia i wyjścia.

- *OPCJE PLIKU ŹRÓDŁOWEGO* (strona 328)
- *OPCJE PLIKU WYJŚCIOWEGO* (strona 328)
- *WYGLĄD* (strona 329)
- *PRZETWARZANIE HEURYSTYCZNE* (strona 332)
- *WYSZUKAJ I ZASTĄP* (strona 332)
- *WYKRYWANIE STRUKTURY* (strona 333)
- *SPIS TREŚCI* (strona 334)
- *METADANE* (strona 334)
- *DEBUG* (strona 335)

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--input-profile

Określ profil wejścia. Profil wejścia dyktuje systemowi konwersji jak interpretować różne informacje w dokumencie wejściowym. Na przykład długości zależne od rozdzielczości (tj. długość w pikselach). Do wyboru jest: cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, irexdr1000, irexdr800, kindle, msreader, mobipocket, nook, sony, sony300, sony900

--list-recipes

Lista wbudowanych źródeł. Korzystając ze źródła możesz utworzyć e-booka za pomocą polecenia ebook-convert "Nazwa źródła.recipe" wynik.epub

--output-profile

Określ profil wyjściowy. Profil wyjściowy stanowi informację dla systemu konwersji jak zoptymalizować tworzony dokument dla wybranego urządzenia. W niektórych przypadkach profil wyjściowy może posłużyć do optymalizacji dla konkretnego urządzenia, ale to jest rzadko wymagane. Do wyboru jest: cybookg3, cybook_opus, default, hanlinv3, hanlinv5, illiad, ipad, ipad3, irexdr1000, irexdr800, jetbook5, kindle, kindle_dx, kindle_fire, kindle_oasis, kindle_pw, kindle_pw3, kindle_scribe, kindle_voyage, kobo, msreader, mobipocket, nook, nook_color, nook_hd_plus, generic_eink, generic_eink_hd, generic_eink_large, pocketbook_inkpad3, pocketbook_lux, pocketbook_hd, pocketbook_900, pocketbook_pro_912, galaxy, sony, sony300, sony900, sony-landscape, sonyt3, tablet

--version

wyświetla wersję programu i kończy działanie

OPCJE PLIKU ŹRÓDŁOWEGO

Opcje sterujące przetwarzaniem pliku źródłowego mobi

--input-encoding

Określ kodowanie znaków dokumentu wejściowego. Jeśli ta opcja zostanie wybrana, wybrane kodowanie zastąpi to zadeklarowane w dokumencie. Szczególnie przydatne jest to w przypadku dokumentów, które nie deklarują kodowania lub które mają błędną deklarację.

OPCJE PLIKU WYJŚCIOWEGO

Opcje sterujące przetwarzaniem pliku wyjściowego epub

--dont-split-on-page-breaks

Wyłącz dzielenie przy znaku podziału strony. Zwykle pliki wejściowe są automatycznie dzielone przy każdym znaku podziału strony na dwa pliki. Dzięki temu książka wyjściowa może być szybciej i wydajniej przetworzona. Jednakże, podział jest powolny i jeśli plik źródłowy zawiera dużą ilość znaków podziału strony, powinieneś wyłączyć podział przy znaku podziału strony.

--epub-flatten

Ta opcja jest wymagana tylko wtedy, gdy zamierzasz wykorzystywać EPUBa z czytnikiem FBReaderJ. To spłaszczy system plików wewnątrz EPUBa, umieszczając wszystkie pliki na najwyższym poziomie.

--epub-inline-toc

Wstaw spis treści, który będzie stanowił część książki.

--epub-max-image-size

Maksymalny rozmiar obrazu (szerokość x wysokość). Wartość none oznacza użycie rozmiaru ekranu z profilu wyjściowego. Wartość profile oznacza, że nie określono maksymalnego rozmiaru. Na przykład wartość 100x200 spowoduje, że rozmiar wszystkich obrazów zostanie zmieniony tak, aby ich szerokość nie przekraczała 100 pikseli, a wysokość nie przekraczała 200 pikseli. Należy pamiętać, że wpływa to tylko na rozmiar samych plików graficznych. Dowolny obraz może być renderowany w innym rozmiarze w zależności od stylizacji zastosowanej do niego w dokumencie.

--epub-toc-at-end

Umieść wstawiany spis treści na końcu książki zamiast na początku.

--epub-version

Wersja pliku EPUB, która zostanie wygenerowana. EPUB 2 jest najbardziej kompatybilny, skorzystaj z EPU" B 3 jeśli tylko wiesz, że takiej wersji potrzebujesz.

--extract-to

Wyodrębnij zawartość wygenerowanego pliku EPUB do określonego folderu. Zawartość folderu jest najpierw usuwana, więc bądź ostrożny.

--flow-size

Dzielenie wszystkich plików HTML większych niż ten rozmiar (w KB). Jest to konieczne, gdyż większość czytników EPUB nie potrafi obsługiwać plików o dużych rozmiarach. Domyślna wartość 260 KB to rozmiar wymagany przez Adobe Digital Editions. Ustaw na 0, aby wyłączyć dzielenie w oparciu o rozmiar.

--no-default-epub-cover

Zazwyczaj, jeżeli plik źródłowy nie ma okładki i nie wskażesz odpowiedniej, jest generowana domyślna okładka zawierająca tytuł, autora itp. Ta opcja wyłącza tworzenie takiej okładki.

--no-svg-cover

Nie używaj SVG do okładki książki. Użyj tej opcji, jeśli twój EPUB będzie odczytywany na urządzeniu, które nie wspiera SVG, jak iPhone czy JetBook Lite. Bez tej opcji takie urządzenia będą wyświetlać okładkę jako pustą stronę.

--preserve-cover-aspect-ratio

Jeśli okładka jest w postaci grafiki SVG, włączenie tej opcji spowoduje zmianę rozmiaru okładki na cały dostępny obszar ekranu, zachowując jednak proporcje obrazu (stosunek jego szerokości do wysokości). Oznacza to, że na brzegach obrazu mogą pojawić się białe pasy, nie zostanie on jednak zniekształcony. Wyłączenie tej opcji może spowodować lekkie zniekształcenie obrazu, nie pojawią się natomiast białe pasy na brzegach.

--pretty-print

Jeśli jest określony, wtyczka spróbuje wygenerować jak najbardziej czytelne dane. Dla niektórych wtyczek generujących dane wyjściowe może nie dawać to żadnych efektów.

--toc-title

Tytuł dla automatycznie wygenerowanego spisu treści.

WYGLĄD

Opcje umożliwiające kontrolę nad wyglądem pliku wyjściowego

--asciiize

Transliteruj znaki Unicode do reprezentacji ASCII. Używaj ostrożnie, ponieważ spowoduje to zamianę znaków Unicode na ASCII. Na przykład zamieni „Pelé” na „Pele”. Należy również pamiętać, że w przypadkach, gdy istnieje wiele reprezentacji znaku (znaki wspólne na przykład w języku chińskim i japońskim), zostanie użyta reprezentacja oparta na bieżącym języku interfejsu calibre.

--base-font-size

Podstawowy rozmiar czcionki w pkt. Wszystkie rozmiary czcionek w wyprodukowanej książce zostaną przeskalowane na podstawie tego rozmiaru. Wybierając większy rozmiar, możesz zwiększyć czcionki w wydruku i odwrotnie. Domyślnie, gdy wartość wynosi zero, podstawowy rozmiar czcionki jest wybierany na podstawie wybranego profilu wyjściowego.

--change-justification

Zmień justowanie. Wartość "lewy" wyrówna cały tekst do lewej. Wartość "wyjustuj" dokona wyjustowania całego tekstu. Wartość "oryginalny" (domyślna) nie zmieni justowania w pliku źródłowym. Weź pod uwagę fakt, że tylko niektóre formaty obsługują justowanie.

--disable-font-rescaling

Nie skaluj rozmiaru czcionek.

--embed-all-fonts

Osadź wszystkie czcionki, które zostały użyte w dokumencie wejściowym, ale nie zostały jeszcze w nim osadzone. Spowoduje to przeszukanie systemu w poszukiwaniu potrzebnych czcionek, a jeśli zostaną one znalezione, osadzenie ich w e-booku. Działa to wyłącznie w przypadku formatów, które obsługują osadzanie czcionek takich jak EPUB, AZW3, DOCX lub PDF. Upewnij się, że masz licencję, jeśli nie używasz darmowych czcionek.

--embed-font-family

Osadź wybraną czcionkę w pliku książki. Czcionka jest traktowana jako "bazowa" dla książki. Jeśli dokument wejściowy używa osobnej czcionki, jego ustawienia mogą nadpisać tę czcionkę bazową. Można użyć filtrów stylów aby usunąć czcionkę z dokumentu wejściowego. Należy pamiętać, że osadzanie czcionek działa tylko w niektórych formatach, głównie EPUB, AZW3 i DOCX.

--expand-css

Domyślnie calibre używa skróconych form właściwości CSS, takich jak margin, padding, border itp. Ta opcja spowoduje, że zostaną użyte pełne formy zamiast skróconych. Pełne wersje są zawsze używane przy generowaniu EPUBów przy wybranym jednym z profili wyjściowych Nook ponieważ Nook nie obsługuje skróconych form CSS.

--extra-css

Ścieżka do pliku stylów CSS lub sam CSS. Plik CSS zostanie dodany to stylów z pliku źródłowego, może więc zostać użyty do nadpisania tych zasad.

--filter-css

Oddzielana przecinkami lista właściwości CSS, które będą usunięte ze wszystkich arkuszy stylów. Jest to użyteczne jeśli jakieś formatowanie koliduje z ustawieniami na czytniku. Przykładem może być font-family, color, margin-left, margin-right

--font-size-mapping

Mapowanie z rozmiarów czcionek w CSS na rozmiar w punktach. Przykładowe ustawienia to: 10,12,14,16,18,20,22,24. Są to mapowania dla rozmiarów od xx-mały do xx-duży, gdzie ostatni rozmiar jest bardzo duży. Algorytm przekalowywania używa tych rozmiarów, aby inteligentnie zmieniać wielkość czcionki. Domyślnie używane jest mapowanie bazujące na wybranym profilu wyjściowym.

--insert-blank-line

Wstaw pusty wiersz pomiędzy akapitami. Ustawienie nie będzie działać, jeśli plik źródłowy nie używa akapitów (znaczników <p> lub <div>).

--insert-blank-line-size

Podaj wysokość pustych wierszy (w em). Puste wiersze między akapitami będą dwukrotnie większe niż ustawiona tu wartość.

--keep-ligatures

Zachowuj ligatury istniejące w dokumencie. Ligatura to szczególna para znaków, taka jak ff, fi, fl i inne. Większość domyślnych czcionek w czytnikach nie zawiera ligatur, więc ich prawidłowe wyświetlanie jest mało prawdopodobne. Domyślnie calibre zamienia ligaturę na odpowiadające jej standardowe znaki. Po włączeniu tej opcji ligatury będą zachowywane.

--line-height

Wysokość wiersza w punktach. Służy do ustawienia odstępu między sąsiednimi wierszami. Jest stosowana tylko w tych elementach, które nie mają określonej własnej wysokości wiersza. W większości przypadków bardziej użyteczna jest opcja "minimalna wysokość wiersza". Domyślnie wysokość wiersza nie jest zmieniana.

--linearize-tables

Niektóre źle zaprojektowane dokumenty używają tabel do rozmieszczenia tekstu na stronie. Często po konwersji w takich dokumentach pojawia się tekst wychodzący poza stronę i inne błędy. Ta opcja wydobędzie tekst z tabel i przedstawi go w sposób ciągły.

--margin-bottom

Ustaw dolny margines w pkt. Domyślnie jest %d to ustawienie domyślne. Ustawienie wartości mniejszej niż zero spowoduje, że margines nie zostanie ustawiony (ustawienie marginesu w oryginalnym dokumencie zostanie zachowane). Uwaga: formaty zorientowane na strony, takie jak PDF i DOCX, mają własne ustawienia marginesów, które mają pierwszeństwo.

--margin-left

Ustaw lewy margines w pkt. Domyślnie jest %d to ustawienie domyślne. Ustawienie wartości mniejszej niż zero spowoduje, że margines nie zostanie ustawiony (ustawienie marginesu w oryginalnym dokumencie zostanie zachowane). Uwaga: formaty zorientowane na strony, takie jak PDF i DOCX, mają własne ustawienia marginesów, które mają pierwszeństwo.

--margin-right

Ustaw prawy margines w pkt. Domyślnie jest %d to ustawienie domyślne. Ustawienie wartości mniejszej niż zero spowoduje, że margines nie zostanie ustawiony (ustawienie marginesu w oryginalnym dokumencie zostanie zachowane). Uwaga: formaty zorientowane na strony, takie jak PDF i DOCX, mają własne ustawienia marginesów, które mają pierwszeństwo.

--margin-top

Ustaw górny margines w pkt. Domyślnie jest %d to ustawienie domyślne. Ustawienie wartości mniejszej niż zero spowoduje, że margines nie zostanie ustawiony (ustawienie marginesu w oryginalnym dokumencie zostanie zachowane). Uwaga: formaty zorientowane na strony, takie jak PDF i DOCX, mają własne ustawienia marginesów, które mają pierwszeństwo.

--minimum-line-height

Minimalna wysokość wiersza, zależna od rozmiaru czcionki wyliczonego dla elementu. calibre będzie pilnować, aby każdy element miał wysokość wiersza nie mniejszą niż ustawiona wartość, niezależnie od tego, co określa dokument wejściowy. Ustaw tę wartość na zero, aby wyłączyć. Domyślnie wynosi ona 120%. Użyj tego ustawienia, zamiast bezpośredniego ustawienia wysokości wiersza, chyba że wiesz co robisz. Na przykład, można uzyskać tekst o „podwójnym odstępie między wierszami” poprzez ustawienie wartości na 240.

--remove-paragraph-spacing

Usuwa odstęp pomiędzy akapitami. Ustawia również wcięcie akapitu w wielkości 1.5em. Usuwanie odstępu nie zadziała, jeśli plik wejściowy nie używa akapitów (znaczników <p> lub <div>).

--remove-paragraph-spacing-indent-size

Kiedy calibre usuwa puste wiersze między akapitami, automatycznie dodaje wcięcia akapitowe, by umożliwić rozpoznanie struktury tekstu. Ta opcja określa wielkość wcięcia akapitowego (w em). Ustawienie wartości ujemnej spowoduje, że zostanie użyta wartość ustawiona w dokumencie źródłowym, czyli praktycznie wcięcie nie jest zmieniane.

--smarten-punctuation

Konwertuj zwykłe cudzysłowy, myślniki i wielokropki na ich poprawne typograficznie odpowiedniki. Aby uzyskać szczegółowe informacje, zobacz <https://daringfireball.net/projects/smartyants>.

--subset-embedded-fonts

Zredukuj osadzone czcionki. Każda osadzona czcionka zostanie zredukowana tak, aby zawierała tylko znaki wykorzystywane w tym dokumencie. Zmniejszy to rozmiar plików z czcionkami. Przydatne przy korzystaniu z czcionek zawierających wiele niewykorzystywanych znaków.

--transform-css-rules

Ścieżka do pliku zawierającego reguły przetwarzania stylów CSS w tej książce. Najprostszym sposobem, by stworzyć taki plik, jest użycie kreatora reguł w interfejsie calibre. Dostań się do niego przez sekcję „Look & Fell → Transform styles” dialogu konwersji. Kiedy stworzysz reguły, możesz użyć przycisku „Eksport”, aby zapisać je do pliku.

--transform-html-rules

Ścieżka do pliku zawierającego reguły przekształcania kodu HTML w tej książce. Najłatwiejszym sposobem utworzenia takiego pliku jest skorzystanie z kreatora tworzenia reguł w GUI calibre. Uzyskaj do niego dostęp w sekcji „Wygląd i styl->Przekształć kod HTML” w oknie dialogowym konwersji. Po utworzeniu reguł możesz użyć przycisku „Eksportuj”, aby zapisać je do pliku.

--unsmarten-punctuation

Przekształć typograficzne cudzysłowy, myślniki i wielokropki na zwykłe odpowiedniki.

PRZETWARZANIE HEURYSTYCZNE

Zmodyfikuj tekst i strukturę używając wzorców. Domyślnie wyłączone. Aby włączyć użyj `--enable-heuristics`. Poszczególne akcje mogą zostać wyłączone przy użyciu `--disable-*`.

--disable-dehyphenate

Przeanalizuj podzielone słowa w całym dokumencie. Dokument jest używany jako słownik do ustalenia czy łączniki powinny być zachowane, czy usunięte.

--disable-delete-blank-paragraphs

Usuń z dokumentu puste akapity, gdy występują one pomiędzy co drugim akapitem

--disable-fix-indents

Zamień wcięcia składające się z wielokrotnych nierozdzielających spacji na wcięcia CSS.

--disable-format-scene-breaks

Podziały scen są wycentrowane. Zamień podziały scen, które wykorzystują wiele wierszy na linie poziome.

--disable-italicize-common-cases

Sprawdź najczęściej używane wyrazy i wzorce, które są oznaczane kursywą i pochyl je.

--disable-markup-chapter-headings

Wykryj niesformatowane tytuły rozdziałów i podrozdziałów. Zamień je na znaczniki h2 i h3. To ustawienie nie wygeneruje spisu treści, ale razem z detekcją struktury może być użyte do jego stworzenia.

--disable-renumber-headings

Wyszukuje wystąpienia kolejnych znaczników <h1> lub <h2>. Znaczniki są ponownie numerowane, aby zapobiec podziałowi w środku nagłówka rozdziału.

--disable-unwrap-lines

Usuwać podziały wierszy na podstawie użytej interpunkcji i formatowania.

--enable-heuristics

Włącz przetwarzanie heurystyczne. Ta opcja musi być włączona, aby nastąpiło jakiekolwiek przetwarzanie heurystyczne.

--html-unwrap-factor

Współczynnik wykorzystywany do określenia długości, przy której wiersz powinien pozostać nieprzełamany. Wartość powinna być z przedziału 0 do 1. Domyślna wartość to 0.4, poniżej połowy długości wiersza. W przypadku, gdy w dokumencie tylko kilka wierszy ma pozostać nieprzełamanych ta wartość powinna zostać zmniejszona

--replace-scene-breaks

Zamień zmiany sceny na określony tekst. Domyślnie wykorzystany jest tekst pochodzący z dokumentu wejściowego.

WYSZUKAJ I ZASTĄP

Modyfikuj tekst dokumentu i jego strukturę, używając wzorów zdefiniowanych przez użytkownika.

--search-replace

Ścieżka do pliku zawierającego wyrażenia regularne wyszukiwania i zamiany. Plik musi zawierać naprzemiennie wiersze wzorca wyszukiwania i zamiany (wiersz zamiany może być pusty). Wyrażenie musi być poprawnym wyrażeniem regularnym Pythona, a plik musi być kodowany w UTF-8.

--sr1-replace

Tekst, który zastąpi ciąg znaleziony za pomocą sr1-search.

--sr1-search

Wzorzec (wyrażenie regularne), który ma zostać zastąpiony przez sr1-replace.

--sr2-replace

Tekst, który zastąpi ciąg znaleziony za pomocą sr2-search.

--sr2-search

Wzorzec (wyrażenie regularne), który ma zostać zastąpiony przez sr2-replace.

--sr3-replace

Tekst, który zastąpi ciąg znaleziony za pomocą sr3-search.

--sr3-search

Wzorzec (wyrażenie regularne), który ma zostać zastąpiony przez sr3-replace.

WYKRYWANIE STRUKTURY

Kontrola autodetekcji struktury dokumentu.

--chapter

Wyrażenie XPath do wykrywania tytułów rozdziałów. Domyślnie znaczniki <h1> lub <h2> zawierające słowa „rozdział”, „książka”, „sekcja”, „prolog”, „epilog” lub „część” są traktowane jako tytuły rozdziałów, jak również wszelkie znaczniki, które mają klasa = „rozdział”. Użyte wyrażenie musi być oceniane jako lista elementów. Aby wyłączyć wykrywanie rozdziałów, użyj wyrażenia „/”. Więcej informacji na temat korzystania z tej funkcji można znaleźć w samouczku XPath w podręczniku użytkownika calibre.

--chapter-mark

Określ jak zaznaczać wykryte rozdziały. Wartość "pagebreak" dzieli stronę między rozdziałami. Wartość "linia" wstawia linię przed rozdziałem. Wartość "brak" wyłącza zaznaczanie rozdziałów, a "wszystko" włącza linie i podział strony jednocześnie.

--disable-remove-fake-margins

Niektóre dokumenty określają marginesy strony poprzez określenie lewego i prawego marginesu dla każdego akapitu indywidualnie. calibre spróbuje wykryć i usunąć te marginesy. Czasami może to spowodować usunięcie marginesów, które nie powinny zostać usunięte. W takim przypadku możesz wyłączyć usuwanie.

--insert-metadata

Wstaw metadane na początku książki. Jest to przydatne, jeśli czytnik nie obsługuje bezpośrednio wyświetlania/wyszukiwania metadanych.

--page-breaks-before

Wyrażenie XPath. Przełamanie stron zostanie wprowadzone przed odnalezionymi elementami. Aby wyłączyć tę funkcję podaj: /

--prefer-metadata-cover

Preferuj okładkę z pliku źródłowego w stosunku do wybranej okładki.

--remove-first-image

Usuń pierwszy obrazek ze źródłowej książki. Opcja przydaje się jeśli książka zawiera obrazek okładki, który nie jest identyfikowany jako okładka. W efekcie jeśli dodasz swoją okładkę, w calibre książka będzie miała dwie okładki, jeśli nie ustawisz tej opcji.

--start-reading-at

Wyrażenie XPath do wykrywania lokalizacji w dokumencie, w której należy rozpocząć czytanie. Niektóre programy do czytania e-booków (przede wszystkim Kindle) używają tej lokalizacji jako miejsca otwierania książki. Więcej informacji na temat korzystania z tej funkcji można znaleźć w samouczku XPath w podręczniku użytkownika calibre.

SPIS TREŚCI

Kontroluje automatyczne generowanie spisu treści. Domyślnie, jeśli plik źródłowy ma już spis treści, to ma on pierwszeństwo przed tym wygenerowanym automatycznie.

--duplicate-links-in-toc

Umożliwia umieszczenie powtarzających się pozycji przy tworzeniu spisu treści z odsyłaczy w dokumencie na przykład kilka razy tego samego słowa, odsyłającego w różne miejsca.

--level1-toc

Wyrażenie XPath określające znaczniki, które powinny zostać dodane do spisu treści na pierwszym poziomie. Jeśli wyrażenie zostanie dodane, ma pierwszeństwo przed innymi sposobami autodetekcji. Więcej informacji na temat XPath znajduje się w sekcji XPath Tutorial, w podręczniku użytkownika calibre.

--level2-toc

Wyrażenie XPath określające znaczniki, które powinny zostać dodane do spisu treści na drugim poziomie. Każdy element jest dodawany w hierarchii pod poprzednim elementem pierwszego poziomu. Więcej informacji na temat XPath znajduje się w sekcji XPath Tutorial, w podręczniku użytkownika calibre.

--level3-toc

Wyrażenie XPath określające znaczniki, które powinny zostać dodane do spisu treści na trzecim poziomie. Każdy element jest dodawany w hierarchii pod poprzednim elementem drugiego poziomu. Więcej informacji na temat XPath znajduje się w sekcji XPath Tutorial, w podręczniku użytkownika calibre.

--max-toc-links

Maksymalna liczba odnośników do wstawienia do spisu treści. Ustaw tę wartość na 0, aby wyłączyć. Domyślnie ustawienie: 50. Odnośniki są dodawane do spisu treści tylko jeśli odnaleziono mniej rozdziałów niż określony próg.

--no-chapters-in-toc

Nie dodawaj automatycznie wykrytych rozdziałów do spisu treści.

--toc-filter

Usuwa te pozycje ze spisu treści, których tytuły pasują do określonego wyrażenia. Zostaną usunięte pasujące i wszystkie ich podrzędne pozycje.

--toc-threshold

Jeśli zostanie wykryte mniej rozdziałów niż podano, odnośniki są dodawane do spisu treści. Domyślnie: 6

--use-auto-toc

Zazwyczaj, jeżeli plik źródłowy ma już spis treści, jest on wykorzystywany zamiast wygenerowanego automatycznie. Z tą opcją zawsze używany jest wygenerowany automatycznie spis treści.

METADANE

Opcje odpowiedzialne za ustawienia metadanych w plikach wyjściowych

--author-sort

Łańcuch znaków używany przy sortowaniu po autorze.

--authors

Wpisz autorów. Jeśli jest więcej niż jeden, należy oddzielić ich znakami ampersand (" & ").

--book-producer

Podaj producenta książki (w Polsce nie praktykowane).

--comments

Ustaw opis dla e-booka.

--cover

Ustaw okładkę wybierając plik lub podając adres URL

--isbn

Wpisz numer ISBN książki.

--language

Podaj język.

--pubdate

Ustaw datę publikacji (zakłada się, że znajduje się w lokalnej strefie czasowej, chyba że strefa czasowa jest wyraźnie określona)

--publisher

Ustaw wydawcę e-booka.

--rating

Wpisz ocenę. Musi to być cyfra pomiędzy 1 a 5.

--read-metadata-from-opf, --from-opf, -m

Odczytaj metadane z wybranego pliku OPF. Metadane odczytane z tego pliku nadpiszą wszystkie metadane pliku źródłowego.

--series

Wybierz serie, do których należy ta książka.

--series-index

Podaj numer książki w cyklu.

--tags

Wpisz etykiety dla książki. Kolejne etykiety oddzielaj przecinkami.

--timestamp

Ustaw stempel czasowy książki (nie używane nigdzie indziej)

--title

Wpisz tytuł.

--title-sort

Wersja tytułu używana przy sortowaniu.

DEBUG

Opcje pomocy przy debugowaniu konwersji

--debug-pipeline, -d

Zapisz dane wyjściowe z różnych etapów potoku konwersji w określonym folderze. Przydatne, jeśli nie masz pewności, na którym etapie procesu konwersji występuje błąd.

--verbose, -v

Poziom komunikatów. Im więcej razy jest podany, tym komunikaty będą obszerniejsze. Podanie parametru dwa razy to maksymalna ilość informacji, raz - normalna, a przy braku parametru komunikaty będą zawierały minimum informacji.

13.1.8 ebook-edit

```
ebook-edit [opts] [path_to_ebook] [name_of_file_inside_book ...]
```

Launch the calibre Edit book tool. You can optionally also specify the names of files inside the book which will be opened for editing automatically.

Ilekoć przekazujesz argumenty do **ebook-edit**, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

[opcje]

--detach

Odłącz od terminala sterującego, jeśli istnieje (tylko Linux)

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--select-text

Tekst do zaznaczenia w książce, gdy jest ona otwarta do edycji

--version

wyświetla wersję programu i kończy działanie

13.1.9 ebook-meta

```
ebook-meta ebook_file [options]
```

Czytanie/zapis metadanych plików książki

Obsługiwane formaty odczytu metadanych: azw, azw1, azw3, azw4, cb7, cbc, cbr, cbz, chm, docx, epub, fb2, fbz, html, htmlz, imp, lit, lrf, lrx, mobi, odt, oebzip, opf, pdb, pdf, pml, pmlz, pobi, prc, rar, rb, rtf, snb, tpz, txt, txtz, updb, zip

Obsługiwane formaty zapisu metadanych: azw, azw1, azw3, azw4, docx, epub, fb2, fbz, htmlz, lrf, mobi, odt, pdb, pdf, prc, rtf, tpz, txtz

Różne typy plików obsługują różne typy metadanych. Jeśli zostaną wprowadzone nieobsługiwane dane, zostaną one pominięte.

Ilekoć przekazujesz argumenty do **ebook-meta**, które zawierają spacje, ujmuj je w cudzysłowy. Na przykład: „/some path/with spaces”

[opcje]

--author-sort

Ciąg znaków używany podczas sortowania według autora. Jeśli nie zostanie określony, a autor(rzy) nie będą wskazani, zostanie automatycznie wygenerowany od autora(ów).

--authors, -a

Wpisz autorów. Jeśli występuje kilku autorów oddzielaj ich za pomocą znaku "&". Dane autorów powinny być podane w kolejności: Imię Nazwisko.

--book-producer, -k

Podaj producenta książki (w Polsce nie praktykowane).

--category

Wpisz kategorię książki.

--comments, -c

Ustaw opis dla e-booka.

--cover

Ustaw podany plik jako okładkę.

--date, -d

Podaj datę publikacji.

--from-opf

Odczytaj metadane z określonego pliku OPF i użyj go do ustawienia metadanych w e-booku. Metadane określone w wierszu polecenia zastąpią metadane odczytane z pliku OPF

--get-cover

Pobierz okładkę z książki i zapisz ją do określonego pliku.

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--identifier

Ustaw identyfikatory książki, można je podawać wielokrotnie. Na przykład: `--identifier` (strona 337) `uri:https://acme.com` `--identifier` (strona 337) `isbn:12345` Aby usunąć identyfikator, nie określaj żadnej wartości, `--identifier` (strona 337) `isbn:` Zauważ, że w przypadku plików EPUB identyfikator oznaczony jako identyfikator pakietu nie może być oddalony.

--index, -i

Podaj numer książki w cyklu.

--isbn

Wpisz numer ISBN książki.

--language, -l

Podaj język.

--lrf-bookid

Ustaw identyfikator książki w plikach LFR

--publisher, -p

Ustaw wydawcę e-booka.

--rating, -r

Wpisz ocenę. Musi to być cyfra pomiędzy 1 a 5.

--series, -s

Wybierz serie, do których należy ta książka.

--tags

Wpisz etykiety dla książki. Kolejne etykiety oddzielaj przecinkami.

--title, -t

Wpisz tytuł.

--title-sort

Forma tytułu używana przy sortowaniu. Jeśli nie jest określona, a tytuł jest podany, zostanie ona automatycznie wygenerowana z tytułu.

--to-opf

Określ nazwę dla pliku OPF. Metadane zostaną zapisane w tym pliku.

--version

wyświetla wersję programu i kończy działanie

13.1.10 ebook-polish

```
ebook-polish [options] input_file [output_file]
```

Dopracowywanie książek jest dodawaniem ostatniego szlif do twoich książek.

Dopracowywanie powoduje jedynie niewielkie zmiany w kodzie książki. W odróżnieniu od konwersji w tym przypadku nie są zmieniane arkusze CSS, nazwy plików, rozmiar czcionki, marginesy itp. Na każdym etapie wykonywane są jedynie minimalne zmiany, potrzebne do uzyskania pożądanego efektu.

Tego narzędzia należy użyć na ostatnim etapie tworzenia książki

Należy pamiętać, że dopracować można jedynie książki w formatach AZW3 lub EPUB

Ileokroć przekazujesz argumenty do **ebook-polish**, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „./some path/with spaces”

[opcje]

--add-soft-hyphens, -H

Dodaje łączniki opcjonalne do wszystkich słów w książce. Pozwala to, bo książka lepiej wyglądała przy wyjustowanym tekście, gdy czytnik nie obsługuje automatycznego dzielenie wyrazów.

--compress-images, -i

Bezstratna kompresja obrazów w książce, aby zmniejszyć rozmiar pliku bez wpływu na jakość obrazu.

--cover, -c

Ścieżka do zdjęcia na okładkę. Zmienia okładkę określoną w e-booku. Jeśli nie ma okładki lub okładka nie jest właściwie zidentyfikowana, wstawia nową.

--embed-fonts, -e

Osadź wszystkie czcionki, które zostały użyte w dokumencie wejściowym, ale nie zostały jeszcze w nim osadzone. Spowoduje to przeszukanie systemu w poszukiwaniu czcionek, a jeśli zostaną one znalezione, osadzenie ich w e-booku. Upewnij się, że masz licencję jeśli nie używasz darmowych czcionek.

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--jacket, -j

Wstaw na początku książki "metryczkę", która zawiera wszystkie informacje o książce takie jak tytuł, etykiety, autorów, serię, komentarze itp. Jeśli wcześniej wstawiono metryczkę zostanie ona zamieniona na nową.

--opf, -o

Ścieżka do pliku OPF. Metadane książki są uaktualniane na podstawie pliku OPF.

--remove-jacket

Usuń poprzednią metryczkę

--remove-soft-hyphens

Usuwa łączniki opcjonalne z wszystkich tekstów w książce.

--remove-unused-css, -u

Usuń wszystkie nieużywane reguły CSS z arkuszy stylów i <style> tagi. Niektóre książki utworzone z szablonów produkcyjnych mogą mieć dużą liczbę dodatkowych reguł CSS które nie pasują do żadnej rzeczywistej treści. Te dodatkowe zasady mogą spowolnić czytelników które muszą je wszystkie przeanalizować.

--smarten-punctuation, -p

Konwertuj myślniki, cudzysłowy, wielokrotne łączniki itp. na ich poprawne typograficzne odpowiedniki. Należy pamiętać, że algorytm może czasami generować błędne rezultaty, szczególnie przy apostrofach na początku skrótów w języku angielskim

--subset-fonts, -f

Podzbiór znaków jest tworzony aby zredukować osadzoną czcionkę jedynie do znaków użytych w książce. Może to znacznie ograniczyć wielkość pliku czcionki (zazwyczaj do około połowy) Jeśli na przykład w książce użyto specjalnej czcionki dla tytułów podzbiór będzie zawierać wyłącznie znaki użyte w tytułach. Albo jeżeli w książce jest osadzona czcionka w odmianie zwykłej i pogrubionej oraz pochylonej ale pogrubienie i pochylenie używane jest rzadko albo nie używane zupełnie to czcionki w odmianie pogrubionej i pochylonej mogą zostać ograniczone do zaledwie kilku znaków albo usunięte całkowicie. Jedyną niedogodnością takiego rozwiązania jest to, że jeśli później zdecydujesz się dodać dodatkowy tekst do książki to może się okazać, że brakuje niektórych znaków

--upgrade-book, -U

Aktualizuj wewnętrzną strukturę książki, jeśli to możliwe. Na przykład aktualizuje książki EPUB 2 do EPUB 3.

--verbose

Wyświetl więcej informacji, przydatne przy debugowaniu.

--version

wyświetla wersję programu i kończy działanie

13.1.11 ebook-viewer

```
ebook-viewer [options] file
```

Zobacz książkę.

Ilećkolwiek przekazujesz argumenty do **ebook-viewer**, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „./some path/with spaces”

[opcje]**--continue**

Kontynuuj czytanie ostatnio otwartej książki

--detach

Odłącz od terminala sterującego, jeśli istnieje (tylko Linux)

--force-reload

Wymuś ponowne załadowanie wszystkich otwartych książek

--full-screen, --fullscreen, -f

Jeśli określono, okno przeglądarki e-booków będzie próbowało otworzyć pełny ekran po uruchomieniu.

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--new-instance

Otwiera nowe okno przeglądarki, nawet jeśli ustawiona jest opcja używania tylko jednego okna przeglądarki.

--open-at

Pozycja, w której ma zostać otwarta określona książka. Pozycja to lokalizacja lub pozycja, którą można uzyskać za pomocą czynności Idź do->Lokalizacja w kontrolkach przeglądarki. Alternatywnie możesz użyć formularza toc:coś, a otworzy się on w miejscu, w którym znajduje się pierwsza pozycja Spisu treści zawierająca ciąg „coś”. Formularz toc:href:something będzie pasował do href (miejsca docelowego łączy wewnętrznego) węzłów toc. Dopasowanie jest dokładne. Jeśli chcesz dopasować podłańcuch, użyj formularza toc:href-contains:coś. Formularz ref:coś będzie używał odwołań do trybu Odniesienia. Formularz search:coś będzie szukał czegoś po otwarciu książki. Forma regex:coś będzie szukać wyrażenia regularnego coś po otwarciu książki.

--raise-window

Jeśli określono, okno przeglądarki e-booków będzie próbowało wysunąć się z przodu po uruchomieniu.

--version

wyświetla wersję programu i kończy działanie

13.1.12 fetch-ebook-metadata

```
fetch-ebook-metadata [opcje]
```

Pobieranie metadanych ze źródeł internetowych. Musisz podać co najmniej jedną z informacji: tytuł, autora lub ISBN.

Ileokroć przekazujesz argumenty do **fetch-ebook-metadata**, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „/some path/with spaces”

[opcje]

--allowed-plugin, -p

Określ nazwę wtyczki służącej do pobierania metadanych. Domyślnie używane są wszystkie wtyczki. Możesz określić kilka wtyczek jednocześnie. Nazwy wszystkich wtyczek: Google, Google Images, Amazon.com, Edelweiss, Open Library, Big Book Search

--authors, -a

Autor(zy) książki

--cover, -c

Podaj nazwę pliku. Okładka, jeśli jest dostępna, zostanie w tym pliku zapisana. Bez tego parametru okładka nie zostanie pobrana.

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--identifier, -I

Identyfikatory, takie jak ASIN/Goodreads id itp. Można podać wiele razy dla wielu identyfikatorów. Na przykład: *--identifier* (strona 340) asin:B0082BAJA0

--isbn, -i

ISBN książki

--opf, -o

Wygeneruj metadane w formacie OPF zamiast w postaci tekstu.

--timeout, -d

Limit czasu w sekundach. Domyślnie 30

--title, -t

Tytuł książki

--verbose, -v

Prześlij log na konsolę (stderr)

--version

wyświetla wersję programu i kończy działanie

13.1.13 lrf2lrs

```
lrf2lrs book.lrf
```

Konwertuje plik LRF do pliku LRS (w formacie XML UTF-8)

Ileć raz przekazujesz argumenty do **lrf2lrs**, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „./some path/with spaces”

[opcje]**--dont-output-resources**

Nie zapisuj zagnieżdżonych w pliku grafik i czcionek na dysku

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--output, -o

Wyjściowy plik LRS

--verbose

Podaj więcej danych

--version

wyświetla wersję programu i kończy działanie

13.1.14 lrfviewer

```
lrfviewer [opcje] book.lrf
```

Odczytaj książkę book.lrf w formacie LRF

Ileć raz przekazujesz argumenty do **lrfviewer**, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „./some path/with spaces”

[opcje]

--disable-hyphenation

Wyłącz dzielenie wyrazów. Powinno znacząco przyspieszyć przetwarzanie.

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--profile

Profil renderowania modułu LRF

--verbose

Wyświetl więcej informacji o procesie renderowania

--version

wyświetla wersję programu i kończy działanie

--visual-debug

Włącz wizualną pomoc przy debugowaniu silnika renderowania

--white-background

Domyślnie tło jest nie do końca białe, ponieważ wydawało mi się to wygodniejsze dla oczu. Dzięki tej opcji możesz ustawić tło na całkowicie białe.

13.1.15 **lrs2lrf**

```
lrs2lrf [opcje] file.lrs
```

Skompiluj plik LRS do pliku LRF.

Ilekoć przekazujesz argumenty do **lrs2lrf**, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „./some path/with spaces”

[opcje]

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--lrs

Konwersja LRS do LRS, opcja przydatna przy debugowaniu.

--output, -o

Ścieżka do pliku wyjściowego

--verbose

Szczegółowe komunikaty w czasie pracy

--version

wyświetla wersję programu i kończy działanie

13.1.16 web2disk

web2disk URL

Gdzie URL to na przykład <https://google.com>

Ilekoć przekazujesz argumenty do **web2disk**, które zawierają spacje, ujmij je w cudzysłowy. Na przykład: „/some path/with spaces”

[opcje]

--base-dir, -d

Folder bazowy, w którym zapisany jest adres URL. Domyślnie jest %d

--delay

Minimalny odstęp w sekundach pomiędzy kolejnymi pobraniami. Domyślnie to 0 s

--dont-download-stylesheets

Nie pobieraj arkuszy stylów CSS.

--encoding

Kodowanie znaków stron internetowych, które próbujesz pobrać. Domyślnie calibre stara się automatycznie wybrać odpowiednie kodowanie.

--filter-regexp

Każdy odsyłacz, który pasuje do tego wyrażenia regularnego zostanie zignorowany. Ta opcja może być użyta wielokrotnie, w efekcie jeśli odsyłacz pasuje do któregośkolwiek wyrażenia to zostanie zignorowany. Domyślnie nie są ignorowane żadne odsyłacze. Jeśli wyrażenie filtrujące i dopasowujące są użyte to wyrażenie filtrujące jest używane jako pierwsze.

--help, -h

wyświetla ten tekst pomocy i kończy działanie

--match-regexp

Podążaj tylko za odnośnikami które pasują do tego wyrażenia regularnego. Ta opcja może być użyta wielokrotnie, w takim przypadku odnośnik będzie śledzony dopóki pasuje do co najmniej jednego wyrażenia regularnego. Domyślnie wszystkie odnośniki są śledzone.

--max-files, -n

Maksymalna liczba plików do pobrania. Stosowane jedynie do plików z etykiet <a href>. Wartość domyślna: 9223372036854775807

--max-recursions, -r

Maksymalna ilość poziomów do rekursji np. poziom zagnieżdżenia odnośników do śledzenia. Domyślnie 1

--timeout, -t

Limit czasu oczekiwania na odpowiedź serwera. Domyślnie: 10.0 sek

--verbose

Pokazuj szczegółową informację wyjściową. Przydatne przy debugowaniu

--version

wyświetla wersję programu i kończy działanie

13.2 Nieudokumentowane polecenia

- ebook-device
- markdown-calibre

Informacje na temat sposobu użycia nieudokumentowanych poleceń można uzyskać wykonując je w terminalu bez argumentów.

Konfigurowanie środowiska programistycznego dla calibre

calibre is completely open source, licensed under the [GNU GPL v3](https://www.gnu.org/licenses/gpl.html)¹¹¹. This means that you are free to download and modify the program to your heart's content. In this section, you will learn how to get a calibre development environment set up on the operating system of your choice. calibre is written primarily in [Python](https://www.python.org)¹¹² with some C/C++ code for speed and system interfacing. Note that calibre requires at least Python 3.8.

Spis treści

- *Design philosophy* (strona 346)
 - *Code layout* (strona 346)
- *Getting the code* (strona 347)
 - *Submitting your changes to be included* (strona 347)
- *Windows development environment* (strona 348)
- *macOS development environment* (strona 349)
- *Linux development environment* (strona 350)
- *Having separate „normal” and „development” calibre installs on the same computer* (strona 350)
- *Wskazówki debugowania* (strona 351)
 - *Using print statements* (strona 351)
 - *Using an interactive Python interpreter* (strona 351)
 - *Using the Python debugger as a remote debugger* (strona 351)
 - *Using the debugger in your favorite Python IDE* (strona 352)
 - *Executing arbitrary scripts in the calibre Python environment* (strona 352)
- *Używanie calibre w twoich projektach* (strona 353)

¹¹¹ <https://www.gnu.org/licenses/gpl.html>

¹¹² <https://www.python.org>

- *Binary install of calibre* (strona 353)
- *Source install on Linux* (strona 353)
- *API documentation for various parts of calibre* (strona 353)

14.1 Design philosophy

calibre has its roots in the Unix world, which means that its design is highly modular. The modules interact with each other via well defined interfaces. This makes adding new features and fixing bugs in calibre very easy, resulting in a frenetic pace of development. Because of its roots, calibre has a comprehensive command line interface for all its functions, documented in `generated/en/cli-index`.

The modular design of calibre is expressed via `Plugins`. There is a *tutorial* (strona 255) on writing calibre plugins. For example, adding support for a new device to calibre typically involves writing less than a 100 lines of code in the form of a device driver plugin. You can browse the *built-in drivers*¹¹³. Similarly, adding support for new conversion formats involves writing input/output format plugins. Another example of the modular design is the *recipe system* (strona 31) for fetching news. For more examples of plugins designed to add features to calibre, see the *Index of plugins*¹¹⁴.

14.1.1 Code layout

All the calibre Python code is in the `calibre` package. This package contains the following main sub-packages

- `devices` - All the device drivers. Just look through some of the built-in drivers to get an idea for how they work.
 - For details, see: `devices.interface` which defines the interface supported by device drivers and `devices.usbms` which defines a generic driver that connects to a USBMS device. All USBMS based drivers in calibre inherit from it.
- `e-books` - All the e-book conversion/metadata code. A good starting point is `calibre.ebooks.conversion.cli` which is the module powering the **ebook-convert** command. The conversion process is controlled via `conversion.plumber`. The format independent code is all in `ebooks.oeb` and the format dependent code is in `ebooks.format_name`.
 - Metadata reading, writing, and downloading is all in `ebooks.metadata`
 - Conversion happens in a pipeline, for the structure of the pipeline, see *Wstęp* (strona 61). The pipeline consists of an input plugin, various transforms and an output plugin. The code that constructs and drives the pipeline is in `plumber.py`. The pipeline works on a representation of an e-book that is like an unzipped epub, with manifest, spine, toc, guide, html content, etc. The class that manages this representation is `OEBBook` in `ebooks.oeb.base`. The various transformations that are applied to the book during conversions live in `oeb/transforms/*.py`. And the input and output plugins live in `conversion/plugins/*.py`.
 - E-book editing happens using a different container object. It is documented in *Dokumentacja API narzędzi do edytowania książek* (strona 363).
- `db` - The database back-end. See *API documentation for the database interface* (strona 353) for the interface to the calibre library.
- `Content server`: `srv` is the calibre Content server.
- `gui2` - The Graphical User Interface. GUI initialization happens in `gui2.main` and `gui2.ui`. The e-book-viewer is in `gui2.viewer`. The e-book editor is in `gui2.tweak_book`.

¹¹³ <https://github.com/kovidgoyal/calibre/tree/master/src/calibre/devices>

¹¹⁴ <https://www.mobileread.com/forums/showthread.php?p=1362767#post1362767>

If you want to locate the entry points for all the various calibre executables, look at the `entry_points` structure in `linux.py`¹¹⁵.

If you need help understanding the code, post in the [development forum](#)¹¹⁶ and you will most likely get help from one of calibre's many developers.

14.2 Getting the code

You can get the calibre source code in two ways, using a version control system or directly downloading a [tarball](#)¹¹⁷.

calibre uses [Git](#)¹¹⁸, a distributed version control system. Git is available on all the platforms calibre supports. After installing Git, you can get the calibre source code with the command:

```
git clone https://github.com/kovidgoyal/calibre.git
```

On Windows you will need the complete path name, that will be something like `C:\Program Files\Git\git.exe`.

calibre is a very large project with a very long source control history, so the above can take a while (10 mins to an hour depending on your internet speed).

If you want to get the code faster, the source code for the latest release is always available as an [archive](#)¹¹⁹.

Aby zaktualizować gałąź ostatniego kodu, użyj komendy:

```
git pull --no-edit
```

You can also browse the code at [GitHub](#)¹²⁰.

14.2.1 Submitting your changes to be included

If you only plan to make a few small changes, you can make your changes and create a „merge directive” which you can then attach to a ticket in the calibre [bug tracker](#)¹²¹. To do this, make your changes, then run:

```
git commit -am "Comment describing your changes"
git format-patch origin/master --stdout > my-changes
```

This will create a `my-changes` file in the current folder, simply attach that to a ticket on the calibre [bug tracker](#)¹²². Note that this will include *all* the commits you have made. If you only want to send some commits, you have to change `origin/master` above. To send only the last commit, use:

```
git format-patch HEAD~1 --stdout > my-changes
```

To send the last *n* commits, replace *1* with *n*, for example, for the last 3 commits:

```
git format-patch HEAD~3 --stdout > my-changes
```

¹¹⁵ <https://github.com/kovidgoyal/calibre/blob/master/src/calibre/linux.py>

¹¹⁶ <https://www.mobileread.com/forums/forumdisplay.php?f=240>

¹¹⁷ <https://calibre-ebook.com/dist/src>

¹¹⁸ <https://www.git-scm.com/>

¹¹⁹ <https://calibre-ebook.com/dist/src>

¹²⁰ <https://github.com/kovidgoyal/calibre>

¹²¹ <https://bugs.launchpad.net/calibre>

¹²² <https://bugs.launchpad.net/calibre>

Be careful to not include merges when using `HEAD~n`.

If you plan to do a lot of development on calibre, then the best method is to create a [GitHub](#)¹²³ account. Below is a basic guide to setting up your own fork of calibre in a way that will allow you to submit pull requests for inclusion into the main calibre repository:

- Setup git on your machine as described in this article: [Setup Git](#)¹²⁴
- Setup ssh keys for authentication to GitHub, as described here: [Generating SSH keys](#)¹²⁵
- Go to <https://github.com/kovidgoyal/calibre> and click the *Fork* button.
- W terminalu:

```
git clone git@github.com:<username>/calibre.git
git remote add upstream https://github.com/kovidgoyal/calibre.git
```

Replace `<username>` above with your GitHub username. That will get your fork checked out locally.

- You can make changes and commit them whenever you like. When you are ready to have your work merged, do a:

```
git push
```

and go to <https://github.com/<username>/calibre> and click the *Pull Request* button to generate a pull request that can be merged.

- You can update your local copy with code from the main repo at any time by doing:

```
git pull upstream
```

You should also keep an eye on the calibre [development forum](#)¹²⁶. Before making major changes, you should discuss them in the forum or contact Kovid directly (his email address is all over the source code).

14.3 Windows development environment

Informacja: You must also get the calibre source code separately as described above.

Install calibre normally, using the [Windows installer](#)¹²⁷. Then open a Command Prompt and change to the previously checked out calibre code folder. For example:

```
cd C:\Users\kovid\work\calibre
```

calibre is the folder that contains the src and resources sub-folders.

The next step is to set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the src folder. So, following the example above, it would be `C:\Users\kovid\work\calibre\src`. [Here is a short guide](#)¹²⁸ to setting environment variables on Windows.

Once you have set the environment variable, open a new command prompt and check that it was correctly set by using the command:

¹²³ <https://github.com>

¹²⁴ <https://help.github.com/articles/set-up-git>

¹²⁵ <https://help.github.com/articles/generating-ssh-keys>

¹²⁶ <https://www.mobilerread.com/forums/forumdisplay.php?f=240>

¹²⁷ https://calibre-ebook.com/download_windows

¹²⁸ <https://docs.python.org/using/windows.html#excursus-setting-environment-variables>


```
echo %CALIBRE_DEVELOP_FROM%
```

Setting this environment variable means that calibre will now load all its Python code from the specified location.

That's it! You are now ready to start hacking on the calibre code. For example, open the file `src\calibre__init__.py` in your favorite editor and add the line:

```
print("Hello, world!")
```

near the top of the file. Now run the command **calibredb**. The very first line of output should be `Hello, world!`.

You can also setup a calibre development environment inside the free Microsoft Visual Studio, if you like, following the instructions [here](#)¹²⁹.

14.4 macOS development environment

Informacja: You must also get the calibre source code separately as described above.

Install calibre normally using the [provided .dmg](#)¹³⁰. Then open a Terminal and change to the previously checked out calibre code folder, for example:

```
cd /Users/kovid/work/calibre
```

calibre is the folder that contains the `src` and `resources` sub-folders. The calibre command line tools are found inside the calibre app bundle, in `/Applications/calibre.app/Contents/MacOS` you should add this folder to your `PATH` environment variable, if you want to run the command line tools easily.

The next step is to create a bash script that will set the environment variable `CALIBRE_DEVELOP_FROM` to the absolute path of the `src` folder when running calibre in debug mode.

Stwórz zwykły plik tekstowy

```
#!/bin/sh
export CALIBRE_DEVELOP_FROM="/Users/kovid/work/calibre/src"
calibre-debug -g
```

Save this file as `/usr/local/bin/calibre-develop`, then set its permissions so that it can be executed:

```
chmod +x /usr/local/bin/calibre-develop
```

Jeśli to zrobisz, uruchom:

```
calibre-develop
```

You should see some diagnostic information in the Terminal window as calibre starts up, and you should see an asterisk after the version number in the GUI window, indicating that you are running from source.

¹²⁹ <https://www.mobileread.com/forums/showthread.php?t=251201>

¹³⁰ https://calibre-ebook.com/download_osx

14.5 Linux development environment

Informacja: You must also get the calibre source code separately as described above.

calibre is primarily developed on Linux. You have two choices in setting up the development environment. You can install the calibre binary as normal and use that as a runtime environment to do your development. This approach is similar to that used in Windows and macOS. Alternatively, you can install calibre from source. Instructions for setting up a development environment from source are in the INSTALL file in the source tree. Here we will address using the binary as a runtime, which is the recommended method.

Install calibre using the [binary installer](#)¹³¹. Then open a terminal and change to the previously checked out calibre code folder, for example:

```
cd /home/kovid/work/calibre
```

calibre is the folder that contains the src and resources sub-folders.

The next step is to set the environment variable CALIBRE_DEVELOP_FROM to the absolute path of the src folder. So, following the example above, it would be /home/kovid/work/calibre/src. How to set environment variables depends on your Linux distribution and what shell you are using.

Informacja: It is recommended to use the binary installer provided from upstream. Should you insist on using a package provided by your distribution, use the CALIBRE_PYTHON_PATH and CALIBRE_RESOURCES_PATH variables instead.

Once you have set the environment variable, open a new terminal and check that it was correctly set by using the command:

```
echo $CALIBRE_DEVELOP_FROM
```

Setting this environment variable means that calibre will now load all its Python code from the specified location.

That's it! You are now ready to start hacking on the calibre code. For example, open the file src/calibre/__init__.py in your favorite editor and add the line:

```
print("Hello, world!")
```

near the top of the file. Now run the command **calibredb**. The very first line of output should be `Hello, world!`.

14.6 Having separate „normal” and „development” calibre installs on the same computer

The calibre source tree is very stable and rarely breaks, but if you feel the need to run from source on a separate test library and run the released calibre version with your everyday library, you can achieve this easily using .bat files or shell scripts to launch calibre. The example below shows how to do this on Windows using .bat files (the instructions for other platforms are the same, just use a shell script instead of a .bat file)

To launch the release version of calibre with your everyday library:

calibre-normal.bat:

¹³¹ https://calibre-ebook.com/download_linux

```
calibre.exe "--with-library=C:\path\to\everyday\library folder"
```

calibre-dev.bat:

```
set CALIBRE_DEVELOP_FROM=C:\path\to\calibre\checkout\src
calibre.exe "--with-library=C:\path\to\test\library folder"
```

14.7 Wskazówki debugowania

Python is a dynamically typed language with excellent facilities for introspection. Kovid wrote the core calibre code without once using a debugger. There are many strategies to debug calibre code:

14.7.1 Using print statements

This is Kovid's favorite way to debug. Simply insert print statements at points of interest and run your program in the terminal. For example, you can start the GUI from the terminal as:

```
calibre-debug -g
```

Similarly, you can start the E-book viewer as:

```
calibre-debug -w /path/to/file/to/be/viewed
```

The e-book editor can be started as:

```
calibre-debug --edit-book /path/to/be/edited
```

14.7.2 Using an interactive Python interpreter

You can insert the following two lines of code to start an interactive Python session at that point:

```
from calibre import ipython
ipython(locals())
```

When running from the command line, this will start an interactive Python interpreter with access to all locally defined variables (variables in the local scope). The interactive prompt even has Tab completion for object properties and you can use the various Python facilities for introspection, such as `dir()`, `type()`, `repr()`, etc.

14.7.3 Using the Python debugger as a remote debugger

You can use the builtin Python debugger (pdb) as a remote debugger from the command line. First, start the remote debugger at the point in the calibre code you are interested in, like this:

```
from calibre.rpdb import set_trace
set_trace()
```

Then run calibre, either as normal, or using one of the calibre-debug commands described in the previous section. Once the above point in the code is reached, calibre will freeze, waiting for the debugger to connect.

Now open a terminal or command prompt and use the following command to start the debugging session:

```
calibre-debug -c "from calibre.rpdb import cli; cli()"
```

You can read about how to use the Python debugger in the [Python stdlib docs for the pdb module](#)¹³².

Informacja: By default, the remote debugger will try to connect on port 4444. You can change it, by passing the port parameter to both the `set_trace()` and the `cli()` functions above, like this: `set_trace(port=1234)` and `cli(port=1234)`.

Informacja: The Python debugger cannot handle multiple threads, so you have to call `set_trace` once per thread, each time with a different port number.

14.7.4 Using the debugger in your favorite Python IDE

It is possible to use the builtin debugger in your favorite Python IDE, if it supports remote debugging. The first step is to add the calibre src checkout to the `PYTHONPATH` in your IDE. In other words, the folder you set as `CALIBRE_DEVELOP_FROM` above, must also be in the `PYTHONPATH` of your IDE.

Then place the IDE's remote debugger module into the `src` sub-folder of the calibre source code checkout. Add whatever code is needed to launch the remote debugger to calibre at the point of interest, for example in the main function. Then run calibre as normal. Your IDE should now be able to connect to the remote debugger running inside calibre.

14.7.5 Executing arbitrary scripts in the calibre Python environment

The **calibre-debug** command provides a couple of handy switches to execute your own code, with access to the calibre modules:

```
calibre-debug -c "some Python code"
```

is great for testing a little snippet of code on the command line. It works in the same way as the `-c` switch to the Python interpreter:

```
calibre-debug myscript.py
```

can be used to execute your own Python script. It works in the same way as passing the script to the Python interpreter, except that the calibre environment is fully initialized, so you can use all the calibre code in your script. To use command line arguments with your script, use the form:

```
calibre-debug myscript.py -- --option1 arg1
```

The `--` causes all subsequent arguments to be passed to your script.

¹³² <https://docs.python.org/library/pdb.html#debugger-commands>

14.8 Używanie calibre w twoich projektach

It is possible to directly use calibre functions/code in your Python project. Two ways exist to do this:

14.8.1 Binary install of calibre

If you have a binary install of calibre, you can use the Python interpreter bundled with calibre, like this:

```
calibre-debug /path/to/your/python/script.py -- arguments to your script
```

14.8.2 Source install on Linux

In addition to using the above technique, if you do a source install on Linux, you can also directly import calibre, as follows:

```
import init_calibre
import calibre

print(calibre.__version__)
```

It is essential that you import the `init_calibre` module before any other calibre modules/packages as it sets up the interpreter to run calibre code.

14.9 API documentation for various parts of calibre

14.9.1 API documentation for the database interface

This API is thread safe (it uses a multiple reader, single writer locking scheme). You can access this API like this:

```
from calibre.library import db
db = db('Path to calibre library folder').new_api
```

If you are in a calibre plugin that is part of the main calibre GUI, you get access to it like this instead:

```
db = self.gui.current_db.new_api
```

class `calibre.db.cache.Cache` (*backend, library_database_instance=None*)

An in-memory cache of the `metadata.db` file from a calibre library. This class also serves as a threadsafe API for accessing the database. The in-memory cache is maintained in normal form for maximum performance.

SQLITE is simply used as a way to read and write from `metadata.db` robustly. All table reading/sorting/searching/caching logic is re-implemented. This was necessary for maximum performance and flexibility.

class `EventType` (*value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None*)

book_created = 4

When a new book record is created in the database, with the book id as the only argument

book_edited = 8

When a book format is edited, with arguments: (book_id, fmt)

books_removed = 5

When books are removed from the database with the list of book ids as the only argument

format_added = 2

When a format is added to a book, with arguments: (book_id, format)

formats_removed = 3

When formats are removed from a book, with arguments: (mapping of book id to set of formats removed from the book)

indexing_progress_changed = 9

When the indexing progress changes

items_removed = 7

When items such as tags or authors are removed from some books. Arguments: (field_name, affected book ids, ids of removed items)

items_renamed = 6

When items such as tags or authors are renamed in some or all books. Arguments: (field_name, affected book ids, map of old item id to new item id)

metadata_changed = 1

When some metadata is changed for some books, with arguments: (name of changed field, set of affected book ids)

add_books (*books*, *add_duplicates=True*, *apply_import_tags=True*, *preserve_uuid=False*, *run_hooks=True*, *dbapi=None*)

Add the specified books to the library. Books should be an iterable of 2-tuples, each 2-tuple of the form (mi, format_map) where mi is a Metadata object and format_map is a dictionary of the form {fmt: path_or_stream}, for example: {'EPUB': '/path/to/file.epub'}.

Returns a pair of lists: *ids*, *duplicates*. *ids* contains the book ids for all newly created books in the database. *duplicates* contains the (mi, format_map) for all books that already exist in the database as per the simple duplicate detection heuristic used by *has_book()* (strona 359).

add_custom_book_data (*name*, *val_map*, *delete_first=False*)

Add data for name where val_map is a map of book_ids to values. If delete_first is True, all previously stored data for name will be removed.

add_extra_files (*book_id*, *map_of_relpath_to_stream_or_path*, *replace=True*, *auto_rename=False*)

Add extra data files

add_format (*book_id*, *fmt*, *stream_or_path*, *replace=True*, *run_hooks=True*, *dbapi=None*)

Add a format to the specified book. Return True if the format was added successfully.

Parametry

- **replace** – If True replace existing format, otherwise if the format already exists, return False.
- **run_hooks** – If True, file type plugins are run on the format before and after being added.
- **dbapi** – Tylko do użytku wewnętrznego

add_listener (*event_callback_function*, *check_already_added=False*)

Register a callback function that will be called after certain actions are taken on this database. The function must take three arguments: (*EventType* (strona 353), *library_id*, *event_type_specific_data*)

add_notes_resource (*path_or_stream_or_data*, *name: str*, *mtime: float = None*) → int

Add the specified resource so it can be referenced by notes and return its content hash

all_book_ids (*type=<class 'frozenset'>*)

Frozen set of all known book ids.

all_field_for (*field*, *book_ids*, *default_value=None*)

Same as `field_for`, except that it operates on multiple books at once

all_field_ids (*name*)

Frozen set of ids for all values in the field *name*.

all_field_names (*field*)

Frozen set of all fields names (should only be used for many-one and many-many fields)

author_data (*author_ids=None*)

Return author data as a dictionary with keys: name, sort, link

If no authors with the specified ids are found an empty dictionary is returned. If *author_ids* is `None`, data for all authors is returned.

author_sort_from_authors (*authors*, *key_func=<function make_change_case_func.<locals>.change_case>*)

Given a list of authors, return the `author_sort` string for the authors, preferring the author sort associated with the author over the computed string.

books_for_field (*name*, *item_id*)

Return all the books associated with the item identified by *item_id*, where the item belongs to the field *name*.

Returned value is a set of book ids, or the empty set if the item or the field does not exist.

books_in_virtual_library (*vl*, *search_restriction=None*, *virtual_fields=None*)

Return the set of books in the specified virtual library

compress_covers (*book_ids*, *jpeg_quality=100*, *progress_callback=None*)

Compress the cover images for the specified books. A compression quality of 100 will perform lossless compression, otherwise lossy compression.

The progress callback will be called with the *book_id* and the old and new sizes for each book that has been processed. If an error occurs, the new size will be a string with the error details.

copy_cover_to (*book_id*, *dest*, *use_hardlink=False*, *report_file_size=None*)

Copy the cover to the file like object *dest*. Returns `False` if no cover exists or *dest* is the same file as the current cover. *dest* can also be a path in which case the cover is copied to it if and only if the path is different from the current path (taking case sensitivity into account).

copy_format_to (*book_id*, *fmt*, *dest*, *use_hardlink=False*, *report_file_size=None*)

Copy the format *fmt* to the file like object *dest*. If the specified format does not exist, raises `NoSuchFormat` error. *dest* can also be a path (to a file), in which case the format is copied to it, iff the path is different from the current path (taking case sensitivity into account).

cover (*book_id*, *as_file=False*, *as_image=False*, *as_path=False*, *as_pixmap=False*)

Return the cover image or None. By default, returns the cover as a bytestring.

WARNING: Using *as_path* will copy the cover to a temp file and return the path to the temp file. You should delete the temp file when you are done with it.

Parametry

- **as_file** – If True return the image as an open file object (a SpooledTemporaryFile)
- **as_image** – Jeśli jest ustawiony na True, grafika jest zwracana w postaci obiektu QImage
- **as_pixmap** – If True return the image as a QPixmap object
- **as_path** – If True return the image as a path pointing to a temporary file

data_for_find_identical_books ()

Return data that can be used to implement *find_identical_books* () (strona 356) in a worker process without access to the db. See *db.utils* for an implementation.

data_for_has_book ()

Return data suitable for use in *has_book* () (strona 359). This can be used for an implementation of *has_book* () (strona 359) in a worker process without access to the db.

delete_custom_book_data (*name*, *book_ids=()*)

Delete data for name. By default deletes all data, if you only want to delete data for some book ids, pass in a list of book ids.

delete_trash_entry (*book_id*, *category*)

Delete an entry from the trash. Here category is «b» for books and «f» for formats.

embed_metadata (*book_ids*, *only_fmts=None*, *report_error=None*, *report_progress=None*)

Update metadata in all formats of the specified *book_ids* to current metadata in the database.

expire_old_trash ()

Expire entries from the trash that are too old

export_note (*field*, *item_id*) → str

Export the note as a single HTML document with embedded images as data: URLs

fast_field_for (*field_obj*, *book_id*, *default_value=None*)

Same as *field_for*, except that it avoids the extra lookup to get the field object

field_for (*name*, *book_id*, *default_value=None*)

Return the value of the field *name* for the book identified by *book_id*. If no such book exists or it has no defined value for the field *name* or no such field exists, then *default_value* is returned.

default_value is not used for title, title_sort, authors, author_sort and series_index. This is because these always have values in the db. *default_value* is used for all custom columns.

The returned value for *is_multiple* fields are always tuples, even when no values are found (in other words, *default_value* is ignored). The exception is identifiers for which the returned value is always a dictionary. The returned tuples are always in link order, that is, the order in which they were created.

field_ids_for (*name*, *book_id*)

Return the ids (as a tuple) for the values that the field *name* has on the book identified by *book_id*. If there are no values, or no such book, or no such field, an empty tuple is returned.

field_supports_notes (*field=None*) → bool

Return True iff the specified field supports notes. If *field* is None return frozenset of all fields that support notes.

find_identical_books (*mi, search_restriction="", book_ids=None*)

Finds books that have a superset of the authors in *mi* and the same title (title is fuzzy matched). See also [data_for_find_identical_books\(\)](#) (strona 356).

format (*book_id, fmt, as_file=False, as_path=False, preserve_filename=False*)

Return the e-book format as a bytestring or *None* if the format doesn't exist, or we don't have permission to write to the e-book file.

Parametry

- **as_file** – If *True* the e-book format is returned as a file object. Note that the file object is a *SpooledTemporaryFile*, so if what you want to do is copy the format to another file, use [copy_format_to\(\)](#) (strona 355) instead for performance.
- **as_path** – Copies the format file to a temp file and returns the path to the temp file
- **preserve_filename** – If *True* and returning a path the filename is the same as that used in the library. Note that using this means that repeated calls yield the same temp file (which is re-created each time)

format_abspath (*book_id, fmt*)

Return absolute path to the e-book file of format *format*. You should almost never use this, as it breaks the threadsafe promise of this API. Instead use, [copy_format_to\(\)](#) (strona 355).

Currently used only in *calibredb* list, the viewer, edit book, compare_format to original format, open with, bulk metadata edit and the catalogs (via [get_data_as_dict\(\)](#)).

Apart from the viewer, open with and edit book, I don't believe any of the others do any file write I/O with the results of this call.

format_hash (*book_id, fmt*)

Return the hash of the specified format for the specified book. The kind of hash is backend dependent, but is usually SHA-256.

format_metadata (*book_id, fmt, allow_cache=True, update_db=False*)

Return the path, size and mtime for the specified format for the specified book. You should not use path unless you absolutely have to, since accessing it directly breaks the threadsafe guarantees of this API. Instead use the [copy_format_to\(\)](#) (strona 355) method.

Parametry

- **allow_cache** – If *True* cached values are used, otherwise a slow filesystem access is done. The cache values could be out of date if access was performed to the filesystem outside of this API.
- **update_db** – If *True* The max_size field of the database is updated for this book.

formats (*book_id, verify_formats=True*)

Return tuple of all formats for the specified book. If *verify_formats* is *True*, verifies that the files exist on disk.

get_all_items_that_have_notes (*field_name=None*) → *set[int] | dict[str, set[int]]*

Return all *item_ids* for items that have notes in the specified field or all fields if *field_name* is *None*

get_all_link_maps_for_book (*book_id*)

Returns all links for all fields referenced by book identified by *book_id*. If *book_id* doesn't exist then the method returns {}.

Example: Assume author A has link X, author B has link Y, tag S has link F, and tag T has link G. If book 1 has author A and tag T, this method returns {«authors»:{«A»:>X}}, «tags»:{«T»,>G}}. If book 2's author is neither A nor B and has no tags, this method returns {}.

Parametry

book_id – the book id in question.

Zwraca

{field: {field_value, link_value}, ... for all fields with a field_value having a non-empty link value for that book

get_categories (*sort='name', book_ids=None, already_fixed=None, first_letter_sort=False*)

Used internally to implement the Tag Browser

get_custom_book_data (*name, book_ids=(), default=None*)

Get data for name. By default returns data for all book_ids, pass in a list of book ids if you only want some data. Returns a map of book_id to values. If a particular value could not be decoded, uses default for it.

get_id_map (*field*)

Return a mapping of id numbers to values for the specified field. The field must be a many-one or many-many field, otherwise a ValueError is raised.

get_ids_for_custom_book_data (*name*)

Return the set of book ids for which name has data.

get_item_id (*field, item_name*)

Return the item id for item_name (case-insensitive) or None if not found

get_item_ids (*field, item_names*)

Return the item id for item_name (case-insensitive)

get_item_name (*field, item_id*)

Return the item name for the item specified by item_id in the specified field. See also [get_id_map\(\)](#) (strona 358).

get_item_name_map (*field, normalize_func=None*)

Return mapping of item values to ids

get_link_map (*for_field*)

Return a dictionary of links for the supplied field.

Parametry

for_field – the lookup name of the field for which the link map is desired

Zwraca

{field_value:link_value, ...} for non-empty links

get_metadata (*book_id, get_cover=False, get_user_categories=True, cover_as_data=False*)

Return metadata for the book identified by book_id as a [calibre.ebooks.metadata.book.base.Metadata](#) (strona 209) object. Note that the list of formats is not verified. If get_cover is True, the cover is returned, either a path to temp file as mi.cover or if cover_as_data is True then as mi.cover_data.

get_next_series_num_for (*series, field='series', current_indices=False*)

Return the next series index for the specified series, taking into account the various preferences that control next series number generation.

Parametry

- **field** – The series-like field (defaults to the builtin series column)
- **current_indices** – If True, returns a mapping of book_id to current series_index value instead.

get_notes_resource (*resource_hash*) → dict | None

Return a dict containing the resource data and name or None if no resource with the specified hash is found

get_proxy_metadata (*book_id*)

Like *get_metadata()* (strona 358) except that it returns a ProxyMetadata object that only reads values from the database on demand. This is much faster than *get_metadata* when only a small number of fields need to be accessed from the returned metadata object.

get_usage_count_by_id (*field*)

Return a mapping of id to usage count for all values of the specified field, which must be a many-one or many-many field.

has_book (*mi*)

Return True iff the database contains an entry with the same title as the passed in Metadata object. The comparison is case-insensitive. See also *data_for_has_book()* (strona 356).

has_format (*book_id*, *fmt*)

Zwraca True jeśli format istnieje

has_id (*book_id*)

Return True iff the specified book_id exists in the db

import_note (*field*, *item_id*, *path_to_html_file*, *path_is_data=False*)

Import a previously exported note or an arbitrary HTML file as the note for the specified item

init ()

Initialize this cache with data from the backend.

items_with_notes_in_book (*book_id: int*) → dict[str, dict[int, str]]

Return a dict of field to items that have associated notes for that field for the specified book

link_for (*field*, *item_id*)

Return the link, if any, for the specified item or None if no link is found

list_extra_files (*book_id*, *use_cache=False*, *pattern=""*) → Tuple[ExtraFile, ...]

Get information about extra files in the book's directory.

Parametry

- **book_id** – the database book id for the book
- **pattern** – the pattern of filenames to search for. Empty pattern matches all extra files. Patterns must use / as separator. Use the DATA_FILE_PATTERN constant to match files inside the data directory.

Zwraca

A tuple of all extra files matching the specified pattern. Each element of the tuple is ExtraFile(relpath, file_path, stat_result). Where relpath is the relative path of the file to the book directory using / as a separator. stat_result is the result of calling os.stat() on the file.

merge_extra_files (*dest_id*, *src_ids*, *replace=False*)

Merge the extra files from src_ids into dest_id. Conflicting files are auto-renamed unless replace=True in which case they are replaced.

move_book_from_trash (*book_id*)

Undelete a book from the trash directory

move_format_from_trash (*book_id*, *fmt*)

Undelete a format from the trash directory

multisort (*fields, ids_to_sort=None, virtual_fields=None*)

Return a list of sorted book ids. If *ids_to_sort* is *None*, all book ids are returned.

fields must be a list of 2-tuples of the form (*field_name*, *ascending=True* or *False*). The most significant field is the first 2-tuple.

notes_data_for (*field, item_id*) → str

Return all notes data as a dict or *None* if note does not exist

notes_for (*field, item_id*) → str

Return the notes document or an empty string if not found

notes_resources_used_by (*field, item_id*)

Return the set of resource hashes of all resources used by the note for the specified item

pref (*name, default=None, namespace=None*)

Return the value for the specified preference or the value specified as *default* if the preference is not set.

read_backup (*book_id*)

Return the OPF metadata backup for the book as a bytestring or *None* if no such backup exists.

remove_books (*book_ids, permanent=False*)

Remove the books specified by the *book_ids* from the database and delete their format files. If *permanent* is *False*, then the format files are placed in the per-library trash directory.

remove_formats (*formats_map, db_only=False*)

Remove the specified formats from the specified books.

Parametry

- **formats_map** – A mapping of *book_id* to a list of formats to be removed from the book.
- **db_only** – If *True*, only remove the record for the format from the db, do not delete the actual format file from the filesystem.

Zwraca

A map of book id to set of formats actually deleted from the filesystem for that book

remove_items (*field, item_ids, restrict_to_book_ids=None*)

Delete all items in the specified field with the specified ids. Returns the set of affected book ids. *restrict_to_book_ids* is an optional set of books ids. If specified the items will only be removed from those books.

rename_extra_files (*book_id, map_of_relpath_to_new_relpath, replace=False*)

Rename extra data files

rename_items (*field, item_id_to_new_name_map, change_index=True, restrict_to_book_ids=None*)

Rename items from a many-one or many-many field such as tags or series.

Parametry

- **change_index** – When renaming in a series-like field also change the *series_index* values.
- **restrict_to_book_ids** – An optional set of book ids for which the rename is to be performed, defaults to all books.

restore_book (*book_id, mi, last_modified, path, formats, annotations=()*)

Restore the book entry in the database for a book that already exists on the filesystem

restore_original_format (*book_id, original_fmt*)

Restore the specified format from the previously saved ORIGINAL_FORMAT, if any. Return True on success. The ORIGINAL_FORMAT is deleted after a successful restore.

property safe_read_lock

A safe read lock is a lock that does nothing if the thread already has a write lock, otherwise it acquires a read lock. This is necessary to prevent DowngradeLockErrors, which can happen when updating the search cache in the presence of composite columns. Updating the search cache holds an exclusive lock, but searching a composite column involves reading field values via ProxyMetadata which tries to get a shared lock. There may be other scenarios that trigger this as well.

This property returns a new lock object on every access. This lock object is not recursive (for performance) and must only be used in a with statement as with `cache.safe_read_lock`: otherwise bad things will happen.

save_original_format (*book_id, fmt*)

Save a copy of the specified format as ORIGINAL_FORMAT, overwriting any existing ORIGINAL_FORMAT.

search (*query, restriction="", virtual_fields=None, book_ids=None*)

Search the database for the specified query, returning a set of matched book ids.

Parametry

- **restriction** – A restriction that is ANDed to the specified query. Note that restrictions are cached, therefore the search for a AND b will be slower than a with restriction b.
- **virtual_fields** – Used internally (virtual fields such as on_device to search over).
- **book_ids** – If not None, a set of book ids for which books will be searched instead of searching all books.

search_notes (*fts_engine_query="", use_stemming=True, highlight_start=None, highlight_end=None, snippet_size=None, restrict_to_fields=(), return_text=True, result_type=<class 'tuple'>, process_each_result=None, limit=None*)

Search the text of notes using an FTS index. If the query is empty return all notes.

set_conversion_options (*options, fmt='PIPE'*)

options must be a map of the form {book_id:conversion_options}

set_cover (*book_id_data_map*)

Set the cover for this book. The data can be either a QImage, QPixmap, file object or bytestring. It can also be None, in which case any existing cover is removed.

set_field (*name, book_id_to_val_map, allow_case_change=True, do_path_update=True*)

Set the values of the field specified by name. Returns the set of all book ids that were affected by the change.

Parametry

- **book_id_to_val_map** – Mapping of book_ids to values that should be applied.
- **allow_case_change** – If True, the case of many-one or many-many fields will be changed. For example, if a book has the tag `tag1` and you set the tag for another book to `Tag1` then the both books will have the tag `Tag1` if `allow_case_change` is True, otherwise they will both have the tag `tag1`.
- **do_path_update** – Używane wewnętrznie, nie należy go zmieniać

set_link_map (*field, value_to_link_map, only_set_if_no_existing_link=False*)

Sets links for item values in field. Note: this method doesn't change values not in the value_to_link_map

Parametry

- **field** – the lookup name
- **value_to_link_map** – dict(field_value:link, ...). Note that these are values, not field ids.

Zwraca

books changed by setting the link

set_metadata (*book_id, mi, ignore_errors=False, force_changes=False, set_title=True, set_authors=True, allow_case_change=False*)

Set metadata for the book *id* from the *Metadata* object *mi*

Setting force_changes=True will force set_metadata to update fields even if mi contains empty values. In this case, «None» is distinguished from «empty». If mi.XXX is None, the XXX is not replaced, otherwise it is. The tags, identifiers, and cover attributes are special cases. Tags and identifiers cannot be set to None so they will always be replaced if force_changes is true. You must ensure that mi contains the values you want the book to have. Covers are always changed if a new cover is provided, but are never deleted. Also note that force_changes has no effect on setting title or authors.

set_notes_for (*field, item_id, doc: str, searchable_text: str = "", resource_hashes=(), remove_unused_resources=False*) → int

Set the notes document. If the searchable text is different from the document, specify it as searchable_text. If the document references resources their hashes must be present in resource_hashes. Set remove_unused_resources to True to cleanup unused resources, note that updating a note automatically cleans up resources pertaining to that note anyway.

set_pref (*name, val, namespace=None*)

Set the specified preference to the specified value. See also [pref\(\)](#) (strona 360).

split_if_is_multiple_composite (*f, val*)

If *f* is a composite column lookup key and the column is is_multiple then split *v* into unique non-empty values. The comparison is case sensitive. Order is not preserved. Return a list() for compatibility with proxy metadata field getters, for example tags.

tags_older_than (*tag, delta=None, must_have_tag=None, must_have_authors=None*)

Return the ids of all books having the tag *tag* that are older than the specified time. tag comparison is case insensitive.

Parametry

- **delta** – A timedelta object or None. If None, then all ids with the tag are returned.
- **must_have_tag** – If not None the list of matches will be restricted to books that have this tag
- **must_have_authors** – A list of authors. If not None the list of matches will be restricted to books that have these authors (case insensitive).

unretire_note_for (*field, item_id*) → int

Unretire a previously retired note for the specified item. Notes are retired when an item is removed from the database

user_categories_for_books (*book_ids, proxy_metadata_map=None*)

Return the user categories for the specified books. proxy_metadata_map is optional and is useful for

a performance boost, in contexts where a ProxyMetadata object for the books already exists. It should be a mapping of book_ids to their corresponding ProxyMetadata objects.

14.9.2 Dokumentacja API narzędzi do edytowania książek

The e-book editing tools consist of a `calibre.ebooks.oeb.polish.container.Container` (strona 363) object that represents a book as a collection of HTML + resource files, and various tools that can be used to perform operations on the container. All the tools are in the form of module level functions in the various `calibre.ebooks.oeb.polish.*` modules. You obtain a container object for a book at a path like this:

```
from calibre.ebooks.oeb.polish.container import get_container
container = get_container('Path to book file', tweak_mode=True)
```

If you are writing a plugin for the E-book editor, you get the current container for the book being edited like this:

```
from calibre.gui2.tweak_book import current_container
container = current_container()
if container is None:
    report_error # No book has been opened yet
```

The Container object

class `calibre.ebooks.oeb.polish.container.Container` (*rootpath, opfpath, log, clone_data=None*)

A container represents an open e-book as a folder full of files and an OPF file. There are two important concepts:

- The root folder. This is the base of the e-book. All the e-books files are inside this folder or in its sub-folders.
- Names: These are paths to the books» files relative to the root folder. They always contain POSIX separators and are unquoted. They can be thought of as canonical identifiers for files in the book. Most methods on the container object work with names. Names are always in the NFC Unicode normal form.
- Clones: the container object supports efficient on-disk cloning, which is used to implement checkpoints in the e-book editor. In order to make this work, you should never access files on the filesystem directly. Instead, use `raw_data()` (strona 366) or `open()` (strona 365) to read/write to component files in the book.

When converting between hrefs and names use the methods provided by this class, they assume all hrefs are quoted.

abspath_to_name (*fullpath, root=None*)

Convert an absolute path to a canonical name relative to `root`

Parametry

root – The base folder. By default the root for this container object is used.

add_file (*name, data, media_type=None, spine_index=None, modify_name_if_needed=False, process_manifest_item=None*)

Add a file to this container. Entries for the file are automatically created in the OPF manifest and spine (if the file is a text document)

add_name_to_manifest (*name, process_manifest_item=None*)

Add an entry to the manifest for a file with the specified name. Returns the manifest id.

add_properties (*name, *properties*)

Add the specified properties to the manifest item identified by name.

apply_unique_properties (*name*, **properties*)

Ensure that the specified properties are set on only the manifest item identified by name. You can pass None as the name to remove the property from all items.

book_type = 'oeb'

The type of book (epub for EPUB files and azw3 for AZW3 files)

commit (*outpath=None*, *keep_parsed=False*)

Commit all dirtied parsed objects to the filesystem and write out the e-book file at outpath.

Parametry

- **output** – The path to write the saved e-book file to. If None, the path of the original book file is used.
- **keep_parsed** – If True the parsed representations of committed items are kept in the cache.

commit_item (*name*, *keep_parsed=False*)

Commit a parsed object to disk (it is serialized and written to the underlying file). If *keep_parsed* is True the parsed representation is retained in the cache. See also: [parsed\(\)](#) (strona 366)

dirty (*name*)

Mark the parsed object corresponding to name as dirty. See also: [parsed\(\)](#) (strona 366).

exists (*name*)

True iff a file/folder corresponding to the canonical name exists. Note that this function suffers from the limitations of the underlying OS filesystem, in particular case (in)sensitivity. So on a case insensitive filesystem this will return True even if the case of name is different from the case of the underlying filesystem file. See also [has_name\(\)](#) (strona 364)

filesize (*name*)

Return the size in bytes of the file represented by the specified canonical name. Automatically handles dirtied parsed objects. See also: [parsed\(\)](#) (strona 366)

generate_item (*name*, *id_prefix=None*, *media_type=None*, *unique_href=True*)

Add an item to the manifest with href derived from the given name. Ensures uniqueness of href and id automatically. Returns generated item.

get_file_path_for_processing (*name*, *allow_modification=True*)

Similar to `open()` except that it returns a file path, instead of an open file object.

property_guide_type_map

Mapping of guide type to canonical name

has_name (*name*)

Return True iff a file with the same canonical name as that specified exists. Unlike [exists\(\)](#) (strona 364) this method is always case-sensitive.

href_to_name (*href*, *base=None*)

Convert an href (relative to base) to a name. base must be a name or None, in which case self.root is used.

insert_into_xml (*parent*, *item*, *index=None*)

Insert item into parent (or append if index is None), fixing indentation. Only works with self closing items.

is_dir = False

If this container represents an unzipped book (a directory)

iterlinks (*name*, *get_line_numbers=True*)

Iterate over all links in *name*. If *get_line_numbers* is *True* the yields results of the form (link, line_number, offset). Where line_number is the line_number at which the link occurs and offset is the number of characters from the start of the line. Note that offset could actually encompass several lines if not zero.

make_name_unique (*name*)

Ensure that *name* does not already exist in this book. If it does, return a modified version that does not exist.

manifest_has_name (*name*)

Return *True* if the manifest has an entry corresponding to *name*

property manifest_id_map

Mapping of manifest id to canonical names

manifest_items_of_type (*predicate*)

The names of all manifest items whose media-type matches *predicate*. *predicate* can be a set, a list, a string or a function taking a single argument, which will be called with the media-type.

manifest_items_with_property (*property_name*)

All manifest items that have the specified property

property manifest_type_map

Mapping of manifest media-type to list of canonical names of that media-type

property mi

The metadata of this book as a Metadata object. Note that this object is constructed on the fly every time this property is requested, so use it sparingly.

name_to_abspath (*name*)

Convert a canonical name to an absolute OS dependent path

name_to_href (*name*, *base=None*)

Convert a name to a href relative to *base*, which must be a name or *None* in which case *self.root* is used as the base

property names_that_must_not_be_changed

Set of names that must never be renamed. Depends on the e-book file format.

property names_that_must_not_be_removed

Set of names that must never be deleted from the container. Depends on the e-book file format.

property names_that_need_not_be_manifested

Set of names that are allowed to be missing from the manifest. Depends on the e-book file format.

open (*name*, *mode='rb'*)

Open the file pointed to by *name* for direct read/write. Note that this will commit the file if it is dirtied and remove it from the parse cache. You must finish with this file before accessing the parsed version of it again, or bad things will happen.

property opf

The parsed OPF file

opf_get_or_create (*name*)

Convenience method to either return the first XML element with the specified name or create it under the *opf:package* element and then return it, if it does not already exist.

property opf_version

The version set on the OPF's <package> element

property opf_version_parsed

The version set on the OPF's <package> element as a tuple of integers

opf_xpath (expr)

Convenience method to evaluate an XPath expression on the OPF file, has the opf: and dc: namespace prefixes pre-defined.

parsed (name)

Return a parsed representation of the file specified by name. For HTML and XML files an lxml tree is returned. For CSS files a css_parser stylesheet is returned. Note that parsed objects are cached for performance. If you make any changes to the parsed object, you must call *dirty()* (strona 364) so that the container knows to update the cache. See also *replace()* (strona 366).

raw_data (name, decode=True, normalize_to_nfc=True)

Return the raw data corresponding to the file specified by name

Parametry

- **decode** – If True and the file has a text based MIME type, decode it and return a unicode object instead of raw bytes.
- **normalize_to_nfc** – If True the returned unicode object is normalized to the NFC normal form as is required for the EPUB and AZW3 file formats.

relpath (path, base=None)

Convert an absolute path (with os separators) to a path relative to base (defaults to self.root). The relative path is *not* a name. Use *abspath_to_name()* (strona 363) for that.

remove_from_spine (spine_items, remove_if_no_longer_in_spine=True)

Remove the specified items (by canonical name) from the spine. If *remove_if_no_longer_in_spine* is True, the items are also deleted from the book, not just from the spine.

remove_from_xml (item)

Removes item from parent, fixing indentation (works only with self closing items)

remove_item (name, remove_from_guide=True)

Remove the item identified by name from this container. This removes all references to the item in the OPF manifest, guide and spine as well as from any internal caches.

rename (current_name, new_name)

Renames a file from *current_name* to *new_name*. It automatically rebases all links inside the file if the folder the file is in changes. Note however, that links are not updated in the other files that could reference this file. This is for performance, such updates should be done once, in bulk.

replace (name, obj)

Replace the parsed object corresponding to name with obj, which must be a similar object, i.e. an lxml tree for HTML/XML or a css_parser stylesheet for a CSS file.

replace_links (name, replace_func)

Replace all links in name using *replace_func*, which must be a callable that accepts a URL and returns the replaced URL. It must also have a «replaced» attribute that is set to True if any actual replacement is done. Convenient ways of creating such callables are using the *LinkReplacer* and *LinkRebaser* classes.

serialize_item (name)

Convert a parsed object (identified by canonical name) into a bytestring. See *parsed()* (strona 366).

set_spine (*spine_items*)

Set the spine to be *spine_items* where *spine_items* is an iterable of the form (name, linear). Will raise an error if one of the names is not present in the manifest.

property spine_items

An iterator yielding the path for every item in the books» spine. See also: *spine_iter* (strona 367) and *spine_items* (strona 367).

property spine_iter

An iterator that yields item, name, is_linear for every item in the books» spine. item is the lxml element, name is the canonical file name and is_linear is True if the item is linear. See also: *spine_names* (strona 367) and *spine_items* (strona 367).

property spine_names

An iterator yielding name and is_linear for every item in the books» spine. See also: *spine_iter* (strona 367) and *spine_items* (strona 367).

Managing component files in a container

`calibre.ebooks.oeb.polish.replace.replace_links` (*container*, *link_map*, *frag_map*=<function <lambda>>, *replace_in_opf*=False)

Replace links to files in the container. Will iterate over all files in the container and change the specified links in them.

Parametry

- **link_map** – A mapping of old canonical name to new canonical name. For example: `{ 'images/old.png': 'images/new.png' }`
- **frag_map** – A callable that takes two arguments (*name*, *anchor*) and returns a new anchor. This is useful if you need to change the anchors in HTML files. By default, it does nothing.
- **replace_in_opf** – Jeśli fałsz, linki nie są zastąpione w pliku OPF.

`calibre.ebooks.oeb.polish.replace.rename_files` (*container*, *file_map*)

Rename files in the container, automatically updating all links to them.

Parametry

file_map – A mapping of old canonical name to new canonical name, for example: `{ 'text/chapter1.html': 'chapter1.html' }`.

`calibre.ebooks.oeb.polish.replace.get_recommended_folders` (*container*, *names*)

Return the folders that are recommended for the given filenames. The recommendation is based on where the majority of files of the same type are located in the container. If no files of a particular type are present, the recommended folder is assumed to be the folder containing the OPF file.

Pretty printing and auto fixing parse errors

`calibre.ebooks.oeb.polish.pretty.fix_html (container, raw)`

Fix any parsing errors in the HTML represented as a string in raw. Fixing is done using the HTML5 parsing algorithm.

`calibre.ebooks.oeb.polish.pretty.fix_all_html (container)`

Fix any parsing errors in all HTML files in the container. Fixing is done using the HTML5 parsing algorithm.

`calibre.ebooks.oeb.polish.pretty.pretty_html (container, name, raw)`

Pretty print the HTML represented as a string in raw

`calibre.ebooks.oeb.polish.pretty.pretty_css (container, name, raw)`

Pretty print the CSS represented as a string in raw

`calibre.ebooks.oeb.polish.pretty.pretty_xml (container, name, raw)`

Pretty print the XML represented as a string in raw. If name is the name of the OPF, extra OPF-specific prettying is performed.

`calibre.ebooks.oeb.polish.pretty.pretty_all (container)`

Pretty print all HTML/CSS/XML files in the container

Managing book jackets

`calibre.ebooks.oeb.polish.jacket.remove_jacket (container)`

Remove an existing jacket, if any. Returns False if no existing jacket was found.

`calibre.ebooks.oeb.polish.jacket.add_or_replace_jacket (container)`

Either create a new jacket from the book's metadata or replace an existing jacket. Returns True if an existing jacket was replaced.

Splitting and merging of files

`calibre.ebooks.oeb.polish.split.split (container, name, loc_or_xpath, before=True, totals=None)`

Split the file specified by name at the position specified by loc_or_xpath. Splitting automatically migrates all links and references to the affected files.

Parametry

- **loc_or_xpath** – Should be an XPath expression such as `//h:div[@id="split_here"]`. Can also be a *loc* which is used internally to implement splitting in the preview panel.
- **before** – If True the split occurs before the identified element otherwise after it.
- **totals** – Used internally

`calibre.ebooks.oeb.polish.split.multisplit (container, name, xpath, before=True)`

Split the specified file at multiple locations (all tags that match the specified XPath expression). See also: [*split\(\)*](#) (strona 368). Splitting automatically migrates all links and references to the affected files.

Parametry

- **before** – If True the splits occur before the identified element otherwise after it.

`calibre.ebooks.oeb.polish.split.merge` (*container, category, names, master*)

Merge the specified files into a single file, automatically migrating all links and references to the affected files. The file must all either be HTML or CSS files.

Parametry

- **category** – Must be either 'text' for HTML files or 'styles' for CSS files
- **names** – The list of files to be merged
- **master** – Which of the merged files is the *master* file, that is, the file that will remain after merging.

Zarządzanie okładkami

`calibre.ebooks.oeb.polish.cover.set_cover` (*container, cover_path, report=None, options=None*)

Set the cover of the book to the image pointed to by `cover_path`.

Parametry

- **cover_path** – Either the absolute path to an image file or the canonical name of an image in the book. When using an image in the book, you must also set options, see below.
- **report** – An optional callable that takes a single argument. It will be called with information about the tasks being processed.
- **options** – None or a dictionary that controls how the cover is set. The dictionary can have entries: **keep_aspect**: True or False (Preserve aspect ratio of covers in EPUB) **no_svg**: True or False (Use an SVG cover wrapper in the EPUB titlepage) **existing**: True or False (`cover_path` refers to an existing image in the book)

`calibre.ebooks.oeb.polish.cover.mark_as_cover` (*container, name*)

Mark the specified image as the cover image.

`calibre.ebooks.oeb.polish.cover.mark_as_titlepage` (*container, name, move_to_start=True*)

Mark the specified HTML file as the titlepage of the EPUB.

Parametry

- **move_to_start** – If True the HTML file is moved to the start of the spine

Praca z CSS

`calibre.ebooks.oeb.polish.fonts.change_font` (*container, old_name, new_name=None*)

Change a font family from `old_name` to `new_name`. Changes all occurrences of the font family in stylesheets, style tags and style attributes. If the `old_name` refers to an embedded font, it is removed. You can set `new_name` to None to remove the font family instead of changing it.

`calibre.ebooks.oeb.polish.css.remove_unused_css` (*container, report=None, remove_unused_classes=False, merge_rules=False, merge_rules_with_identical_properties=False, remove_unreferenced_sheets=False*)

Remove all unused CSS rules from the book. An unused CSS rule is one that does not match any actual content.

Parametry

- **report** – An optional callable that takes a single argument. It is called with information about the operations being performed.

- **remove_unused_classes** – If True, class attributes in the HTML that do not match any CSS rules are also removed.
- **merge_rules** – If True, rules with identical selectors are merged.
- **merge_rules_with_identical_properties** – If True, rules with identical properties are merged.
- **remove_unreferenced_sheets** – If True, stylesheets that are not referenced by any content are removed

`calibre.ebooks.oeb.polish.css.filter_css(container, properties, names=())`

Remove the specified CSS properties from all CSS rules in the book.

Parametry

- **properties** – Set of properties to remove. For example: `{'font-family', 'color'}`.
- **names** – The files from which to remove the properties. Defaults to all HTML and CSS files in the book.

Praca ze spisem treści

`calibre.ebooks.oeb.polish.toc.from_xpaths(container, xpaths, prefer_title=False)`

Generate a Table of Contents from a list of XPath expressions. Each expression in the list corresponds to a level of the generate ToC. For example: `['//h:h1', '//h:h2', '//h:h3']` will generate a three level Table of Contents from the `<h1>`, `<h2>` and `<h3>` tags.

`calibre.ebooks.oeb.polish.toc.from_links(container)`

Generate a Table of Contents from links in the book.

`calibre.ebooks.oeb.polish.toc.from_files(container)`

Generate a Table of Contents from files in the book.

`calibre.ebooks.oeb.polish.toc.create_inline_toc(container, title=None)`

Create an inline (HTML) Table of Contents from an existing NCX Table of Contents.

Parametry

title – The title for this table of contents.

Edytor książek

class `calibre.gui2.tweak_book.plugin.Tool`

Klasy bazowe: `object`

The base class for individual tools in an Edit Book plugin. Useful members include:

- `self.plugin`: A reference to the `calibre.customize.Plugin` (strona 256) object to which this tool belongs.
- `self.boss` (strona 371)
- `self.gui` (strona 371)

Methods that must be overridden in sub classes:

- `create_action()` (strona 371)
- `register_shortcut()` (strona 371)

name = None

Set this to a unique name it will be used as a key

allowed_in_toolbar = True

If True the user can choose to place this tool in the plugins toolbar

allowed_in_menu = True

If True the user can choose to place this tool in the plugins menu

toolbar_button_popup_mode = 'delayed'

The popup mode for the menu (if any) of the toolbar button. Possible values are «delayed», «instant», «button»

property boss

The `calibre.gui2.tweak_book.boss.Boss` (strona 372) object. Used to control the user interface.

property gui

Główne okno

property current_container

Return the current `calibre.ebooks.oeb.polish.container.Container` (strona 363) object that represents the book being edited.

register_shortcut (*qaction, unique_name, default_keys=(), short_text=None, description=None, **extra_data*)

Register a keyboard shortcut that will trigger the specified `qaction`. This keyboard shortcut will become automatically customizable by the user in the Keyboard shortcuts section of the editor preferences.

Parametry

- **qaction** – A QAction object, it will be triggered when the configured key combination is pressed by the user.
- **unique_name** – A unique name for this shortcut/action. It will be used internally, it must not be shared by any other actions in this plugin.
- **default_keys** – A list of the default keyboard shortcuts. If not specified no default shortcuts will be set. If the shortcuts specified here conflict with either builtin shortcuts or shortcuts from user configuration/other plugins, they will be ignored. In that case, users will have to configure the shortcuts manually via Preferences. For example: `default_keys=('Ctrl+J', 'F9')`.
- **short_text** – An optional short description of this action. If not specified the text from the QAction will be used.
- **description** – An optional longer description of this action, it will be used in the preferences entry for this shortcut.

create_action (*for_toolbar=True*)

Create a QAction that will be added to either the plugins toolbar or the plugins menu depending on `for_toolbar`. For example:

```
def create_action(self, for_toolbar=True):
    ac = QAction(get_icons('myicon.png'), 'Do something')
    if for_toolbar:
        # We want the toolbar button to have a popup menu
        menu = QMenu()
        ac.setMenu(menu)
        menu.addAction('Do something else')
        subaction = menu.addAction('And another')
```

(ciąg dalszy na następnej stronie)

(kontynuacja poprzedniej strony)

```

# Register a keyboard shortcut for this toolbar action be
# careful to do this for only one of the toolbar action or
# the menu action, not both.
self.register_shortcut(ac, 'some-unique-name', default_keys=('Ctrl+K',
↵))
return ac

```

Zobacz także:Method `register_shortcut()` (strona 371).

Controlling the editor's user interface

The e-book editor's user interface is controlled by a single global *Boss* object. This has many useful methods that can be used in plugin code to perform common tasks.

class `calibre.gui2.tweak_book.boss.Boss` (*parent*, *notify=None*)

add_savepoint (*msg*)

Create a restore checkpoint with the name specified as *msg*

apply_container_update_to_gui (*mark_as_modified=True*)

Update all the components of the user interface to reflect the latest data in the current book container.

Parametry

mark_as_modified – If *True*, the book will be marked as modified, so the user will be prompted to save it when quitting.

close_editor (*name*)

Close the editor that is editing the file specified by *name*

commit_all_editors_to_container ()

Commit any changes that the user has made to files open in editors to the container. You should call this method before performing any actions on the current container

property currently_editing

Return the name of the file being edited currently or *None* if no file is being edited

edit_file (*name*, *syntax=None*, *use_template=None*)

Open the file specified by *name* in an editor

Parametry

- **syntax** – The media type of the file, for example, `'text/html'`. If not specified it is guessed from the file extension.
- **use_template** – A template to initialize the opened editor with

open_book (*path=None*, *edit_file=None*, *clear_notify_data=True*, *open_folder=False*, *search_text=None*)

Open the e-book at *path* for editing. Will show an error if the e-book is not in a supported format or the current book has unsaved changes.

Parametry

edit_file – The name of a file inside the newly opened book to start editing. Can also be a list of names.

rewind_savepoint ()

Undo the previous creation of a restore checkpoint, useful if you create a checkpoint, then abort the operation with no changes

save_book ()

Save the book. Saving is performed in the background

set_modified ()

Oznacz książkę jako zmodyfikowaną

show_current_diff (allow_revert=True, to_container=None)

Show the changes to the book from its last checkpointed state

Parametry

- **allow_revert** – If True the diff dialog will have a button to allow the user to revert all changes
- **to_container** – A container object to compare the current container to. If None, the previously checkpointed container is used

show_editor (name)

Show the editor that is editing the file specified by name

sync_preview_to_editor ()

Sync the position of the preview panel to the current cursor position in the current editor

Zarządzanie prawami cyfrowymi (DRM)

Zarządzanie prawami cyfrowymi (DRM) to ogólny termin określający technologie kontroli dostępu, które mogą być wykorzystywane przez producentów sprzętu, wydawców, właścicieli praw autorskich i osoby fizyczne w celu nałożenia ograniczeń na korzystanie z treści i urządzeń cyfrowych. Czasami jest to również lekceważąco określane jako zarządzanie ograniczeniami cyfrowymi. Termin ten jest używany do opisanie każdej technologii, która uniemożliwia korzystanie (zgodnie z prawem lub w inny sposób) treści cyfrowych, które nie było pożądane lub przewidziane przez dostawcę treści. Termin generalnie nie odnosi się do innych form ochrony przed kopiowaniem, które można obejść bez modyfikowania pliku lub urządzenia, takich jak numery seryjne lub pliki kluczy. Może również odnosić się do ograniczeń związanych z konkretnymi wystąpieniami dzieł lub urządzeń cyfrowych. Technologie DRM próbują kontrolować korzystanie z mediów cyfrowych, uniemożliwiając użytkownikom końcowym dostęp, kopiowanie lub konwersję do innych formatów. Zobacz [Wikipedia](https://en.wikipedia.org/wiki/Digital_rights_management)¹³³.

15.1 Co DRM niesie dla mnie osobiście?

Wydawcy e-booków DRMed twierdzą, że DRM służy w całości autorom i ochronie ich artystycznej integralności oraz zapobieganiu piractwu. Ale DRM NIE zapobiega piractwu. Osoby, które chcą piratować treści lub gdy kupujesz e-booka z DRM, tak naprawdę nie jesteś jego właścicielem, ale kupiłeś pozwolenie na wykorzystanie go w sposób podyktowany Tobie przez sprzedawcę. DRM ogranicza to, co możesz zrobić z e-bookami, które „kupiłeś”. Często ludzie, którzy kupują książki z DRM, nie zdają sobie sprawy z zakresu tych ograniczeń. Te ograniczenia uniemożliwiają zmianę formatu e-booka według własnych upodobań, w tym wprowadzanie zmian stylistycznych, takich jak dostosowywanie rozmiarów czcionek, chociaż istnieje oprogramowanie, które umożliwia wykonywanie takich czynności w przypadku książek bez DRM. Ludzie często są zaskoczeni, że e-book, który kupili w określonym formacie, nie może zostać przekonwertowany na inny format, jeśli ebook ma DRM. Więc jeśli masz Amazon Kindle i kupujesz książkę sprzedawaną przez Barnes and Nobles, powinieneś wiedzieć, że jeśli ten e-book ma DRM, nie będziesz mógł go przeczytać na swoim Kindle. Zauważ, że mówię o książce, którą kupujesz, a nie kradniesz lub piratujesz, ale KUPUJĘ.

¹³³ https://en.wikipedia.org/wiki/Digital_rights_management

15.2 Co DRM daje autorom?

Wydawcy e-booków DRM twierdzą, że DRM służy w całości autorom i ochronie ich artystycznej integralności oraz zapobieganiu piractwu. Ale DRM NIE zapobiega piractwu. Ludzie, którzy chcą piracić treści lub używać pirackich treści, nadal to robią i odnoszą sukces. Trzy główne systemy DRM dla e-booków są obecnie obsługiwane przez Amazon, Adobe, Barnes i Noble, a wszystkie trzy systemy DRM zostały złamane. Wszystko, co robi DRM, to niedogodności dla uprawnionych użytkowników. Można argumentować, że w rzeczywistości szkodzi to autorom, ponieważ ludzie, którzy kupiliby książkę, wybierają piracką wersję, ponieważ nie są skłonni znosić DRM. Te, które piraciłyby w przypadku braku DRM, robią to również w jego obecności. Powtarzając, kluczową kwestią jest to, że DRM * nie zapobiega piractwu *. Tak więc DRM jest nie tylko bezcelowe i szkodliwe dla kupujących e-booki, ale także strata pieniędzy.

15.3 DRM a wolność

Chociaż treści cyfrowe mogą być wykorzystywane do udostępniania informacji i dzieł twórczych dla wszystkich oraz do wzmacniania pozycji ludzkości, nie leży to w interesie niektórych wydawców, którzy chcą odciągnąć ludzi od tej możliwości wolności po prostu po to, aby zachować ich znaczenie w rozwijającym się świecie. tak szybko, że nie mogą nadążyć.

15.4 Dlaczego calibre nie umożliwia DRM?

calibre to oprogramowanie typu open source, podczas gdy DRM ze swej natury jest zamknięte. Gdyby kaliber miał obsługiwać otwieranie lub przeglądanie plików DRM, można by go lekko zmodyfikować, aby służył jako narzędzie do usuwania DRM, co jest niezgodne z dzisiejszym prawem. Oprogramowanie open source i DRM to zderzenie zasad. Podczas gdy DRM polega na kontrolowaniu użytkownika, oprogramowanie open source ma na celu wzmocnienie użytkownika. Te dwie rzeczy po prostu nie mogą współistnieć.

15.5 Jaki jest stosunek calibre do dostawców treści?

Jesteśmy głęboko przekonani, że autorzy i inni dostawcy treści powinni otrzymywać wynagrodzenie za ich wysiłki, ale DRM nie jest sposobem, aby się do tego zabrać. Opracowujemy bazę danych e-booków wolnych od DRM z różnych źródeł, aby pomóc Ci znaleźć alternatywy wolne od DRM oraz pomóc niezależnym autorom i wydawcom e-booków wolnych od DRM w upublicznianiu ich treści. Mamy nadzieję, że uznasz to za przydatne i prosimy, abyś nie piracił treści udostępnionych tutaj.

15.6 Jak mogę wspomóc walkę z DRM?

Jako ktoś, kto czyta i kupuje e-booki, możesz pomóc w walce z DRM. Nie kupuj e-booków z DRM. Jest kilku wydawców, którzy publikują e-booki bez DRM. Postaraj się sprawdzić, czy mają e-booka, którego szukasz. Jeśli lubisz książki niektórych niezależnych autorów, którzy sprzedają e-booki bez DRM i możesz sobie na to pozwolić, przekaz im darowizny. Są to dobrze wydane pieniądze, ponieważ ich e-booki są zwykle tańsze (mogą być wyjątki) niż te, które kupujesz od wydawców książek DRM i prawdopodobnie będą działać na wszystkich posiadanych urządzeniach w przyszłości, oszczędzając koszty zakupu e-book ponownie. Nie zniechęcaj wydawców i autorów e-książek bez DRM poprzez piractwo ich treści. Dostawcy treści zasługują na wynagrodzenie za swój wysiłek. Nie karz ich za to, że próbują poprawić wrażenia z czytania, udostępniając e-booki wolne od DRM. Na dłuższą metę jest to dla ciebie szkodliwe. Jeśli kupiłeś książki od sprzedawców, którzy posiadają zarówno DRM, jak i książki wolne od DRM, nie wiedząc, czy mają DRM, czy też nie warto zostawić komentarza lub recenzji na stronie, informując przyszłych kupujących o jego statusie

DRM. Wielu sprzedawców uważa, że nie jest ważne, aby wyraźnie wskazywać kupującym, czy e-book zawiera DRM, czy nie. Tu¹³⁴ znajdziesz przewodnik po życiu bez DRM.

¹³⁴ <https://www.defectivebydesign.org/guide/ebooks>

RSS

**** RSS **** (Really Simple Syndication) to format kanału internetowego, który jest używany do publikowania często aktualizowanych treści, takich jak artykuły z wiadomościami, posty na blogu itp. Jest to format szczególnie przystosowany do czytania przez komputery oraz jest zatem preferowanym sposobem pobierania treści z internetu do e-booka. W Internecie jest używanych wiele innych formatów kanałów i firma Calibre rozumie większość z nich. W szczególności ma dobre wsparcie dla formatu *** ATOM ***, który jest powszechnie używany w blogach.

źródło

Przepis to zestaw instrukcji, które uczą kalibru, jak przekształcić internetowe źródło wiadomości, takie jak czasopismo lub blog, w e-book. Przepis to zasadniczo kod „Python <<https://www.python.org>>”_. W związku z tym jest w stanie przekształcić dowolnie złożone źródła wiadomości w e-booki. Na najprostszym poziomie jest to tylko zbiór zmiennych, takich jak adresy URL, które dostarczają kalibru wystarczającej ilości informacji, aby wyjść do Internetu i pobrać wiadomości.

HTML

HTML (*Hyper Text Mark-Up Language*), podzbiór języka SGML (Standard Generalized Mark-Up Language), przeznaczonego do publikowania danych w formie elektronicznej. Jest standardem dla WWW.

CSS

CSS (*Cascading Style Sheets*) jest językiem, opisującym wygląd treści zawartych w dokumencie *HTML*.

API

API (*Application Programming Interface*) to interfejs programistyczny, za pomocą którego biblioteka może być wywoływane przez programy komputerowe.

LRF

**** LRF **** Format e-booka odczytywany przez czytniki e-booków SONY.

URL

URL (*Uniform Resource Locator*) format adresów internetowych, np.: <http://example.com>

wyrażenie regularne

Wyrażenia regularne zapewniają zwięzły i elastyczny sposób identyfikowania ciągów interesującego tekstu, takich jak określone znaki, słowa lub wzorce znaków. Zobacz :doc:`samouczek`, <regexp> aby zapoznać się z wprowadzeniem do wyrażeń regularnych.

C

- `calibre.customize`, 256
- `calibre.customize.conversion`, 266
- `calibre.db.cache`, 353
- `calibre.devices.interface`, 269
- `calibre.ebooks.metadata.book.base`, 209
- `calibre.ebooks.metadata.sources.base`, 262
- `calibre.ebooks.oeb.polish.container`, 363
- `calibre.ebooks.oeb.polish.cover`, 369
- `calibre.ebooks.oeb.polish.css`, 369
- `calibre.ebooks.oeb.polish.jacket`, 368
- `calibre.ebooks.oeb.polish.pretty`, 368
- `calibre.ebooks.oeb.polish.replace`, 367
- `calibre.ebooks.oeb.polish.split`, 368
- `calibre.ebooks.oeb.polish.toc`, 370
- `calibre.gui2.tweak_book.boss`, 372
- `calibre.utils.formatter_functions`, 190
- `calibre.web.feeds.news`, 39

Symbole

-1	calibredb-add command line option, 316	calibredb-export command line option, 318
-H	ebook-polish command line option, 338	--allowed-plugin fetch-ebook-metadata command line option, 340
-I	calibredb-add command line option, 316	--append calibredb-set_custom command line option, 323
	fetch-ebook-metadata command line option, 340	--as-extra-data-file calibredb-add_format command line option, 317
-S	calibredb-add command line option, 316	--as-opf calibredb-show_metadata command line option, 317
-T	calibredb-add command line option, 316	--ascending calibredb-list command line option, 314
-U	ebook-polish command line option, 339	--asciiize ebook-convert command line option, 329
--access-log	calibre-server command line option, 307	--attachment calibre-smtp command line option, 311
--add	calibredb-add command line option, 316	--auth-mode calibre-server command line option, 307
--add-plugin	calibre-customize command line option, 304	--author-sort ebook-convert command line option, 334
--add-simple-plugin	calibre-debug command line option, 305	ebook-meta command line option, 336
--add-soft-hyphens	ebook-polish command line option, 338	--authors calibredb-add command line option, 315
--ajax-timeout	calibre-server command line option, 307	ebook-convert command line option, 334
--all	calibredb-backup_metadata command line option, 324	ebook-meta command line option, 336
		fetch-ebook-metadata command line option, 340
		--auto-reload calibre-server command line option, 307

--automerger
calibredb-add command line option,
315

--ban-after
calibre-server command line option,
307

--ban-for
calibre-server command line option,
307

--base-dir
web2disk command line option, 343

--base-font-size
ebook-convert command line option,
329

--book-list-mode
calibre-server command line option,
308

--book-producer
ebook-convert command line option,
334
ebook-meta command line option, 336

--build-plugin
calibre-customize command line
option, 304

--cafile
calibre-smtp command line option, 312

--catalog-title
calibredb-catalog command line
option, 320

--categories
calibredb-list_categories command
line option, 324

--category
ebook-meta command line option, 336

--change-justification
ebook-convert command line option,
329

--chapter
ebook-convert command line option,
333

--chapter-mark
ebook-convert command line option,
333

--command
calibre-debug command line option,
305

--comments
ebook-convert command line option,
334
ebook-meta command line option, 337

--compress-images
ebook-polish command line option, 338

--compress-min-size
calibre-server command line option,
308

--continue
ebook-viewer command line option, 339

--cover
calibredb-add command line option,
315
ebook-convert command line option,
335
ebook-meta command line option, 337
ebook-polish command line option, 338
fetch-ebook-metadata command line
option, 340

--cross-reference-authors
calibredb-catalog command line
option, 320

--csv
calibredb-check_library command
line option, 323
calibredb-list_categories command
line option, 324

--custom-list-template
calibre-server command line option,
308

--customize-plugin
calibre-customize command line
option, 304

--daemonize
calibre-server command line option,
308

--date
ebook-meta command line option, 337

--debug-device-driver
calibre-debug command line option,
305

--debug-pipeline
calibredb-catalog command line
option, 320
ebook-convert command line option,
335

--default-programs
calibre-debug command line option,
305

--delay
web2disk command line option, 343

--detach
calibre command line option, 304
ebook-edit command line option, 336
ebook-viewer command line option, 339

--details
calibredb-custom_columns command
line option, 322

--dialect
calibredb-list_categories command

line option, 324	332
--diff	--disable-use-bonjour
calibre-debug command line option, 305	calibre-server command line option, 308
--disable-allow-socket-preallocation	--disable-use-sendfile
calibre-server command line option, 308	calibre-server command line option, 309
--disable-auth	--display
calibre-server command line option, 308	calibredb-add_custom_column command line option, 322
--disable-dehyphenate	--displayed-fields
ebook-convert command line option, 332	calibre-server command line option, 308
--disable-delete-blank-paragraphs	--do-not-match-on-related-words
ebook-convert command line option, 332	calibredb-fts_search command line option, 326
--disable-fallback-to-detected-interface	--dont-asciiize
calibre-server command line option, 308	calibredb-export command line option, 318
--disable-fix-indents	--dont-download-stylesheets
ebook-convert command line option, 332	web2disk command line option, 343
--disable-font-rescaling	--dont-output-resources
ebook-convert command line option, 329	lrf2lrs command line option, 341
--disable-format-scene-breaks	--dont-replace
ebook-convert command line option, 332	calibredb-add_format command line option, 317
--disable-hyphenation	--dont-save-cover
lrfviewer command line option, 342	calibredb-export command line option, 318
--disable-italicize-common-cases	--dont-save-extra-files
ebook-convert command line option, 332	calibredb-export command line option, 318
--disable-local-write	--dont-split-on-page-breaks
calibre-server command line option, 308	ebook-convert command line option, 328
--disable-log-not-found	--dont-update-metadata
calibre-server command line option, 308	calibredb-export command line option, 318
--disable-markup-chapter-headings	--dont-verify-server-certificate
ebook-convert command line option, 332	calibre-smtp command line option, 312
--disable-plugin	--dont-write-opf
calibre-customize command line option, 304	calibredb-export command line option, 318
--disable-remove-fake-margins	--duplicate-links-in-toc
ebook-convert command line option, 333	ebook-convert command line option, 334
--disable-renumber-headings	--duplicates
ebook-convert command line option, 332	calibredb-add command line option, 315
--disable-unwrap-lines	--edit-book
ebook-convert command line option,	calibre-debug command line option, 305
	--embed-all-fonts
	ebook-convert command line option,

329
 --embed-font-family
 ebook-convert command line option, 329
 --embed-fonts
 ebook-polish command line option, 338
 --empty
 calibredb-add command line option, 315
 --enable-allow-socket-preallocation
 calibre-server command line option, 308
 --enable-auth
 calibre-server command line option, 308
 --enable-fallback-to-detected-interface
 calibre-server command line option, 308
 --enable-heuristics
 ebook-convert command line option, 332
 --enable-local-write
 calibre-server command line option, 308
 --enable-log-not-found
 calibre-server command line option, 308
 --enable-plugin
 calibre-customize command line option, 304
 --enable-use-bonjour
 calibre-server command line option, 308
 --enable-use-sendfile
 calibre-server command line option, 309
 --encoding
 web2disk command line option, 343
 --encryption-method
 calibre-smtp command line option, 312
 --epub-flatten
 ebook-convert command line option, 328
 --epub-inline-toc
 ebook-convert command line option, 328
 --epub-max-image-size
 ebook-convert command line option, 328
 --epub-toc-at-end
 ebook-convert command line option, 328
 --epub-version
 ebook-convert command line option, 328
 calibredb-catalog command line option, 320
 --exclusion-rules
 calibredb-catalog command line option, 320
 --exec-file
 calibre-debug command line option, 306
 --expand-css
 ebook-convert command line option, 329
 --explode-book
 calibre-debug command line option, 306
 --export-all-calibre-data
 calibre-debug command line option, 306
 --extra-css
 ebook-convert command line option, 330
 --extract-to
 ebook-convert command line option, 328
 --field
 calibredb-set_metadata command line option, 318
 --fields
 calibredb-list command line option, 314
 --filter-css
 ebook-convert command line option, 330
 --filter-regexp
 web2disk command line option, 343
 --fix-multiprocessing
 calibre-debug command line option, 306
 --flow-size
 ebook-convert command line option, 328
 --font-size-mapping
 ebook-convert command line option, 330
 --for-machine
 calibredb-list command line option, 314
 --force
 calibredb-remove_custom_column command line option, 322
 --force-reload
 ebook-viewer command line option, 339

```

--fork
    calibre-smtp command line option,311
--formats
    calibredb-export      command    line
        option,319
--from-opf
    ebook-convert command line option,
        335
    ebook-meta command line option,337
--full-screen
    ebook-viewer command line option,339
--fullscreen
    ebook-viewer command line option,339
--generate-authors
    calibredb-catalog      command    line
        option,320
--generate-descriptions
    calibredb-catalog      command    line
        option,320
--generate-genres
    calibredb-catalog      command    line
        option,320
--generate-recently-added
    calibredb-catalog      command    line
        option,320
--generate-series
    calibredb-catalog      command    line
        option,320
--generate-titles
    calibredb-catalog      command    line
        option,320
--genre-source-field
    calibredb-catalog      command    line
        option,320
--get-cover
    ebook-meta command line option,337
--gui
    calibre-debug command line option,
        306
--gui-debug
    calibre-debug command line option,
        306
--header-note-source-field
    calibredb-catalog      command    line
        option,320
--help
    calibre command line option,304
    calibre-customize      command    line
        option,305
    calibre-debug command line option,
        306
    calibre-server command line option,
        309
    calibre-smtp command line option,311
    command line option,314
    ebook-convert command line option,
        327
    ebook-edit command line option,336
    ebook-meta command line option,337
    ebook-polish command line option,338
    ebook-viewer command line option,339
    fetch-ebook-metadata command line
        option,340
    lrf2lrs command line option,341
    lrfviewer command line option,342
    lrs2lrf command line option,342
    web2disk command line option,343
--html-unwrap-factor
    ebook-convert command line option,
        332
--identifier
    calibredb-add command line option,
        316
    ebook-meta command line option,337
    fetch-ebook-metadata command line
        option,340
--ids
    calibredb-catalog      command    line
        option,319
--ignore
    calibredb-add command line option,
        316
--ignore_extensions
    calibredb-check_library command
        line option,323
--ignore_names
    calibredb-check_library command
        line option,323
--ignore-plugins
    calibre command line option,304
--ignored-fields
    calibre-server command line option,
        309
--implode-book
    calibre-debug command line option,
        306
--import-calibre-data
    calibre-debug command line option,
        306
--include-snippets
    calibredb-fts_search command line
        option,326
--index
    ebook-meta command line option,337
--indexing-speed
    calibredb-fts_index command line
        option,325
--indexing-threshold

```

calibredb-fts_search command line option, 326	334
--input-encoding ebook-convert command line option, 328	--library-path command line option, 314
--input-profile ebook-convert command line option, 327	--limit calibredb-list command line option, 314
--insert-blank-line ebook-convert command line option, 330	calibredb-search command line option, 325
--insert-blank-line-size ebook-convert command line option, 330	--line-height ebook-convert command line option, 330
--insert-metadata ebook-convert command line option, 333	--line-width calibredb-list command line option, 314
--inspect-mobi calibre-debug command line option, 306	--linearize-tables ebook-convert command line option, 330
--is-multiple calibredb-add_custom_column command line option, 322	--list-fields calibredb-set_metadata command line option, 318
--isbn calibredb-add command line option, 316	--list-plugins calibre-customize command line option, 305
ebook-convert command line option, 335	--list-recipes ebook-convert command line option, 327
ebook-meta command line option, 337	--listen-on calibre-server command line option, 309
fetch-ebook-metadata command line option, 340	--localhost calibre-smtp command line option, 311
--item_count calibredb-list_categories command line option, 324	--log calibre-server command line option, 309
--jacket ebook-polish command line option, 338	--lrf-bookid ebook-meta command line option, 337
--keep-ligatures ebook-convert command line option, 330	--lrs lrs2lrf command line option, 342
--language ebook-convert command line option, 335	--manage-users calibre-server command line option, 309
ebook-meta command line option, 337	--margin-bottom ebook-convert command line option, 330
--languages calibredb-add command line option, 316	--margin-left ebook-convert command line option, 330
--level1-toc ebook-convert command line option, 334	--margin-right ebook-convert command line option, 330
--level2-toc ebook-convert command line option, 334	--margin-top ebook-convert command line option, 331
--level3-toc ebook-convert command line option,	--match-end-marker

calibredb-fts_search command line option, 326	--num-per-page calibre-server command line option, 310
--match-regex web2disk command line option, 343	--one-book-per-directory calibredb-add command line option, 316
--match-start-marker calibredb-fts_search command line option, 326	--only-formats calibredb-embed_metadata command line option, 325
--max-files web2disk command line option, 343	--open-at ebook-viewer command line option, 340
--max-header-line-size calibre-server command line option, 309	--opf ebook-polish command line option, 338 fetch-ebook-metadata command line option, 340
--max-job-time calibre-server command line option, 309	--outbox calibre-smtp command line option, 311
--max-jobs calibre-server command line option, 309	--output lrf2lrs command line option, 341 lrs2lrf command line option, 342
--max-log-size calibre-server command line option, 309	--output-format calibredb-fts_search command line option, 326
--max-opds-items calibre-server command line option, 309	--output-profile calibredb-catalog command line option, 321 ebook-convert command line option, 327
--max-opds-ungrouped-items calibre-server command line option, 309	--page-breaks-before ebook-convert command line option, 333
--max-recursions web2disk command line option, 343	--password calibre-smtp command line option, 312 command line option, 314
--max-request-body-size calibre-server command line option, 309	--paths calibre-debug command line option, 306
--max-toc-links ebook-convert command line option, 334	--permanent calibredb-remove command line option, 317
--merge-comments-rule calibredb-catalog command line option, 320	--pidfile calibre-server command line option, 310
--minimum-line-height ebook-convert command line option, 331	--port calibre-server command line option, 310 calibre-smtp command line option, 312
--new-instance ebook-viewer command line option, 340	--prefer-metadata-cover ebook-convert command line option, 333
--no-chapters-in-toc ebook-convert command line option, 334	--prefix calibredb-list command line option, 315
--no-default-epub-cover ebook-convert command line option, 328	--prefix-rules
--no-svg-cover ebook-convert command line option, 328	
--no-update-check calibre command line option, 304	

calibredb-catalog	command line option, 321	--remove-soft-hyphens	ebook-polish command line option, 338
--preserve-cover-aspect-ratio		--remove-unused-css	ebook-polish command line option, 339
ebook-convert	command line option, 329	--replace-scene-breaks	ebook-convert command line option, 332
--preset		--replace-whitespace	calibredb-export command line option, 319
calibredb-catalog	command line option, 321	--report	calibredb-check_library command line option, 323
--pretty-print		--restrict-to	calibredb-fts_search command line option, 326
ebook-convert	command line option, 329	--run-plugin	calibre-debug command line option, 306
--profile		--run-test	calibre-debug command line option, 306
lrfviewer	command line option, 342	--run-without-debug	calibre-debug command line option, 306
--progress		--search	calibredb-catalog command line option, 319
calibredb-export	command line option, 319		calibredb-list command line option, 315
--pubdate		--search-replace	ebook-convert command line option, 332
ebook-convert	command line option, 335	--search-the-net-urls	calibre-server command line option, 310
--publisher		--select-text	ebook-edit command line option, 336
ebook-convert	command line option, 335	--separator	calibredb-list command line option, 315
ebook-meta	command line option, 337	--series	calibredb-add command line option, 316
--raise-window			ebook-convert command line option, 335
ebook-viewer	command line option, 340		ebook-meta command line option, 337
--rating		--series-index	calibredb-add command line option, 316
ebook-convert	command line option, 335		ebook-convert command line option, 335
ebook-meta	command line option, 337	--shutdown-running-calibre	calibre command line option, 304
--read-metadata-from-opf			
ebook-convert	command line option, 335		
--really-do-it			
calibredb-restore_database	command line option, 323		
--recurse			
calibredb-add	command line option, 316		
--relay			
calibre-smtp	command line option, 312		
--remove-first-image			
ebook-convert	command line option, 333		
--remove-jacket			
ebook-polish	command line option, 338		
--remove-paragraph-spacing			
ebook-convert	command line option, 331		
--remove-paragraph-spacing-indent-size			
ebook-convert	command line option, 331		
--remove-plugin			
calibre-customize	command line option, 305		

calibre-debug command line option, 307	--tags calibredb-add command line option, 316
--shutdown-timeout calibre-server command line option, 310	ebook-convert command line option, 335
--single-dir calibredb-export command line option, 319	ebook-meta command line option, 337
--smarten-punctuation ebook-convert command line option, 331	--template calibredb-export command line option, 319
ebook-polish command line option, 339	calibredb-list command line option, 315
--sort-by calibredb-list command line option, 315	--template_file calibredb-list command line option, 315
--sr1-replace ebook-convert command line option, 332	--template_heading calibredb-list command line option, 315
--sr1-search ebook-convert command line option, 332	--test-build calibre-debug command line option, 307
--sr2-replace ebook-convert command line option, 333	--thumb-width calibredb-catalog command line option, 321
--sr2-search ebook-convert command line option, 333	--timefmt calibredb-export command line option, 319
--sr3-replace ebook-convert command line option, 333	--timeout calibre-server command line option, 310
--sr3-search ebook-convert command line option, 333	calibre-smtp command line option, 311
--ssl-certfile calibre-server command line option, 310	command line option, 314
--ssl-keyfile calibre-server command line option, 310	fetch-ebook-metadata command line option, 341
--start-in-tray calibre command line option, 304	web2disk command line option, 343
--start-reading-at ebook-convert command line option, 333	--timestamp ebook-convert command line option, 335
--subject calibre-smtp command line option, 311	--title calibredb-add command line option, 316
--subset-embedded-fonts ebook-convert command line option, 331	ebook-convert command line option, 335
--subset-font calibre-debug command line option, 307	ebook-meta command line option, 337
--subset-fonts ebook-polish command line option, 339	fetch-ebook-metadata command line option, 341
	--title-sort ebook-convert command line option, 335
	ebook-meta command line option, 337
	--to-dir calibredb-export command line option, 319
	--to-lowercase calibredb-export command line

option, 319

--to-opf
ebook-meta command line option, 337

--toc-filter
ebook-convert command line option, 334

--toc-threshold
ebook-convert command line option, 334

--toc-title
ebook-convert command line option, 329

--transform-css-rules
ebook-convert command line option, 331

--transform-html-rules
ebook-convert command line option, 331

--trusted-ips
calibre-server command line option, 310

--unsmarten-punctuation
ebook-convert command line option, 331

--upgrade-book
ebook-polish command line option, 339

--url-prefix
calibre-server command line option, 310

--use-auto-toc
ebook-convert command line option, 334

--use-existing-cover
calibredb-catalog command line option, 321

--userdb
calibre-server command line option, 310

--username
calibre-smtp command line option, 312
command line option, 314

--vacuum-fts-db
calibredb-check_library command line option, 323

--verbose
calibre command line option, 304
calibre-smtp command line option, 311
calibredb-catalog command line option, 319
ebook-convert command line option, 335
ebook-polish command line option, 339
fetch-ebook-metadata command line option, 341
lrf2lrs command line option, 341
lrfviewer command line option, 342
lrs2lrf command line option, 342
web2disk command line option, 343

--version
calibre command line option, 304
calibre-customize command line option, 305
calibre-debug command line option, 307
calibre-server command line option, 310
calibre-smtp command line option, 311
command line option, 314
ebook-convert command line option, 327
ebook-edit command line option, 336
ebook-meta command line option, 338
ebook-polish command line option, 339
ebook-viewer command line option, 340
fetch-ebook-metadata command line option, 341
lrf2lrs command line option, 341
lrfviewer command line option, 342
lrs2lrf command line option, 342
web2disk command line option, 343

--viewer
calibre-debug command line option, 307

--visual-debug
lrfviewer command line option, 342

--wait-for-completion
calibredb-fts_index command line option, 325

--white-background
lrfviewer command line option, 342

--width
calibredb-list_categories command line option, 324

--with-library
calibre command line option, 304
command line option, 314

--worker-count
calibre-server command line option, 310

-a
calibre-customize command line option, 304
calibre-smtp command line option, 311
calibredb-add command line option, 315
calibredb-set_custom command line option, 323
ebook-meta command line option, 336

	fetch-ebook-metadata	command line	-g		
	option,	340		calibre-debug	command line option,
-b				306	
	calibre-customize	command line	-h		calibre command line option,
	option,	304		304	
-c				calibre-customize	command line
	calibre-debug	command line		option,	305
	305			calibre-debug	command line option,
	calibreddb-add	command line		306	
	315			calibre-server	command line option,
	calibreddb-check_library	command		309	
	line option,	323		calibre-smtp	command line option,
	calibreddb-list_categories	command		311	
	line option,	324		command line option,	314
	ebook-meta	command line		ebook-convert	command line option,
	option,	337		327	
	ebook-polish	command line		ebook-edit	command line option,
	option,	338		ebook-meta	command line option,
	fetch-ebook-metadata	command line		337	
	option,	340		ebook-polish	command line option,
				338	
-d				ebook-viewer	command line option,
	calibre-debug	command line		339	
	305			fetch-ebook-metadata	command line
	calibreddb-add	command line		option,	340
	315			lrf2lrs	command line option,
	calibreddb-custom_columns	command		341	
	line option,	322		lrfviewer	command line option,
	ebook-convert	command line		342	
	option,	335	-i	lrs2lrf	command line option,
	ebook-meta	command line		342	
	option,	337		web2disk	command line option,
	fetch-ebook-metadata	command line		343	
	option,	341			
	web2disk	command line		calibre-debug	command line option,
	option,	343		306	
-e				calibreddb-add	command line option,
	calibre-debug	command line		316	
	306			calibreddb-catalog	command line
	calibre-smtp	command line		option,	319
	option,	312		calibreddb-list_categories	command
	calibreddb-add	command line		line option,	324
	315			ebook-meta	command line option,
	calibreddb-check_library	command		337	
	line option,	323		ebook-polish	command line option,
	ebook-polish	command line		338	
	option,	338		fetch-ebook-metadata	command line
-f			-j	option,	340
	calibre-debug	command line			
	option,	307	-k	ebook-polish	command line option,
	calibre-smtp	command line		338	
	option,	311		ebook-meta	command line option,
	calibreddb-embed_metadata	command		336	
	line option,	325	-l		
	calibreddb-list	command line		calibre-customize	command line
	option,	314		option,	305
	calibreddb-remove_custom_column	command		calibre-smtp	command line option,
	line option,	322		311	
	calibreddb-set_metadata	command line		calibreddb-add	command line option,
	option,	318		316	
	ebook-polish	command line		calibreddb-search	command line
	option,	339		option,	325
	ebook-viewer	command line		calibreddb-set_metadata	command line
	option,	339		option,	318
			-m	ebook-meta	command line option,
				337	

calibre-debug command line option, 306
calibreddb-add command line option, 315
ebook-convert command line option, 335
-n
calibreddb-check_library command line option, 323
web2disk command line option, 343
-o
calibre-smtp command line option, 311
ebook-polish command line option, 338
fetch-ebook-metadata command line option, 340
lrf2lrs command line option, 341
lrs2lrf command line option, 342
-p
calibre-smtp command line option, 312
ebook-meta command line option, 337
ebook-polish command line option, 339
fetch-ebook-metadata command line option, 340
-r
calibre-customize command line option, 305
calibre-debug command line option, 306
calibre-smtp command line option, 312
calibreddb-add command line option, 316
calibreddb-check_library command line option, 323
calibreddb-list_categories command line option, 324
calibreddb-restore_database command line option, 323
ebook-meta command line option, 337
web2disk command line option, 343
-s
calibre command line option, 304
calibre-debug command line option, 307
calibre-smtp command line option, 311
calibreddb-add command line option, 316
calibreddb-catalog command line option, 319
calibreddb-list command line option, 315
ebook-meta command line option, 337
-t
calibre-debug command line option, 306

calibre-smtp command line option, 311
calibreddb-add command line option, 316
calibreddb-list command line option, 315
ebook-meta command line option, 337
fetch-ebook-metadata command line option, 341
web2disk command line option, 343
-u
calibre-smtp command line option, 312
ebook-polish command line option, 339
-v
calibre command line option, 304
calibre-smtp command line option, 311
calibreddb-catalog command line option, 319
ebook-convert command line option, 335
fetch-ebook-metadata command line option, 341
-w
calibre-debug command line option, 307
calibreddb-list command line option, 314
calibreddb-list_categories command line option, 324
-x
calibre-debug command line option, 306

A

abort_article() (*calibre.web.feeds.news.BasicNewsRecipe metoda*), 39
abort_recipe_processing() (*calibre.web.feeds.news.BasicNewsRecipe metoda*), 39
abspath_to_name() (*calibre.ebooks.oeb.polish.container.Container metoda*), 363
accept_drag_move_event() (*calibre.gui2.actions.InterfaceAction metoda*), 284
accept_enter_event() (*calibre.gui2.actions.InterfaceAction metoda*), 284
accepts_drops (*calibre.gui2.actions.InterfaceAction atrybut*), 284
action_add_menu (*calibre.gui2.actions.InterfaceAction atrybut*), 284
action_menu_clone_qaction (*calibre.gui2.actions.InterfaceAction atrybut*), 284

action_shortcut_name
 (calibre.gui2.actions.InterfaceAction atrybut),
 284
 action_spec (calibre.gui2.actions.InterfaceAction
 atrybut), 284
 action_type (calibre.gui2.actions.InterfaceAction
 atrybut), 284
 add_annotation_to_library()
 (calibre.devices.usbms.device.Device metoda),
 281
 add_book() (calibre.devices.interface.BookList metoda),
 277
 add_books() (calibre.db.cache.Cache metoda), 354
 add_books_to_metadata()
 (calibre.devices.interface.DevicePlugin metoda
 klasy), 273
 add_books_to_metadata()
 (calibre.devices.usbms.driver.USBMS metoda),
 282
 add_custom_book_data() (calibre.db.cache.Cache
 metoda), 354
 add_extra_files() (calibre.db.cache.Cache
 metoda), 354
 add_file() (calibre.ebooks.oeb.polish.container.Container
 metoda), 363
 add_format() (calibre.db.cache.Cache metoda), 354
 add_listener() (calibre.db.cache.Cache metoda),
 354
 add_name_to_manifest()
 (calibre.ebooks.oeb.polish.container.Container
 metoda), 363
 add_notes_resource() (calibre.db.cache.Cache
 metoda), 355
 add_or_replace_jacket() (w module
 calibre.ebooks.oeb.polish.jacket), 368
 add_properties() (calibre.ebooks.oeb.polish.container.Container
 metoda), 363
 add_savepoint() (calibre.gui2.tweak_book.boss.Boss
 metoda), 372
 add_toc_thumbnail()
 (calibre.web.feeds.news.BasicNewsRecipe
 metoda), 39
 adeify_images() (calibre.web.feeds.news.BasicNewsRecipe
 metoda klasy), 39
 all_book_ids() (calibre.db.cache.Cache metoda),
 355
 all_field_for() (calibre.db.cache.Cache metoda),
 355
 all_field_ids() (calibre.db.cache.Cache metoda),
 355
 all_field_keys() (calibre.ebooks.metadata.book.base.Metadata
 metoda), 210
 all_field_names() (calibre.db.cache.Cache
 metoda), 355
 all_non_none_fields()
 (calibre.ebooks.metadata.book.base.Metadata
 metoda), 210
 allowed_in_menu (calibre.gui2.tweak_book.plugin.Tool
 atrybut), 371
 allowed_in_toolbar
 (calibre.gui2.tweak_book.plugin.Tool atrybut),
 371
 API, 379
 apply_container_update_to_gui()
 (calibre.gui2.tweak_book.boss.Boss metoda),
 372
 apply_unique_properties()
 (calibre.ebooks.oeb.polish.container.Container
 metoda), 363
 articles_are_obfuscated
 (calibre.web.feeds.news.BasicNewsRecipe
 atrybut), 44
 ASK_TO_ALLOW_CONNECT
 (calibre.devices.interface.DevicePlugin atrybut),
 270
 author (calibre.customize.InterfaceActionBase atrybut),
 286
 author (calibre.customize.MetadataReaderPlugin
 atrybut), 260
 author (calibre.customize.MetadataWriterPlugin
 atrybut), 261
 author (calibre.customize.Plugin atrybut), 257
 author (calibre.customize.PreferencesPlugin atrybut),
 287
 author (calibre.devices.usbms.driver.USBMS atrybut),
 281
 author (calibre.ebooks.metadata.sources.base.Source
 atrybut), 262
 author_data() (calibre.db.cache.Cache metoda), 355
 Container_sort_from_authors()
 (calibre.db.cache.Cache metoda), 355
 auto_cleanup (calibre.web.feeds.news.BasicNewsRecipe
 atrybut), 44
 auto_cleanup_keep
 (calibre.web.feeds.news.BasicNewsRecipe
 atrybut), 44
 auto_repeat (calibre.gui2.actions.InterfaceAction
 atrybut), 284
 auto_trim_covers (calibre.ebooks.metadata.sources.base.Source
 atrybut), 263

B

BACKLOADING_ERROR_MESSAGE
 (calibre.devices.usbms.device.Device atrybut),
 279
 BasicNewsRecipe (klasa w module
 calibre.web.feeds.news), 39
 BCD (calibre.devices.interface.DevicePlugin atrybut), 269

BCD (<i>calibre.devices.usbms.device.Device</i> atrybut), 278	BuiltinCmp (klasa w module <i>calibre.utils.formatter_functions</i>), 204
book_class (<i>calibre.devices.usbms.driver.USBMS</i> atrybut), 281	BuiltinConnectedDeviceName (klasa w module <i>calibre.utils.formatter_functions</i>), 195
book_created (<i>calibre.db.cache.Cache.EventType</i> atrybut), 353	BuiltinConnectedDeviceUUID (klasa w module <i>calibre.utils.formatter_functions</i>), 195
book_edited (<i>calibre.db.cache.Cache.EventType</i> atrybut), 353	BuiltinContains (klasa w module <i>calibre.utils.formatter_functions</i>), 198
book_type (<i>calibre.ebooks.oeb.polish.container.Container</i> atrybut), 364	BuiltinCount (klasa w module <i>calibre.utils.formatter_functions</i>), 200
BookList (klasa w module <i>calibre.devices.interface</i>), 277	BuiltinCurrentLibraryName (klasa w module <i>calibre.utils.formatter_functions</i>), 195
booklist_class (<i>calibre.devices.usbms.driver.USBMS</i> atrybut), 281	BuiltinCurrentLibraryPath (klasa w module <i>calibre.utils.formatter_functions</i>), 196
books() (<i>calibre.devices.interface.DevicePlugin</i> metoda), 273	BuiltinCurrentVirtualLibraryName (klasa w module <i>calibre.utils.formatter_functions</i>), 196
books() (<i>calibre.devices.usbms.driver.USBMS</i> metoda), 281	BuiltinDateArithmetic (klasa w module <i>calibre.utils.formatter_functions</i>), 192
books_for_field() (<i>calibre.db.cache.Cache</i> metoda), 355	BuiltinDaysBetween (klasa w module <i>calibre.utils.formatter_functions</i>), 193
books_in_virtual_library() (<i>calibre.db.cache.Cache</i> metoda), 355	BuiltinDivide (klasa w module <i>calibre.utils.formatter_functions</i>), 191
books_removed (<i>calibre.db.cache.Cache.EventType</i> atrybut), 354	BuiltinEval (klasa w module <i>calibre.utils.formatter_functions</i>), 204
boss (<i>calibre.gui2.tweak_book.plugin.Tool</i> property), 371	BuiltinExtraFileModtime (klasa w module <i>calibre.utils.formatter_functions</i>), 208
Boss (klasa w module <i>calibre.gui2.tweak_book.boss</i>), 372	BuiltinExtraFileNames (klasa w module <i>calibre.utils.formatter_functions</i>), 208
BuiltinAdd (klasa w module <i>calibre.utils.formatter_functions</i>), 190	BuiltinExtraFileSize (klasa w module <i>calibre.utils.formatter_functions</i>), 208
BuiltinAnd (klasa w module <i>calibre.utils.formatter_functions</i>), 192	BuiltinField (klasa w module <i>calibre.utils.formatter_functions</i>), 196
BuiltinAnnotationCount (klasa w module <i>calibre.utils.formatter_functions</i>), 194	BuiltinFieldExists (klasa w module <i>calibre.utils.formatter_functions</i>), 198
BuiltinApproximateFormats (klasa w module <i>calibre.utils.formatter_functions</i>), 194	BuiltinFinishFormatting (klasa w module <i>calibre.utils.formatter_functions</i>), 193
BuiltinArguments (klasa w module <i>calibre.utils.formatter_functions</i>), 203	BuiltinFirstMatchingCmp (klasa w module <i>calibre.utils.formatter_functions</i>), 205
BuiltinAssign (klasa w module <i>calibre.utils.formatter_functions</i>), 203	BuiltinFirstNonEmpty (klasa w module <i>calibre.utils.formatter_functions</i>), 198
BuiltinAuthorLinks (klasa w module <i>calibre.utils.formatter_functions</i>), 195	BuiltinFloor (klasa w module <i>calibre.utils.formatter_functions</i>), 191
BuiltinAuthorSorts (klasa w module <i>calibre.utils.formatter_functions</i>), 195	BuiltinFormatDate (klasa w module <i>calibre.utils.formatter_functions</i>), 193
BuiltinBookCount (klasa w module <i>calibre.utils.formatter_functions</i>), 208	BuiltinFormatDateField (klasa w module <i>calibre.utils.formatter_functions</i>), 194
BuiltinBooksize (klasa w module <i>calibre.utils.formatter_functions</i>), 195	BuiltinFormatNumber (klasa w module <i>calibre.utils.formatter_functions</i>), 194
BuiltinBookValues (klasa w module <i>calibre.utils.formatter_functions</i>), 208	BuiltinFormatsModtimes (klasa w module <i>calibre.utils.formatter_functions</i>), 196
BuiltinCapitalize (klasa w module <i>calibre.utils.formatter_functions</i>), 205	BuiltinFormatsPaths (klasa w module <i>calibre.utils.formatter_functions</i>), 196
BuiltinCeiling (klasa w module <i>calibre.utils.formatter_functions</i>), 191	BuiltinFormatsSizes (klasa w module <i>calibre.utils.formatter_functions</i>), 196
BuiltinCharacter (klasa w module <i>calibre.utils.formatter_functions</i>), 206	
BuiltinCheckYesNo (klasa w module <i>calibre.utils.formatter_functions</i>), 198	

BuiltinFractionalPart	(klasa w module	BuiltinMod	(klasa w module
calibre.utils.formatter_functions), 191		calibre.utils.formatter_functions), 191	
BuiltinGetLink	(klasa w module	BuiltinMultiply	(klasa w module
calibre.utils.formatter_functions), 209		calibre.utils.formatter_functions), 191	
BuiltinGetNote	(klasa w module	BuiltinNot	(klasa w module
calibre.utils.formatter_functions), 209		calibre.utils.formatter_functions), 192	
BuiltinGlobals	(klasa w module	BuiltinOndevice	(klasa w module
calibre.utils.formatter_functions), 204		calibre.utils.formatter_functions), 197	
BuiltinHasCover	(klasa w module	BuiltinOr	(klasa w module
calibre.utils.formatter_functions), 197		calibre.utils.formatter_functions), 192	
BuiltinHasExtraFiles	(klasa w module	BuiltinPrint	(klasa w module
calibre.utils.formatter_functions), 209		calibre.utils.formatter_functions), 204	
BuiltinHasNote	(klasa w module	BuiltinRange	(klasa w module
calibre.utils.formatter_functions), 209		calibre.utils.formatter_functions), 202	
BuiltinHumanReadable	(klasa w module	BuiltinRe	(klasa w module
calibre.utils.formatter_functions), 194		calibre.utils.formatter_functions), 206	
BuiltinIdentifierInList	(klasa w module	BuiltinReGroup	(klasa w module
calibre.utils.formatter_functions), 199		calibre.utils.formatter_functions), 206	
BuiltinIfempty	(klasa w module	BuiltinRound	(klasa w module
calibre.utils.formatter_functions), 198		calibre.utils.formatter_functions), 192	
BuiltinInList	(klasa w module	BuiltinSeriesSort	(klasa w module
calibre.utils.formatter_functions), 199		calibre.utils.formatter_functions), 197	
BuiltinIsMarked	(klasa w module	BuiltinSetGlobals	(klasa w module
calibre.utils.formatter_functions), 197		calibre.utils.formatter_functions), 209	
BuiltinLanguageCodes	(klasa w module	BuiltinShorten	(klasa w module
calibre.utils.formatter_functions), 197		calibre.utils.formatter_functions), 206	
BuiltinListCountMatching	(klasa w module	BuiltinStrcatMax	(klasa w module
calibre.utils.formatter_functions), 200		calibre.utils.formatter_functions), 207	
BuiltinListDifference	(klasa w module	BuiltinStrcmp	(klasa w module
calibre.utils.formatter_functions), 200		calibre.utils.formatter_functions), 205	
BuiltinListEquals	(klasa w module	BuiltinStrcmpcase	(klasa w module
calibre.utils.formatter_functions), 201		calibre.utils.formatter_functions), 205	
BuiltinListIntersection	(klasa w module	BuiltinStrInList	(klasa w module
calibre.utils.formatter_functions), 201		calibre.utils.formatter_functions), 200	
BuiltinListitem	(klasa w module	BuiltinStrlen	(klasa w module
calibre.utils.formatter_functions), 200		calibre.utils.formatter_functions), 207	
BuiltinListJoin	(klasa w module	BuiltinSubitems	(klasa w module
calibre.utils.formatter_functions), 201		calibre.utils.formatter_functions), 203	
BuiltinListRe	(klasa w module	BuiltinSublist	(klasa w module
calibre.utils.formatter_functions), 201		calibre.utils.formatter_functions), 203	
BuiltinListReGroup	(klasa w module	BuiltinSubstr	(klasa w module
calibre.utils.formatter_functions), 202		calibre.utils.formatter_functions), 207	
BuiltinListRemoveDuplicates	(klasa w module	BuiltinSubtract	(klasa w module
calibre.utils.formatter_functions), 202		calibre.utils.formatter_functions), 191	
BuiltinListSort	(klasa w module	BuiltinSwapAroundArticles	(klasa w module
calibre.utils.formatter_functions), 202		calibre.utils.formatter_functions), 207	
BuiltinListSplit	(klasa w module	BuiltinSwapAroundComma	(klasa w module
calibre.utils.formatter_functions), 202		calibre.utils.formatter_functions), 207	
BuiltinListUnion	(klasa w module	BuiltinSwitch	(klasa w module
calibre.utils.formatter_functions), 202		calibre.utils.formatter_functions), 199	
BuiltinLookup	(klasa w module	BuiltinSwitchIf	(klasa w module
calibre.utils.formatter_functions), 199		calibre.utils.formatter_functions), 199	
BuiltinLowercase	(klasa w module	BuiltinTemplate	(klasa w module
calibre.utils.formatter_functions), 205		calibre.utils.formatter_functions), 204	

BuiltinTest	(klasa w module <i>calibre.utils.formatter_functions</i>), 198	module, 368
BuiltinTitlecase	(klasa w module <i>calibre.utils.formatter_functions</i>), 205	module, 368
BuiltinToday	(klasa w module <i>calibre.utils.formatter_functions</i>), 193	module, 367
BuiltinToHex	(klasa w module <i>calibre.utils.formatter_functions</i>), 207	module, 368
BuiltinTransliterate	(klasa w module <i>calibre.utils.formatter_functions</i>), 208	module, 370
BuiltinUppercase	(klasa w module <i>calibre.utils.formatter_functions</i>), 206	module, 372
BuiltinUserCategories	(klasa w module <i>calibre.utils.formatter_functions</i>), 197	module, 190
C		calibre.web.feeds.news module, 39
Cache	(klasa w module <i>calibre.db.cache</i>), 353	calibre-customize command line option
Cache.EventType	(klasa w module <i>calibre.db.cache</i>), 353	--add-plugin, 304
cached_cover_url_is_reliable	(<i>calibre.ebooks.metadata.sources.base.Source</i> atrybut), 262	--build-plugin, 304
calibre command line option		--customize-plugin, 304
--detach, 304		--disable-plugin, 304
--help, 304		--enable-plugin, 304
--ignore-plugins, 304		--help, 305
--no-update-check, 304		--list-plugins, 305
--shutdown-running-calibre, 304		--remove-plugin, 305
--start-in-tray, 304		--version, 305
--verbose, 304		-a, 304
--version, 304		-b, 304
--with-library, 304		-h, 305
-h, 304		-l, 305
-s, 304		-r, 305
-v, 304		calibre-debug command line option
calibre.customize	module, 256	--add-simple-plugin, 305
calibre.customize.conversion	module, 266	--command, 305
calibre.db.cache	module, 353	--debug-device-driver, 305
calibre.devices.interface	module, 269	--default-programs, 305
calibre.ebooks.metadata.book.base	module, 209	--diff, 305
calibre.ebooks.metadata.sources.base	module, 262	--edit-book, 305
calibre.ebooks.oeb.polish.container	module, 363	--exec-file, 306
calibre.ebooks.oeb.polish.cover	module, 369	--explode-book, 306
calibre.ebooks.oeb.polish.css	module, 369	--export-all-calibre-data, 306
calibre.ebooks.oeb.polish.jacket		--fix-multiprocessing, 306
		--gui, 306
		--gui-debug, 306
		--help, 306
		--implode-book, 306
		--import-calibre-data, 306
		--inspect-mobi, 306
		--paths, 306
		--run-plugin, 306
		--run-test, 306
		--run-without-debug, 306
		--shutdown-running-calibre, 307
		--subset-font, 307
		--test-build, 307

```

--version, 307
--viewer, 307
-c, 305
-d, 305
-e, 306
-f, 307
-g, 306
-h, 306
-i, 306
-m, 306
-r, 306
-s, 307
-t, 306
-w, 307
-x, 306
calibre-server command line option
--access-log, 307
--ajax-timeout, 307
--auth-mode, 307
--auto-reload, 307
--ban-after, 307
--ban-for, 307
--book-list-mode, 308
--compress-min-size, 308
--custom-list-template, 308
--daemonize, 308
--disable-allow-socket-
    -preallocation, 308
--disable-auth, 308
--disable-fallback-to-detected-
    -interface, 308
--disable-local-write, 308
--disable-log-not-found, 308
--disable-use-bonjour, 308
--disable-use-sendfile, 309
--displayed-fields, 308
--enable-allow-socket-
    -preallocation, 308
--enable-auth, 308
--enable-fallback-to-detected-
    -interface, 308
--enable-local-write, 308
--enable-log-not-found, 308
--enable-use-bonjour, 308
--enable-use-sendfile, 309
--help, 309
--ignored-fields, 309
--listen-on, 309
--log, 309
--manage-users, 309
--max-header-line-size, 309
--max-job-time, 309
--max-jobs, 309
--max-log-size, 309
--max-opds-items, 309
--max-opds-ungrouped-items, 309
--max-request-body-size, 309
--num-per-page, 310
--pidfile, 310
--port, 310
--search-the-net-urls, 310
--shutdown-timeout, 310
--ssl-certfile, 310
--ssl-keyfile, 310
--timeout, 310
--trusted-ips, 310
--url-prefix, 310
--userdb, 310
--version, 310
--worker-count, 310
-h, 309
calibre-smtp command line option
--attachment, 311
--cafile, 312
--dont-verify-server-certificate,
    312
--encryption-method, 312
--fork, 311
--help, 311
--localhost, 311
--outbox, 311
--password, 312
--port, 312
--relay, 312
--subject, 311
--timeout, 311
--username, 312
--verbose, 311
--version, 311
-a, 311
-e, 312
-f, 311
-h, 311
-l, 311
-o, 311
-p, 312
-r, 312
-s, 311
-t, 311
-u, 312
-v, 311
calibreddb-add command line option
-l, 316
-I, 316
-S, 316
-T, 316
--add, 316
--authors, 315

```

```
--automerger, 315
--cover, 315
--duplicates, 315
--empty, 315
--identifier, 316
--ignore, 316
--isbn, 316
--languages, 316
--one-book-per-directory, 316
--recurse, 316
--series, 316
--series-index, 316
--tags, 316
--title, 316
-a, 315
-c, 315
-d, 315
-e, 315
-i, 316
-l, 316
-m, 315
-r, 316
-s, 316
-t, 316
calibredb-add_custom_column    command
    line option
    --display, 322
    --is-multiple, 322
calibredb-add_format           command    line
    option
    --as-extra-data-file, 317
    --dont-replace, 317
calibredb-backup_metadata      command line
    option
    --all, 324
calibredb-catalog              command line option
    --catalog-title, 320
    --cross-reference-authors, 320
    --debug-pipeline, 320
    --exclude-genre, 320
    --exclusion-rules, 320
    --generate-authors, 320
    --generate-descriptions, 320
    --generate-genres, 320
    --generate-recently-added, 320
    --generate-series, 320
    --generate-titles, 320
    --genre-source-field, 320
    --header-note-source-field, 320
    --ids, 319
    --merge-comments-rule, 320
    --output-profile, 321
    --prefix-rules, 321
    --preset, 321
    --search, 319
    --thumb-width, 321
    --use-existing-cover, 321
    --verbose, 319
    -i, 319
    -s, 319
    -v, 319
calibredb-check_library        command    line
    option
    --csv, 323
    --ignore_extensions, 323
    --ignore_names, 323
    --report, 323
    --vacuum-fts-db, 323
    -c, 323
    -e, 323
    -n, 323
    -r, 323
calibredb-custom_columns       command line
    option
    --details, 322
    -d, 322
calibredb-embed_metadata       command line
    option
    --only-formats, 325
    -f, 325
calibredb-export               command line option
    --all, 318
    --dont-asciiize, 318
    --dont-save-cover, 318
    --dont-save-extra-files, 318
    --dont-update-metadata, 318
    --dont-write-opf, 318
    --formats, 319
    --progress, 319
    --replace-whitespace, 319
    --single-dir, 319
    --template, 319
    --timefmt, 319
    --to-dir, 319
    --to-lowercase, 319
calibredb-fts_index            command line option
    --indexing-speed, 325
    --wait-for-completion, 325
calibredb-fts_search           command    line
    option
    --do-not-match-on-related-words, 326
    --include-snippets, 326
    --indexing-threshold, 326
    --match-end-marker, 326
    --match-start-marker, 326
    --output-format, 326
    --restrict-to, 326
calibredb-list                 command line option
```

```

--ascending, 314
--fields, 314
--for-machine, 314
--limit, 314
--line-width, 314
--prefix, 315
--search, 315
--separator, 315
--sort-by, 315
--template, 315
--template_file, 315
--template_heading, 315
-f, 314
-s, 315
-t, 315
-w, 314
calibredb-list_categories command line
    option
    --categories, 324
    --csv, 324
    --dialect, 324
    --item_count, 324
    --width, 324
    -c, 324
    -i, 324
    -r, 324
    -w, 324
calibredb-remove command line option
    --permanent, 317
calibredb-remove_custom_column command
    line option
    --force, 322
    -f, 322
calibredb-restore_database command
    line option
    --really-do-it, 323
    -r, 323
calibredb-search command line option
    --limit, 325
    -l, 325
calibredb-set_custom command line
    option
    --append, 323
    -a, 323
calibredb-set_metadata command line
    option
    --field, 318
    --list-fields, 318
    -f, 318
    -l, 318
calibredb-show_metadata command line
    option
    --as-opf, 317
    can_be_disabled(calibre.customize.conversion.InputFormatPlugin
        atrybut), 266
    can_be_disabled(calibre.customize.conversion.OutputFormatPlugin
        atrybut), 267
    can_be_disabled(calibre.customize.InterfaceActionBase
        atrybut), 286
    can_be_disabled(calibre.customize.Plugin atrybut),
        257
    can_be_disabled(calibre.customize.PreferencesPlugin
        atrybut), 287
    CAN_DO_DEVICE_DB_PLUGBOARD
        (calibre.devices.interface.DevicePlugin atrybut),
        269
    can_get_multiple_covers
        (calibre.ebooks.metadata.sources.base.Source
        atrybut), 263
    can_handle() (calibre.devices.interface.DevicePlugin
        metoda), 271
    can_handle_windows()
        (calibre.devices.interface.DevicePlugin metoda),
        271
    can_handle_windows()
        (calibre.devices.usbms.device.Device metoda),
        280
    CAN_SET_METADATA(calibre.devices.interface.DevicePlugin
        atrybut), 269
    CAN_SET_METADATA(calibre.devices.usbms.driver.USBMS
        atrybut), 281
    canonicalize_internal_url()
        (calibre.web.feeds.news.BasicNewsRecipe
        metoda), 40
    capabilities(calibre.ebooks.metadata.sources.base.Source
        atrybut), 262
    card_prefix() (calibre.devices.interface.DevicePlugin
        metoda), 272
    card_prefix() (calibre.devices.usbms.device.Device
        metoda), 279
    CatalogPlugin (klasa w module calibre.customize),
        261
    category (calibre.customize.PreferencesPlugin atrybut),
        287
    category_order (calibre.customize.PreferencesPlugin
        atrybut), 287
    center_navbar(calibre.web.feeds.news.BasicNewsRecipe
        atrybut), 44
    change_font() (w module
        calibre.ebooks.oeb.polish.fonts), 369
    changed_signal(calibre.gui2.preferences.ConfigWidgetBase
        atrybut), 288
    changed_signal(calibre.gui2.preferences.ConfigWidgetInterface
        atrybut), 287
    clean_downloaded_metadata()
        (calibre.ebooks.metadata.sources.base.Source
        metoda), 264

```

cleanup() (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 40

CLI (*klasa w module calibre.devices.usbms.cli*), 281

cli_main() (*calibre.customize.Plugin* metoda), 258

cli_options (*calibre.customize.CatalogPlugin* atrybut), 261

clone_browser() (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 40

close_editor() (*calibre.gui2.tweak_book.boss.Boss* metoda), 372

command line option

- help, 314
- library-path, 314
- password, 314
- timeout, 314
- username, 314
- version, 314
- with-library, 314
- h, 314

commit() (*calibre.ebooks.oeb.polish.container.Container* metoda), 364

commit() (*calibre.gui2.preferences.ConfigWidgetBase* metoda), 289

commit() (*calibre.gui2.preferences.ConfigWidgetInterface* metoda), 288

commit_all_editors_to_container() (*calibre.gui2.tweak_book.boss.Boss* metoda), 372

commit_item() (*calibre.ebooks.oeb.polish.container.Container* metoda), 364

common_options (*calibre.customize.conversion.InputFormatPlugin* atrybut), 266

common_options (*calibre.customize.conversion.OutputFormatPlugin* atrybut), 268

compress_covers() (*calibre.db.cache.Cache* metoda), 355

compress_news_images (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 44

compress_news_images_auto_size (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 44

compress_news_images_max_size (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 44

config_help_message (*calibre.ebooks.metadata.sources.base.Source* atrybut), 263

config_widget (*calibre.customize.PreferencesPlugin* atrybut), 287

config_widget() (*calibre.customize.Plugin* metoda), 257

config_widget() (*calibre.devices.interface.DevicePlugin* metoda klasy), 274

config_widget() (*calibre.ebooks.metadata.sources.base.Source* metoda), 263

ConfigWidgetBase (*klasa w module calibre.gui2.preferences*), 288

ConfigWidgetInterface (*klasa w module calibre.gui2.preferences*), 287

Container (*klasa w module calibre.ebooks.oeb.polish.container*), 363

contains(), 161

conversion_options (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 44

convert() (*calibre.customize.conversion.InputFormatPlugin* metoda), 267

convert() (*calibre.customize.conversion.OutputFormatPlugin* metoda), 268

copy_cover_to() (*calibre.db.cache.Cache* metoda), 355

copy_format_to() (*calibre.db.cache.Cache* metoda), 355

core_usage (*calibre.customize.conversion.InputFormatPlugin* atrybut), 266

cover() (*calibre.db.cache.Cache* metoda), 355

cover_margins (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 45

create_action() (*calibre.gui2.tweak_book.plugin.Tool* metoda), 371

create_inline_toc() (*w module calibre.ebooks.oeb.polish.toc*), 370

create_menu_action() (*calibre.gui2.actions.InterfaceAction* metoda), 284

create_widget() (*calibre.customize.PreferencesPlugin* metoda), 287

CSS, 379

current_container (*calibre.gui2.tweak_book.plugin.Tool* property), 371

currently_editing (*calibre.gui2.tweak_book.boss.Boss* property), 372

custom_field_keys() (*calibre.ebooks.metadata.book.base.Metadata* metoda), 210

customization_help() (*calibre.customize.Plugin* metoda), 258

customization_help() (*calibre.ebooks.metadata.sources.base.Source* metoda), 263

D

data_for_find_identical_books() (*calibre.db.cache.Cache* metoda), 356

`data_for_has_book()` (*calibre.db.cache.Cache metoda*), 356
`debug_managed_device_detection()` (*calibre.devices.interface.DevicePlugin metoda*), 271
`deepcopy()` (*calibre.ebooks.metadata.book.base.Metadata metoda*), 210
`default_cover()` (*calibre.web.feeds.news.BasicNewsRecipe metoda*), 40
`delay` (*calibre.web.feeds.news.BasicNewsRecipe atrybut*), 45
`delete_books()` (*calibre.devices.interface.DevicePlugin metoda*), 274
`delete_books()` (*calibre.devices.usbms.driver.USBMS metoda*), 282
`delete_custom_book_data()` (*calibre.db.cache.Cache metoda*), 356
`delete_trash_entry()` (*calibre.db.cache.Cache metoda*), 356
`description` (*calibre.customize.conversion.OutputFormatPlugin property*), 268
`description` (*calibre.customize.Plugin atrybut*), 256
`description` (*calibre.customize.PreferencesPlugin atrybut*), 287
`description` (*calibre.devices.usbms.driver.USBMS atrybut*), 281
`description` (*calibre.web.feeds.news.BasicNewsRecipe atrybut*), 45
`detect_managed_devices()` (*calibre.devices.interface.DevicePlugin metoda*), 271
`Device` (*klasa w module calibre.devices.usbms.device*), 278
`DevicePlugin` (*klasa w module calibre.devices.interface*), 269
`dirty()` (*calibre.ebooks.oeb.polish.container.Container metoda*), 364
`do_user_config()` (*calibre.customize.Plugin metoda*), 257
`dont_add_to` (*calibre.gui2.actions.InterfaceAction atrybut*), 284
`dont_remove_from` (*calibre.gui2.actions.InterfaceAction atrybut*), 284
`download()` (*calibre.web.feeds.news.BasicNewsRecipe metoda*), 40
`download_cover()` (*calibre.ebooks.metadata.sources.base.Source metoda*), 265
`drop_event()` (*calibre.gui2.actions.InterfaceAction metoda*), 284

E

`ebook-convert` command line option
`--asciiize`, 329
`--author-sort`, 334
`--authors`, 334
`--base-font-size`, 329
`--book-producer`, 334
`--change-justification`, 329
`--chapter`, 333
`--chapter-mark`, 333
`--comments`, 334
`--cover`, 335
`--debug-pipeline`, 335
`--disable-dehyphenate`, 332
`--disable-delete-blank-paragraphs`, 332
`--disable-fix-indents`, 332
`--disable-font-rescaling`, 329
`--disable-format-scene-breaks`, 332
`--disable-italicize-common-cases`, 332
`--disable-markup-chapter-headings`, 332
`--disable-remove-fake-margins`, 333
`--disable-renumber-headings`, 332
`--disable-unwrap-lines`, 332
`--dont-split-on-page-breaks`, 328
`--duplicate-links-in-toc`, 334
`--embed-all-fonts`, 329
`--embed-font-family`, 329
`--enable-heuristics`, 332
`--epub-flatten`, 328
`--epub-inline-toc`, 328
`--epub-max-image-size`, 328
`--epub-toc-at-end`, 328
`--epub-version`, 328
`--expand-css`, 329
`--extra-css`, 330
`--extract-to`, 328
`--filter-css`, 330
`--flow-size`, 328
`--font-size-mapping`, 330
`--from-opf`, 335
`--help`, 327
`--html-unwrap-factor`, 332
`--input-encoding`, 328
`--input-profile`, 327
`--insert-blank-line`, 330
`--insert-blank-line-size`, 330
`--insert-metadata`, 333
`--isbn`, 335
`--keep-ligatures`, 330
`--language`, 335
`--level1-toc`, 334
`--level2-toc`, 334
`--level3-toc`, 334
`--line-height`, 330
`--linearize-tables`, 330

```

--list-recipes, 327
--margin-bottom, 330
--margin-left, 330
--margin-right, 330
--margin-top, 331
--max-toc-links, 334
--minimum-line-height, 331
--no-chapters-in-toc, 334
--no-default-epub-cover, 328
--no-svg-cover, 328
--output-profile, 327
--page-breaks-before, 333
--prefer-metadata-cover, 333
--preserve-cover-aspect-ratio, 329
--pretty-print, 329
--pubdate, 335
--publisher, 335
--rating, 335
--read-metadata-from-opf, 335
--remove-first-image, 333
--remove-paragraph-spacing, 331
--remove-paragraph-spacing-indent-
    -size, 331
--replace-scene-breaks, 332
--search-replace, 332
--series, 335
--series-index, 335
--smarten-punctuation, 331
--sr1-replace, 332
--sr1-search, 332
--sr2-replace, 333
--sr2-search, 333
--sr3-replace, 333
--sr3-search, 333
--start-reading-at, 333
--subset-embedded-fonts, 331
--tags, 335
--timestamp, 335
--title, 335
--title-sort, 335
--toc-filter, 334
--toc-threshold, 334
--toc-title, 329
--transform-css-rules, 331
--transform-html-rules, 331
--unsmarten-punctuation, 331
--use-auto-toc, 334
--verbose, 335
--version, 327
-d, 335
-h, 327
-m, 335
-v, 335
ebook-edit command line option
    --detach, 336
    --help, 336
    --select-text, 336
    --version, 336
    -h, 336
ebook-meta command line option
    --author-sort, 336
    --authors, 336
    --book-producer, 336
    --category, 336
    --comments, 337
    --cover, 337
    --date, 337
    --from-opf, 337
    --get-cover, 337
    --help, 337
    --identifier, 337
    --index, 337
    --isbn, 337
    --language, 337
    --lrf-bookid, 337
    --publisher, 337
    --rating, 337
    --series, 337
    --tags, 337
    --title, 337
    --title-sort, 337
    --to-opf, 337
    --version, 338
    -a, 336
    -c, 337
    -d, 337
    -h, 337
    -i, 337
    -k, 336
    -l, 337
    -p, 337
    -r, 337
    -s, 337
    -t, 337
ebook-polish command line option
    -H, 338
    -U, 339
    --add-soft-hyphens, 338
    --compress-images, 338
    --cover, 338
    --embed-fonts, 338
    --help, 338
    --jacket, 338
    --opf, 338
    --remove-jacket, 338
    --remove-soft-hyphens, 338
    --remove-unused-css, 339
    --smarten-punctuation, 339

```



```

--subset-fonts, 339
--upgrade-book, 339
--verbose, 339
--version, 339
-c, 338
-e, 338
-f, 339
-h, 338
-i, 338
-j, 338
-o, 338
-p, 339
-u, 339
ebook-viewer command line option
--continue, 339
--detach, 339
--force-reload, 339
--full-screen, 339
--fullscreen, 339
--help, 339
--new-instance, 340
--open-at, 340
--raise-window, 340
--version, 340
-f, 339
-h, 339
edit_file() (calibre.gui2.tweak_book.boss.Boss
metoda), 372
eject() (calibre.devices.interface.DevicePlugin metoda),
272
eject() (calibre.devices.usbms.device.Device metoda),
280
embed_metadata() (calibre.db.cache.Cache metoda),
356
encoding (calibre.web.feeds.news.BasicNewsRecipe
atrybut), 45
exists() (calibre.ebooks.oeb.polish.container.Container
metoda), 364
expire_old_trash() (calibre.db.cache.Cache
metoda), 356
export_note() (calibre.db.cache.Cache metoda), 356
extra_css (calibre.web.feeds.news.BasicNewsRecipe
atrybut), 45
extract_readable_article()
(calibre.web.feeds.news.BasicNewsRecipe
metoda), 40
F
fast_field_for() (calibre.db.cache.Cache metoda),
356
feeds (calibre.web.feeds.news.BasicNewsRecipe atrybut),
45
fetch-ebook-metadata command line
option
-I, 340
--allowed-plugin, 340
--authors, 340
--cover, 340
--help, 340
--identifier, 340
--isbn, 340
--opf, 340
--timeout, 341
--title, 341
--verbose, 341
--version, 341
-a, 340
-c, 340
-d, 341
-h, 340
-i, 340
-o, 340
-p, 340
-t, 341
-v, 341
field_for() (calibre.db.cache.Cache metoda), 356
field_ids_for() (calibre.db.cache.Cache metoda),
356
field_supports_notes() (calibre.db.cache.Cache
metoda), 356
file_type (calibre.customize.conversion.OutputFormatPlugin
atrybut), 268
file_types (calibre.customize.CatalogPlugin atrybut),
261
file_types (calibre.customize.conversion.InputFormatPlugin
atrybut), 266
file_types (calibre.customize.FileTypePlugin atrybut),
258
file_types (calibre.customize.MetadataReaderPlugin
atrybut), 260
file_types (calibre.customize.MetadataWriterPlugin
atrybut), 261
filename_callback()
(calibre.devices.usbms.device.Device metoda),
280
filesize() (calibre.ebooks.oeb.polish.container.Container
metoda), 364
FileTypePlugin (klasa w module calibre.customize),
258
filter_css() (w module
calibre.ebooks.oeb.polish.css), 370
filter_regexp (calibre.web.feeds.news.BasicNewsRecipe
atrybut), 45
find_identical_books() (calibre.db.cache.Cache
metoda), 356
fix_all_html() (w module
calibre.ebooks.oeb.polish.pretty), 368
fix_html() (w module calibre.ebooks.oeb.polish.pretty),

```

- 368
- `for_viewer()` (*calibre.customize.conversion.InputFormatPlugin* atrybut), 266
- `format()` (*calibre.db.cache.Cache* metoda), 357
- `format_abspath()` (*calibre.db.cache.Cache* metoda), 357
- `format_added` (*calibre.db.cache.Cache.EventType* atrybut), 354
- `format_field()` (*calibre.ebooks.metadata.book.base.Metadata* metoda), 211
- `format_hash()` (*calibre.db.cache.Cache* metoda), 357
- `format_metadata()` (*calibre.db.cache.Cache* metoda), 357
- `FORMATS` (*calibre.devices.interface.DevicePlugin* atrybut), 269
- `FORMATS` (*calibre.devices.usbms.driver.USBMS* atrybut), 281
- `formats()` (*calibre.db.cache.Cache* metoda), 357
- `formats_removed` (*calibre.db.cache.Cache.EventType* atrybut), 354
- `free_space()` (*calibre.devices.interface.DevicePlugin* metoda), 273
- `free_space()` (*calibre.devices.usbms.device.Device* metoda), 279
- `from_files()` (w *calibre.ebooks.oeb.polish.toc* module), 370
- `from_links()` (w *calibre.ebooks.oeb.polish.toc* module), 370
- `from_xpaths()` (w *calibre.ebooks.oeb.polish.toc* module), 370
- ## G
- `generate_item()` (*calibre.ebooks.oeb.polish.container.Container* metoda), 364
- `genesis()` (*calibre.gui2.actions.InterfaceAction* metoda), 285
- `genesis()` (*calibre.gui2.preferences.ConfigWidgetInterface* metoda), 288
- `get_all_items_that_have_notes()` (*calibre.db.cache.Cache* metoda), 357
- `get_all_link_maps_for_book()` (*calibre.db.cache.Cache* metoda), 357
- `get_all_standard_metadata()` (*calibre.ebooks.metadata.book.base.Metadata* metoda), 210
- `get_all_user_metadata()` (*calibre.ebooks.metadata.book.base.Metadata* metoda), 210
- `get_annotations()` (*calibre.devices.usbms.device.Device* metoda), 281
- `get_article_url()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 40
- `get_author_tokens()` (*calibre.ebooks.metadata.sources.base.Source* metoda), 263
- `get_book_url()` (*calibre.ebooks.metadata.sources.base.Source* metoda), 264
- `get_book_url_name()` (*calibre.ebooks.metadata.sources.base.Source* metoda), 264
- `get_book_urls()` (*calibre.ebooks.metadata.sources.base.Source* metoda), 264
- `get_browser()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 40
- `get_cached_cover_url()` (*calibre.ebooks.metadata.sources.base.Source* metoda), 264
- `get_categories()` (*calibre.db.cache.Cache* metoda), 358
- `get_collections()` (*calibre.devices.interface.BookList* metoda), 277
- `get_cover_url()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 41
- `get_custom_book_data()` (*calibre.db.cache.Cache* metoda), 358
- `get_device_information()` (*calibre.devices.interface.DevicePlugin* metoda), 272
- `get_device_information()` (*calibre.devices.usbms.driver.USBMS* metoda), 281
- `get_device_uid()` (*calibre.devices.interface.DevicePlugin* metoda), 275
- `get_driveinfo()` (*calibre.devices.interface.DevicePlugin* metoda), 272
- `get_extra_css()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 41
- `get_feeds()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 41
- `get_file()` (*calibre.devices.interface.DevicePlugin* metoda), 274
- `get_file_path_for_processing()` (*calibre.ebooks.oeb.polish.container.Container* metoda), 364
- `get_id_map()` (*calibre.db.cache.Cache* metoda), 358
- `get_identifiers()` (*calibre.ebooks.metadata.book.base.Metadata* metoda), 210
- `get_ids_for_custom_book_data()` (*calibre.db.cache.Cache* metoda), 358
- `get_images()` (*calibre.customize.conversion.InputFormatPlugin* metoda), 267
- `get_item_id()` (*calibre.db.cache.Cache* metoda), 358
- `get_item_ids()` (*calibre.db.cache.Cache* metoda), 358

`get_item_name()` (*calibre.db.cache.Cache* metoda), 358
`get_item_name_map()` (*calibre.db.cache.Cache* metoda), 358
`get_link_map()` (*calibre.db.cache.Cache* metoda), 358
`get_masthead_title()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 41
`get_masthead_url()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 41
`get_metadata()` (*calibre.customize.MetadataReaderPlugin* metoda), 260
`get_metadata()` (*calibre.db.cache.Cache* metoda), 358
`get_next_series_num_for()` (*calibre.db.cache.Cache* metoda), 358
`get_notes_resource()` (*calibre.db.cache.Cache* metoda), 358
`get_obfuscated_article()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 41
`get_open_popup_message()` (*calibre.devices.interface.DevicePlugin* metoda klasy), 270
`get_option()` (*calibre.devices.interface.DevicePlugin* metoda), 276
`get_proxy_metadata()` (*calibre.db.cache.Cache* metoda), 359
`get_recommended_folders()` (w module *calibre.ebooks.oeb.polish.replace*), 367
`get_standard_metadata()` (*calibre.ebooks.metadata.book.base.Metadata* metoda), 210
`get_title_tokens()` (*calibre.ebooks.metadata.sources.base.Source* metoda), 263
`get_url_specific_delay()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 41
`get_usage_count_by_id()` (*calibre.db.cache.Cache* metoda), 359
`get_user_blacklisted_devices()` (*calibre.devices.interface.DevicePlugin* metoda), 275
`get_user_metadata()` (*calibre.ebooks.metadata.book.base.Metadata* metoda), 210
`gui` (*calibre.gui2.tweak_book.plugin.Tool* property), 371
`gui_category` (*calibre.customize.PreferencesPlugin* atrybut), 287
`gui_configuration_widget()` (*calibre.customize.conversion.InputFormatPlugin* metoda), 267
`gui_configuration_widget()` (*calibre.customize.conversion.OutputFormatPlugin* metoda), 268
`gui_layout_complete()` (*calibre.gui2.actions.InterfaceAction* metoda), 286
`gui_name` (*calibre.customize.PreferencesPlugin* atrybut), 287
`guide_type_map` (*calibre.ebooks.oeb.polish.container.Container* property), 364
H
`handle_gzip` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 45
`has_book()` (*calibre.db.cache.Cache* metoda), 359
`has_format()` (*calibre.db.cache.Cache* metoda), 359
`has_html_comments` (*calibre.ebooks.metadata.sources.base.Source* atrybut), 262
`has_id()` (*calibre.db.cache.Cache* metoda), 359
`has_name()` (*calibre.ebooks.oeb.polish.container.Container* metoda), 364
`href_to_name()` (*calibre.ebooks.oeb.polish.container.Container* metoda), 364
HTML, 379
I
`icon` (*calibre.customize.PreferencesPlugin* atrybut), 287
`icon` (*calibre.devices.interface.DevicePlugin* atrybut), 270
`id_from_url()` (*calibre.ebooks.metadata.sources.base.Source* metoda), 264
`identify()` (*calibre.ebooks.metadata.sources.base.Source* metoda), 264
`identify_results_keygen()` (*calibre.ebooks.metadata.sources.base.Source* metoda), 264
`ignore_connected_device()` (*calibre.devices.interface.DevicePlugin* metoda), 275
`ignore_duplicate_articles` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 45
`ignore_ssl_errors` (*calibre.ebooks.metadata.sources.base.Source* atrybut), 262
`image_url_processor()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda klasy), 41
`import_note()` (*calibre.db.cache.Cache* metoda), 359
`index_to_soup()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 41
`indexing_progress_changed` (*calibre.db.cache.Cache.EventType* atrybut),

- 354
- `init()` (*calibre.db.cache.Cache* metoda), 359
- `initial_tab_changed()`
(*calibre.gui2.preferences.ConfigWidgetInterface* metoda), 288
- `initialization_complete()`
(*calibre.gui2.actions.InterfaceAction* metoda), 286
- `initialize()` (*calibre.customize.CatalogPlugin* metoda), 262
- `initialize()` (*calibre.customize.Plugin* metoda), 257
- `initialize()` (*calibre.gui2.preferences.ConfigWidgetBase* metoda), 289
- `initialize()` (*calibre.gui2.preferences.ConfigWidgetInterface* metoda), 288
- `InputFormatPlugin` (klasa w module *calibre.customize.conversion*), 266
- `insert_into_xml()`
(*calibre.ebooks.oeb.polish.container.Container* metoda), 364
- `installation_type` (*calibre.customize.Plugin* atrybut), 257
- `InterfaceAction` (klasa w module *calibre.gui2.actions*), 283
- `InterfaceActionBase` (klasa w module *calibre.customize*), 286
- `InternalMetadataCompareKeyGen` (klasa w module *calibre.ebooks.metadata.sources.base*), 265
- `is_configured()` (*calibre.ebooks.metadata.sources.base.Source* metoda), 263
- `is_dir` (*calibre.ebooks.oeb.polish.container.Container* atrybut), 364
- `is_dynamically_controllable()`
(*calibre.devices.interface.DevicePlugin* metoda), 275
- `is_image_collection`
(*calibre.customize.conversion.InputFormatPlugin* atrybut), 266
- `is_link_wanted()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 41
- `is_null()` (*calibre.ebooks.metadata.book.base.Metadata* metoda), 210
- `is_running()` (*calibre.devices.interface.DevicePlugin* metoda), 276
- `is_usb_connected()`
(*calibre.devices.interface.DevicePlugin* metoda), 270
- `items_removed` (*calibre.db.cache.Cache.EventType* atrybut), 354
- `items_renamed` (*calibre.db.cache.Cache.EventType* atrybut), 354
- `items_with_notes_in_book()`
(*calibre.db.cache.Cache* metoda), 359
- `iterlinks()` (*calibre.ebooks.oeb.polish.container.Container* metoda), 364
- ## K
- `keep_only_tags` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 46
- ## L
- `language` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 46
- `library_about_to_change()`
(*calibre.gui2.actions.InterfaceAction* metoda), 285
- `library_changed()`
(*calibre.gui2.actions.InterfaceAction* metoda), 286
- `link_for()` (*calibre.db.cache.Cache* metoda), 359
- `list_extra_files()` (*calibre.db.cache.Cache* metoda), 359
- `load_actual_plugin()`
(*calibre.customize.InterfaceActionBase* metoda), 286
- `load_resources()` (*calibre.customize.Plugin* metoda), 257
- `load_resources()` (*calibre.gui2.actions.InterfaceAction* metoda), 285
- `location_selected()`
(*calibre.gui2.actions.InterfaceAction* metoda), 285
- ## LRF, 379
- `lrf2lrs` command line option
--dont-output-resources, 341
--help, 341
--output, 341
--verbose, 341
--version, 341
-h, 341
-o, 341
- `lrfviewer` command line option
--disable-hyphenation, 342
--help, 342
--profile, 342
--verbose, 342
--version, 342
--visual-debug, 342
--white-background, 342
-h, 342
- `lrs2lrf` command line option
--help, 342
--lrs, 342
--output, 342
--verbose, 342
--version, 342
-h, 342

-o, 342

M

- `make_name_unique()`
(*calibre.ebooks.oeb.polish.container.Container*
metoda), 365
- `MANAGES_DEVICE_PRESENCE`
(*calibre.devices.interface.DevicePlugin* atrybut),
270
- `manifest_has_name()`
(*calibre.ebooks.oeb.polish.container.Container*
metoda), 365
- `manifest_id_map` (*calibre.ebooks.oeb.polish.container.Container*
property), 365
- `manifest_items_of_type()`
(*calibre.ebooks.oeb.polish.container.Container*
metoda), 365
- `manifest_items_with_property()`
(*calibre.ebooks.oeb.polish.container.Container*
metoda), 365
- `manifest_type_map`
(*calibre.ebooks.oeb.polish.container.Container*
property), 365
- `mark_as_cover()` (w module
calibre.ebooks.oeb.polish.cover), 369
- `mark_as_titlepage()` (w module
calibre.ebooks.oeb.polish.cover), 369
- `masthead_url` (*calibre.web.feeds.news.BasicNewsRecipe*
atrybut), 46
- `match_regexps` (*calibre.web.feeds.news.BasicNewsRecipe*
atrybut), 46
- `max_articles_per_feed`
(*calibre.web.feeds.news.BasicNewsRecipe*
atrybut), 46
- `MAX_PATH_LEN` (*calibre.devices.usbms.device.Device*
atrybut), 279
- `merge()` (w module *calibre.ebooks.oeb.polish.split*), 368
- `merge_extra_files()` (*calibre.db.cache.Cache*
metoda), 359
- `Metadata` (klasa w module
calibre.ebooks.metadata.book.base), 209
- `metadata_changed` (*calibre.db.cache.Cache.EventType*
atrybut), 354
- `metadata_for_field()`
(*calibre.ebooks.metadata.book.base.Metadata*
metoda), 210
- `MetadataReaderPlugin` (klasa w module
calibre.customize), 260
- `MetadataWriterPlugin` (klasa w module
calibre.customize), 261
- `mi` (*calibre.ebooks.oeb.polish.container.Container*
property), 365
- `minimum_calibre_version`
(*calibre.customize.Plugin* atrybut), 257
- `module`
calibre.customize, 256
calibre.customize.conversion, 266
calibre.db.cache, 353
calibre.devices.interface, 269
calibre.ebooks.metadata.book.base,
209
calibre.ebooks.metadata.sources.base,
262
calibre.ebooks.oeb.polish.container,
363
calibre.ebooks.oeb.polish.cover, 369
calibre.ebooks.oeb.polish.css, 369
calibre.ebooks.oeb.polish.jacket,
368
calibre.ebooks.oeb.polish.pretty,
368
calibre.ebooks.oeb.polish.replace,
367
calibre.ebooks.oeb.polish.split, 368
calibre.ebooks.oeb.polish.toc, 370
calibre.gui2.tweak_book.boss, 372
calibre.utils.formatter_functions,
190
calibre.web.feeds.news, 39
- `move_book_from_trash()` (*calibre.db.cache.Cache*
metoda), 359
- `move_format_from_trash()`
(*calibre.db.cache.Cache* *metoda*), 359
- `multisort()` (*calibre.db.cache.Cache* *metoda*), 359
- `multisplit()` (w module
calibre.ebooks.oeb.polish.split), 368

N

- `name` (*calibre.customize.Plugin* atrybut), 256
- `name` (*calibre.gui2.actions.InterfaceAction* atrybut), 283
- `name` (*calibre.gui2.tweak_book.plugin.Tool* atrybut), 370
- `name()`, 161
- `name_order` (*calibre.customize.PreferencesPlugin*
atrybut), 287
- `name_to_abspath()`
(*calibre.ebooks.oeb.polish.container.Container*
metoda), 365
- `name_to_href()` (*calibre.ebooks.oeb.polish.container.Container*
metoda), 365
- `names_that_must_not_be_changed`
(*calibre.ebooks.oeb.polish.container.Container*
property), 365
- `names_that_must_not_be_removed`
(*calibre.ebooks.oeb.polish.container.Container*
property), 365
- `names_that_need_not_be_manifested`
(*calibre.ebooks.oeb.polish.container.Container*
property), 365

- p
-
- needs_subscription (calibre.web.feeds.news.BasicNewsRecipe atrybut), 46
-
- NEWS_IN_FOLDER (calibre.devices.usbms.device.Device atrybut), 279
-
- no_stylesheets (calibre.web.feeds.news.BasicNewsRecipe atrybut), 46
-
- normalize_path() (calibre.devices.usbms.driver.USBMS metoda klasy), 283
-
- notes_data_for() (calibre.db.cache.Cache metoda), 360
-
- notes_for() (calibre.db.cache.Cache metoda), 360
-
- notes_resources_used_by() (calibre.db.cache.Cache metoda), 360
-
- NUKE_COMMENTS (calibre.devices.interface.DevicePlugin atrybut), 270
-
- O
-
- oldest_article (calibre.web.feeds.news.BasicNewsRecipe atrybut), 46
-
- on_import (calibre.customize.FileTypePlugin atrybut), 258
-
- on_postconvert (calibre.customize.FileTypePlugin atrybut), 259
-
- on_postdelete (calibre.customize.FileTypePlugin atrybut), 259
-
- on_postimport (calibre.customize.FileTypePlugin atrybut), 259
-
- on_postprocess (calibre.customize.FileTypePlugin atrybut), 259
-
- on_preprocess (calibre.customize.FileTypePlugin atrybut), 259
-
- open() (calibre.devices.interface.DevicePlugin metoda), 272
-
- open() (calibre.devices.usbms.device.Device metoda), 280
-
- open() (calibre.ebooks.oeb.polish.container.Container metoda), 365
-
- open_book() (calibre.gui2.tweak_book.boss.Boss metoda), 372
-
- OPEN_FEEDBACK_MESSAGE (calibre.devices.interface.DevicePlugin atrybut), 270
-
- opf (calibre.ebooks.oeb.polish.container.Container property), 365
-
- opf_get_or_create() (calibre.ebooks.oeb.polish.container.Container metoda), 365
-
- opf_version (calibre.ebooks.oeb.polish.container.Container property), 365
-
- opf_version_parsed (calibre.ebooks.oeb.polish.container.Container property), 365
-
- opf_xpath() (calibre.ebooks.oeb.polish.container.Container metoda), 366
-
- options (calibre.customize.conversion.InputFormatPlugin atrybut), 266
-
- options (calibre.customize.conversion.OutputFormatPlugin atrybut), 268
-
- options (calibre.ebooks.metadata.sources.base.Source atrybut), 262
-
- OSX_MAIN_MEM_VOL_PAT (calibre.devices.usbms.device.Device atrybut), 278
-
- output_encoding (calibre.customize.conversion.InputFormatPlugin atrybut), 266
-
- OutputFormatPlugin (klasa w module calibre.customize.conversion), 267
-
- P
-
- parse_feeds() (calibre.web.feeds.news.BasicNewsRecipe metoda), 42
-
- parse_index() (calibre.web.feeds.news.BasicNewsRecipe metoda), 42
-
- parsed() (calibre.ebooks.oeb.polish.container.Container metoda), 366
-
- path_sep (calibre.devices.interface.DevicePlugin atrybut), 269
-
- Plugin (klasa w module calibre.customize), 256
-
- populate_article_metadata() (calibre.web.feeds.news.BasicNewsRecipe metoda), 42
-
- popup_type (calibre.gui2.actions.InterfaceAction atrybut), 283
-
- post_yank_cleanup() (calibre.devices.interface.DevicePlugin metoda), 272
-
- post_yank_cleanup() (calibre.devices.usbms.device.Device metoda), 280
-
- postadd() (calibre.customize.FileTypePlugin metoda), 260
-
- postconvert() (calibre.customize.FileTypePlugin metoda), 259
-
- postdelete() (calibre.customize.FileTypePlugin metoda), 260
-
- postimport() (calibre.customize.FileTypePlugin metoda), 259
-
- postprocess_book() (calibre.customize.conversion.InputFormatPlugin metoda), 267
-
- postprocess_book() (calibre.web.feeds.news.BasicNewsRecipe metoda), 42
-
- postprocess_html() (calibre.web.feeds.news.BasicNewsRecipe metoda), 42

`pref()` (*calibre.db.cache.Cache* metoda), 360
`prefer_results_with_isbn` (*calibre.ebooks.metadata.sources.base.Source* atrybut), 263
`PreferencesPlugin` (klasa w module *calibre.customize*), 287
`prepare_addable_books()` (*calibre.devices.interface.DevicePlugin* metoda), 275
`preprocess_html()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 43
`preprocess_image()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 43
`preprocess_raw_html()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 43
`preprocess_regexp` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 46
`pretty_all()` (w module *calibre.ebooks.oeb.polish.pretty*), 368
`pretty_css()` (w module *calibre.ebooks.oeb.polish.pretty*), 368
`pretty_html()` (w module *calibre.ebooks.oeb.polish.pretty*), 368
`pretty_xml()` (w module *calibre.ebooks.oeb.polish.pretty*), 368
`print_version()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda klasy), 43
`priority` (*calibre.customize.Plugin* atrybut), 257
`priority` (*calibre.gui2.actions.InterfaceAction* atrybut), 283
`PRODUCT_ID` (*calibre.devices.interface.DevicePlugin* atrybut), 269
`PRODUCT_ID` (*calibre.devices.usbms.device.Device* atrybut), 278
`publication_date()` (*calibre.web.feeds.news.BasicNewsRecipe* metoda), 43
`publication_type` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 47
R
`raw_data()` (*calibre.ebooks.oeb.polish.container.Container* metoda), 366
`re:test()`, 161
`read_backup()` (*calibre.db.cache.Cache* metoda), 360
`recipe_disabled` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 47
`recommendations` (*calibre.customize.conversion.InputFormatPlugin* atrybut), 267
`recommendations` (*calibre.customize.conversion.OutputFormatPlugin* atrybut), 268
`recursions` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 47
`refresh_gui()` (*calibre.gui2.preferences.ConfigWidgetInterface* metoda), 288
`register()` (*calibre.gui2.preferences.ConfigWidgetBase* metoda), 289
`register_shortcut()` (*calibre.gui2.tweak_book.plugin.Tool* metoda), 371
`relpath()` (*calibre.ebooks.oeb.polish.container.Container* metoda), 366
`remove_attributes` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 47
`remove_book()` (*calibre.devices.interface.BookList* metoda), 277
`remove_books()` (*calibre.db.cache.Cache* metoda), 360
`remove_books_from_metadata()` (*calibre.devices.interface.DevicePlugin* metoda klasy), 274
`remove_books_from_metadata()` (*calibre.devices.usbms.driver.USBMS* metoda), 282
`remove_empty_feeds` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 47
`remove_formats()` (*calibre.db.cache.Cache* metoda), 360
`remove_from_spine()` (*calibre.ebooks.oeb.polish.container.Container* metoda), 366
`remove_from_xml()` (*calibre.ebooks.oeb.polish.container.Container* metoda), 366
`remove_item()` (*calibre.ebooks.oeb.polish.container.Container* metoda), 366
`remove_items()` (*calibre.db.cache.Cache* metoda), 360
`remove_jacket()` (w module *calibre.ebooks.oeb.polish.jacket*), 368
`remove_javascript` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 47
`remove_stale_user_metadata()` (*calibre.ebooks.metadata.book.base.Metadata* metoda), 211
`remove_tags` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 47
`remove_tags_after` (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 47

remove_tags_before
(calibre.web.feeds.news.BasicNewsRecipe
atrybut), 47

remove_unused_css() (w module
calibre.ebooks.oeb.polish.css), 369

rename() (calibre.ebooks.oeb.polish.container.Container
metoda), 366

rename_extra_files() (calibre.db.cache.Cache
metoda), 360

rename_files() (w module
calibre.ebooks.oeb.polish.replace), 367

rename_items() (calibre.db.cache.Cache metoda),
360

replace() (calibre.ebooks.oeb.polish.container.Container
metoda), 366

replace_links() (calibre.ebooks.oeb.polish.container.Container
metoda), 366

replace_links() (w module
calibre.ebooks.oeb.polish.replace), 367

requires_version(calibre.web.feeds.news.BasicNewsRecipe
atrybut), 48

reset() (calibre.devices.interface.DevicePlugin metoda),
271

reset() (calibre.devices.usbms.device.Device metoda),
279

resolve_internal_links
(calibre.web.feeds.news.BasicNewsRecipe
atrybut), 48

restart_critical(calibre.gui2.preferences.ConfigWidgetBase
atrybut), 288

restart_critical(calibre.gui2.preferences.ConfigWidgetInterface
atrybut), 288

restore_book() (calibre.db.cache.Cache metoda),
360

restore_defaults()
(calibre.gui2.preferences.ConfigWidgetBase
metoda), 289

restore_defaults()
(calibre.gui2.preferences.ConfigWidgetInterface
metoda), 288

restore_defaults_desc
(calibre.gui2.preferences.ConfigWidgetInterface
atrybut), 288

restore_original_format()
(calibre.db.cache.Cache metoda), 360

reverse_article_order
(calibre.web.feeds.news.BasicNewsRecipe
atrybut), 48

rewind_savepoint()
(calibre.gui2.tweak_book.boss.Boss metoda),
372

RSS, 379

run() (calibre.customize.CatalogPlugin metoda), 262

run() (calibre.customize.FileTypePlugin metoda), 259

S

safe_read_lock (calibre.db.cache.Cache property),
361

sanitize_callback()
(calibre.devices.usbms.device.Device metoda),
280

sanitize_path_components()
(calibre.devices.usbms.device.Device metoda),
280

save_book() (calibre.gui2.tweak_book.boss.Boss
metoda), 373

save_original_format() (calibre.db.cache.Cache
metoda), 361

save_settings() (calibre.customize.Plugin metoda),
257

save_settings() (calibre.devices.interface.DevicePlugin
metoda klasy), 274

save_settings() (calibre.ebooks.metadata.sources.base.Source
metoda), 263

scale_news_images
(calibre.web.feeds.news.BasicNewsRecipe
atrybut), 48

scale_news_images_to_device
(calibre.web.feeds.news.BasicNewsRecipe
atrybut), 48

search() (calibre.db.cache.Cache metoda), 361

search_notes() (calibre.db.cache.Cache metoda),
361

serialize_item() (calibre.ebooks.oeb.polish.container.Container
metoda), 366

set_all_user_metadata()
(calibre.ebooks.metadata.book.base.Metadata
metoda), 210

set_conversion_options()
(calibre.db.cache.Cache metoda), 361

set_cover() (calibre.db.cache.Cache metoda), 361

set_cover() (w module
calibre.ebooks.oeb.polish.cover), 369

set_driveinfo_name()
(calibre.devices.interface.DevicePlugin metoda),
275

set_driveinfo_name()
(calibre.devices.usbms.driver.USBMS metoda),
281

set_field() (calibre.db.cache.Cache metoda), 361

set_identifier() (calibre.ebooks.metadata.book.base.Metadata
metoda), 210

set_identifiers()
(calibre.ebooks.metadata.book.base.Metadata
metoda), 210

set_library_info()
(calibre.devices.interface.DevicePlugin metoda),
275

set_link_map() (calibre.db.cache.Cache metoda),

361
 set_metadata() (*calibre.customize.MetadataWriterPlugin metoda*), 261
 set_metadata() (*calibre.db.cache.Cache metoda*), 362
 set_modified() (*calibre.gui2.tweak_book.boss.Boss metoda*), 373
 set_notes_for() (*calibre.db.cache.Cache metoda*), 362
 set_option() (*calibre.devices.interface.DevicePlugin metoda*), 276
 set_plugboards() (*calibre.devices.interface.DevicePlugin metoda*), 274
 set_pref() (*calibre.db.cache.Cache metoda*), 362
 set_progress_reporter() (*calibre.devices.interface.DevicePlugin metoda*), 272
 set_progress_reporter() (*calibre.devices.usbms.device.Device metoda*), 279
 set_spine() (*calibre.ebooks.oeb.polish.container.Container metoda*), 366
 set_user_blacklisted_devices() (*calibre.devices.interface.DevicePlugin metoda*), 275
 set_user_metadata() (*calibre.ebooks.metadata.book.base.Metadata metoda*), 211
 settings() (*calibre.devices.interface.DevicePlugin metoda klasy*), 274
 show_current_diff() (*calibre.gui2.tweak_book.boss.Boss metoda*), 373
 show_editor() (*calibre.gui2.tweak_book.boss.Boss metoda*), 373
 shutdown() (*calibre.devices.interface.DevicePlugin metoda*), 275
 shutting_down() (*calibre.gui2.actions.InterfaceAction metoda*), 286
 simultaneous_downloads (*calibre.web.feeds.news.BasicNewsRecipe atrybut*), 48
 skip_ad_pages() (*calibre.web.feeds.news.BasicNewsRecipe metoda*), 43
 SLOW_DRIVEINFO (*calibre.devices.interface.DevicePlugin atrybut*), 270
 smart_update() (*calibre.ebooks.metadata.book.base.Metadata metoda*), 211
 sort_index_by() (*calibre.web.feeds.news.BasicNewsRecipe metoda*), 43
 Source (*klasa w module calibre.ebooks.metadata.sources.base*), 262
 specialize() (*calibre.customize.conversion.InputFormatPlugin metoda*), 267
 specialize_css_for_output() (*calibre.customize.conversion.OutputFormatPlugin metoda*), 268
 specialize_global_preferences() (*calibre.devices.interface.DevicePlugin metoda*), 275
 specialize_options() (*calibre.customize.conversion.OutputFormatPlugin metoda*), 268
 spine_items (*calibre.ebooks.oeb.polish.container.Container property*), 367
 spine_iter (*calibre.ebooks.oeb.polish.container.Container property*), 367
 spine_names (*calibre.ebooks.oeb.polish.container.Container property*), 367
 split() (*w module calibre.ebooks.oeb.polish.split*), 368
 split_if_is_multiple_composite() (*calibre.db.cache.Cache metoda*), 362
 split_jobs() (*calibre.ebooks.metadata.sources.base.Source metoda*), 264
 standard_field_keys() (*calibre.ebooks.metadata.book.base.Metadata metoda*), 210
 STANDARD_METADATA_FIELDS (*w module calibre.ebooks.metadata.book.base*), 211
 start_plugin() (*calibre.devices.interface.DevicePlugin metoda*), 276
 startup() (*calibre.devices.interface.DevicePlugin metoda*), 275
 stop_plugin() (*calibre.devices.interface.DevicePlugin metoda*), 276
 summary_length (*calibre.web.feeds.news.BasicNewsRecipe atrybut*), 48
 supported_platforms (*calibre.customize.conversion.InputFormatPlugin atrybut*), 266
 supported_platforms (*calibre.customize.conversion.OutputFormatPlugin atrybut*), 268
 supported_platforms (*calibre.customize.InterfaceActionBase atrybut*), 286
 supported_platforms (*calibre.customize.MetadataReaderPlugin atrybut*), 260
 supported_platforms (*calibre.customize.MetadataWriterPlugin atrybut*), 261
 supported_platforms (*calibre.customize.Plugin atrybut*), 256
 supported_platforms (*calibre.customize.PreferencesPlugin atrybut*), 287
 supported_platforms

(*calibre.devices.usbms.driver.USBMS* atrybut), 281

supported_platforms (*calibre.ebooks.metadata.sources.base.Source* atrybut), 262

supports_collections() (*calibre.devices.interface.BookList* metoda), 277

supports_gzip_transfer_encoding (*calibre.ebooks.metadata.sources.base.Source* atrybut), 262

supports_restoring_to_defaults (*calibre.gui2.preferences.ConfigWidgetBase* atrybut), 288

supports_restoring_to_defaults (*calibre.gui2.preferences.ConfigWidgetInterface* atrybut), 288

sync_booklists() (*calibre.devices.interface.DevicePlugin* metoda), 274

sync_booklists() (*calibre.devices.usbms.driver.USBMS* metoda), 283

sync_preview_to_editor() (*calibre.gui2.tweak_book.boss.Boss* metoda), 373

synchronize_with_db() (*calibre.devices.interface.DevicePlugin* metoda), 276

T

tag_browser_context_action() (*calibre.gui2.actions.InterfaceAction* metoda), 286

tag_to_string() (*calibre.web.feeds.news.BasicNewsRecipe* metoda klasy), 43

tags_older_than() (*calibre.db.cache.Cache* metoda), 362

template_css (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 48

template_to_attribute() (*calibre.ebooks.metadata.book.base.Metadata* metoda), 211

temporary_file() (*calibre.customize.Plugin* metoda), 258

test_fields() (*calibre.ebooks.metadata.sources.base.Source* metoda), 264

THUMBNAIL_COMPRESSION_QUALITY (*calibre.devices.interface.DevicePlugin* atrybut), 269

THUMBNAIL_HEIGHT (*calibre.devices.interface.DevicePlugin* atrybut), 269

timefmt (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 48

timeout (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 48

title (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 48

to_html() (*calibre.ebooks.metadata.book.base.Metadata* metoda), 211

Tool (klasa w module *calibre.gui2.tweak_book.plugin*), 370

toolbar_button_popup_mode (*calibre.gui2.tweak_book.plugin.Tool* atrybut), 371

total_space() (*calibre.devices.interface.DevicePlugin* metoda), 272

total_space() (*calibre.devices.usbms.device.Device* metoda), 279

touched_fields (*calibre.ebooks.metadata.sources.base.Source* atrybut), 262

type (*calibre.customize.CatalogPlugin* atrybut), 261

type (*calibre.customize.conversion.InputFormatPlugin* atrybut), 266

type (*calibre.customize.conversion.OutputFormatPlugin* atrybut), 267

type (*calibre.customize.FileTypePlugin* atrybut), 259

type (*calibre.customize.InterfaceActionBase* atrybut), 286

type (*calibre.customize.MetadataReaderPlugin* atrybut), 260

type (*calibre.customize.MetadataWriterPlugin* atrybut), 261

type (*calibre.customize.Plugin* atrybut), 257

type (*calibre.customize.PreferencesPlugin* atrybut), 287

type (*calibre.devices.interface.DevicePlugin* atrybut), 269

type (*calibre.ebooks.metadata.sources.base.Source* atrybut), 262

U

unretire_note_for() (*calibre.db.cache.Cache* metoda), 362

upload_books() (*calibre.devices.interface.DevicePlugin* metoda), 273

upload_books() (*calibre.devices.usbms.driver.USBMS* metoda), 282

upload_cover() (*calibre.devices.usbms.driver.USBMS* metoda), 282

URL, **379**

USBMS (klasa w module *calibre.devices.usbms.driver*), 281

use_embedded_content (*calibre.web.feeds.news.BasicNewsRecipe* atrybut), 48

user_categories_for_books() (*calibre.db.cache.Cache* metoda), 362

user_feedback_after_callback (*calibre.devices.interface.DevicePlugin* atrybut), 270

UserAnnotation (*calibre.devices.interface.DevicePlugin* atrybut), 270

V

VENDOR_ID (*calibre.devices.interface.DevicePlugin* atrybut), 269

VENDOR_ID (*calibre.devices.usbms.device.Device* atrybut), 278

version (*calibre.customize.MetadataReaderPlugin* atrybut), 260

version (*calibre.customize.MetadataWriterPlugin* atrybut), 261

version (*calibre.customize.Plugin* atrybut), 256

VIRTUAL_BOOK_EXTENSION_MESSAGE (*calibre.devices.interface.DevicePlugin* atrybut), 270

VIRTUAL_BOOK_EXTENSIONS (*calibre.devices.interface.DevicePlugin* atrybut), 270

W

WANTS_UPDATED_THUMBNAILS (*calibre.devices.interface.DevicePlugin* atrybut), 269

web2disk command line option

- base-dir, 343
- delay, 343
- dont-download-stylesheets, 343
- encoding, 343
- filter-regexp, 343
- help, 343
- match-regexp, 343
- max-files, 343
- max-recursions, 343
- timeout, 343
- verbose, 343
- version, 343
- d, 343
- h, 343
- n, 343
- r, 343
- t, 343

WINDOWS_CARD_A_MEM (*calibre.devices.usbms.device.Device* atrybut), 278

WINDOWS_CARD_B_MEM (*calibre.devices.usbms.device.Device* atrybut), 278

WINDOWS_MAIN_MEM (*calibre.devices.usbms.device.Device* atrybut), 278

windows_sort_drives() (*calibre.devices.usbms.device.Device* metoda), 280

wyrażenie regularne, **379**



źródło, **379**